IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 4, 2021, accepted April 20, 2021, date of publication April 22, 2021, date of current version May 4, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3075069

Digital Twin Driven Requirement Conversion in Smart Customized Design

YE GU, SHUYOU ZHANG, AND LEMIAO QIU
State Key Laboratory of Fluid Power and Mechatronic Systems, Zhejiang University, Hangzhou 310027, China

Corresponding author: Lemiao Qiu (qiulm@zju.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1700701, in part by the National Natural Science Foundation of China under Grant 51875516, and in part by the Jiangsu Province Science and Technology Achievement Transforming Fund Project of China under Grant BA2018083.

ABSTRACT Requirement conversion (RC) is an important activity in product customized design. Nowadays, the RC process is mostly driven by designers' knowledge, the RC model is passive to dynamic customer requirements, and the RC behavior is a black box to customers. Digital twin (DT) is characterized as a self-reinforcing mechanism driven by mirroring between the physical and virtual spaces, which is powerful in addressing these challenges for RC. This paper proposes a digital twin driven requirement conversion (DTRC) architecture and a tri-model-based approach integrated by digital model, behavior model, and evolution model for DTRC development. Firstly, the digital model based on the artificial neural network can simulate the virtual twin data to compensate for the absence of real-world data. Then, driven by the virtual-reality integration data, the behavior model mirrors and visualizes the RC behavior in real world based on the decision tree. Finally, a genetic algorithm based evolution model optimizes the RC rules via physical data throughout the whole product life cycle. A case study of DTRC for elevator customized design is further conducted to validate feasibility and effectiveness of the proposed approach. Experimental results show that DTRC outperforms other RC approaches in terms of conversion accuracy. Meanwhile, DTRC can visualize and optimize the conversion path through the tree topology, which is beneficial to the customer participation and proactive to the dynamic environment.

INDEX TERMS Requirement conversion, digital twin, smart customization, tri-model-based approach, elevator customized design.

I. INTRODUCTION

Nowadays, more than half of the industrial products are customized [1]. The three cores of product customization are the fulfilment of diversified customer requirements, the rapid response ability of manufacturers, and the low customization cost for manufacturers. Product customization capabilities have been the important metrics to evaluate the viability and competitiveness of all kinds of manufacturers. With the advances in Internet and 5G communication, big data driven smart manufacturing has become the focus of the transformation and upgrading of global manufacturing [2]. Traditional mass customization is shifting to data-driven smart customization [3], which is a more intelligent customer-oriented decision-making paradigm regarding the whole product life cycle (PLC) [4]. Customized design is the initiation of product customization, directing the economic, social, cultural, and ecological value of the whole PLC.

The success of customized products depends mainly on the identification and fulfilment of personalized customer requirements [5]. Requirement conversion (RC), an important activity in customized design, is defined by the conversion from customer requirements to design specifications [6]. Customer requirements in customization are unconstrained, leading to different design specifications with different orders. If there exists an error in design specifications, specific aspects of PLC will be susceptible to a domino effect of defaults, which may undermine the final products, lower customers' satisfaction, and cause a restart.

From a review of existing literature, there exist four main technical challenges for RC. Firstly, the driving force of RC is mostly the designers' knowledge and experience, insufficient use of big data through PLC. Secondly, the data driving RC comes from the real world, which is constrained by what

The associate editor coordinating the review of this manuscript and approving it for publication was Jingang Jiang.

has already occurred and is limited in the accuracy and individuality of RC results. Then, existing RC approaches are usually passive to the uncertainty and variability of customer requirements. Finally, the RC process is a black box, limiting customer participation and interaction. Therefore, a static and unexplainable RC model solely driven by single-source data is ineffective in smart customization.

Digital twin (DT) is an emerging technology to fulfil the requirements of smart customization by mirroring the physical status of customization in a virtual space [4]. DT can be extended into five dimensions: physical entity (PE), virtual entity (VE), DT data (DD), connections (CN), and services (Ss) [7]. VE is the mirror image of PE. DD is the data collected in physical space and simulated in virtual space in real-time. CN enables the co-evolution between PE and VE by updating VE with DD and feeding the improvements of VE into PE. Ss improves the practical value of DT, where stakeholders limited by domain knowledge can also participate in the decision-making process. DT is characterized as a self-reinforcing mechanism driven by the convergence, integration, and synchronization of the physical and virtual data, which is powerful in addressing major technical challenges for RC.

Given the major challenges for RC and the potentiality of DT, the authors are motivated to employ DT to augment RC in the context of smart customization. The rest of the paper aims to demonstrate this concept by providing a novel digital twin driven requirement conversion (DTRC) framework. A tri-model-based approach integrated by digital model, behavior model, and evolution model is proposed for DTRC development. A case study on the elevator customized design is further conducted to validate feasibility and effectiveness of DTRC. The novelties of DTRC include (1) taking advantage of big data to drive RC for better performance; (2) being proactive to the uncertain and dynamic customer requirements; (3) promoting customer participation by visualizing RC behavior and rules.

This paper is structured as follows. Section II reviews the related work in RC and the DT applications in product design. Section III presents the problem description and framework formulation for DTRC. Following that, Section IV details a tri-model-based approach for DTRC development, i.e., digital model, behavior model, and evolution model. A case study of DTRC for elevator customized design is conducted in Section V. Concluding remarks and future work in Section VI end the paper.

II. LITERATURE REVIEW

A. RC IN PRODUCT CUSTOMIZATION

RC is defined by the transformation from customer requirements to design specifications. Yin *et al.* [8] focused on the relationships between product supply and customer requirements in the context of Industry 2.0–4.0, and concluded that a production system in the future has to adapt to the environment with diversified customer requirements. Shao *et al.* [9] analyzed the dependency relationships between customer groups and clusters of product specifications by rough set theory to discover the conversion rules. Geng *et al.* [10] proposed three-domain RC framework based on quality function deployment (QFD), where three domains include dependency relationships between and within customer requirements, product-, and service-related engineering characteristics. Lee *et al.* [11] combined association rule mining and decision tree to discover the relationships among items. Wang [12] adopted grey system theory to determine the influence weighting of form parameters and support vector regression to explore the bi-directional relationship between customer needs and product forms in Kansei engineering.

Some intelligent algorithms, such as data mining and knowledge driving, in RC studies have also attracted researchers' attention. Yu *et al.* [13] defined the topology of neural network by the implicit design knowledge and extracts RC knowledge by decision tree. Song *et al.* [14] combined rough set theory and grey relational analysis to prioritize more rationally the technical attributes in QFD-based RC model. Ma *et al.* [15] proposed the multidisciplinary requirement modeling to adapt to the complex mechatronic products, which is driven by experts' knowledge. Beernaert and Etman [16] coordinated RC decisions that involve multiple components of a complex product by analytical target cascading. Li *et al.* [17] adopted the knowledge graph technique and concept-knowledge model to possess the knowledge reasoning ability under multidisciplinary fields and achieve hybrid intelligence by user-generated data. Cong *et al.* [18] proposed design context-awareness and design entropy theory to represent the process of entropy reduction in RC.

Although many approaches in previous research have explored and improved the performance of RC, these methods are driven mostly by the single-source data, rather than the continuously updating big data. As shown in Table 1, three characteristics of requirement conversion differentiate smart customization from traditional customization. On the basis of existing RC approaches, RC in smart customization should innovate on big data driving, adaptive evolution, and explainable conversion.

TABLE 1. Comparison of requirement conversion in traditional customization and smart customization.

Characteristic	Traditional customization	Smart customization
Driving force	Designers' knowledge and experience	Big data, including physical PLC data and virtual data compensating for the absence of real-world data
Response to uncertain requirements	Passive	Proactive
Explainability of decision-making process	Black box	White box, promoting customer participation

B. DT IN PRODUCT DESIGN

DT technology has been proved to be the efficient path to realize Industry 4.0 [19]. Although the five sequential stages of PLC are equally important [20], more than half of DTs are applied in the manufacturing and usage stages [21]. With the product design process becoming more digitalized than ever before, DT-enhanced product design (i.e., the first stage of PLC) is gaining ever-increasing attention [22].

DT-based design focuses on the convergence and co-evolution between product physical and virtual spaces, which is helpful to quickly and correctly rule out the design mistakes [23], optimize the decision-making process [24], and verify the design results [25]. Tao *et al.* [22] proposed a DT-driven product design framework for the iterative redesign of an existing product instead of a completely new product, focusing on the convergence between product physical and virtual data sources. Guo *et al.* [23] used a modular method to develop a flexible DT, avoiding the problems caused by design change and hidden design flaws. Liu *et al.* [25] proposed DT-based design approach to realize the hardware-in-the-loop simulation to locate the design deficiencies and avoid reconfiguration. Liu *et al.* [26] introduced digital twin-driven methodology, called "iterative design optimization between static configuration and dynamic execution," for rapid individualized design.

DT-based design is robust for highly dynamic production environments by context awareness. Schleich *et al.* [27] proposed a skin model shapes based DT for geometrical variations management in design, with properties of scalability, interoperability, expansibility, and fidelity. Dias-Ferreira *et al.* [28] presented a bio-inspired self-organizing architecture for highly dynamic production environments, where DT was used to visualize the various interaction patterns. Motivated by context awareness and the knowledge graph based decision support, Lim *et al.* [29] proposed a DT-enhanced system for product family design and optimization. Leng *et al.* [30] proposed a context-aware digital twin solution for adaptive synchronization between cyber and physical systems under the dynamic blockchain network topology.

DT-based design can promote customer participation and interaction for mass individualization and customer-oriented paradigm. Zheng *et al.* [31] proposed a novel platform-based, data-driven, and DT-enabled design approach for service innovation to achieve individual customer satisfaction with less environment impact, meanwhile, the proposed DT [32] can provide the service of co-design/co-creation of products for customers in the cloud-based environment. Wang *et al.* [4] investigated a new paradigm of DT-driven smart customization, where DT not only allows designers to better understand customer requirements and spark new design concepts but also allows customers to design and modify their products at any time to adapt to their new requirements.

DT-enhanced product design is characterized by intelligence in design process [23], [24], robustness for dynamic environments [28], [31] and promotion for customer interaction [4], [32] in some researches. Although these studies reflected DT influence in the design stage, integrating DT to augment RC—the beginning of the design process—had not been considered. Dassault Systèmes Company in Paris, France, says that creating and leveraging DT at the beginning of the process (not later on in the detailed design stage) can realize the huge potential of DT [33]. To fill this gap, a transparent DTRC capable of big data driving and self-reinforcement is essentially required to aid smart customized design, thereby improving smart customization processes. Meanwhile, DTRC fulfils only a single PLC aspect. An indepth study of DT development for a single problem helps understand the problem-solving mechanism. DTRC can also be integrated into the practical engineering DT regarding the whole PLC.

TABLE 2. Symbols used in this paper.

Symbol	Description
$\boldsymbol{CR}$	Customer requirement vector
$\boldsymbol{DS}$	Design specification vector
cr	Value of the customer requirement feature
ds	Value of the design specification feature
Ω_{CR}	Customer requirements domain
Ω_{DS}	Design specifications domain
δ	Input value of the digital model (encoding of cr)
ω	Output value of the digital model (decoding of ds)
M_{initial}	Behavior model
M_{DT}	Digital twin model
$Tree$	Visualization of RC
S	Virtual-reality integration dataset
$Rule$	RC rules extracted from $Tree$
A	Judgment in $Rule$
x	Encoding of $Rule$

III. FRAMEWORK OF DIGITAL TWIN DRIVEN REQUIREMENT CONVERSION (DTRC)

A. PROBLEM DESCRIPTION

This paper addresses a decision-making problem called RC at the beginning of product customized design, which begins with customer requirements and end with a set of design specifications. The parameters used in this paper are summarized in Table 2

RC process can be expressed as

$$\boldsymbol{DS} = f(\boldsymbol{CR}), \tag{1}$$

where $\boldsymbol{CR} = [cr_1, cr_2, \ldots, cr_n] \in \Omega_{CR}$ and $\boldsymbol{DS} = [ds_1, ds_2, \ldots, ds_m] \in \Omega_{DS}$ represent the customer requirements and design specifications, respectively. The customer requirements domain Ω_{CR} is then defined as a set of $\boldsymbol{CR}$s, while the design specifications domain Ω_{DS} is defined as a set of $\boldsymbol{DS}$s. $f(\cdot)$ is the nonlinear mapping process from Ω_{CR} to Ω_{DS}. cr_i, $i = 1, 2, \ldots, n$, is the i^{th} feature in $\boldsymbol{CR}$, while ds_j, $j = 1, 2, \ldots, m$, is the j^{th} feature in $\boldsymbol{DS}$. Each feature, cr_i or ds_j, is either categorical variable (one out of a finite set of options) or continuous variable (unconstrained numerical value). Ω^i_{CR} is the set of cr_i—the sub-domain of Ω_{CR}. Ω^j_{DS} is the set of ds_j.

Following this opinion, Fig. 1 illustrates the principle of RC. The complexity and variability of RC in customization are caused by the ever-changing individual requirements from customers and the "engineer to order (ETO)" production paradigm. The driving force of RC is mostly based on designers' knowledge and experience combined with QFD [10],

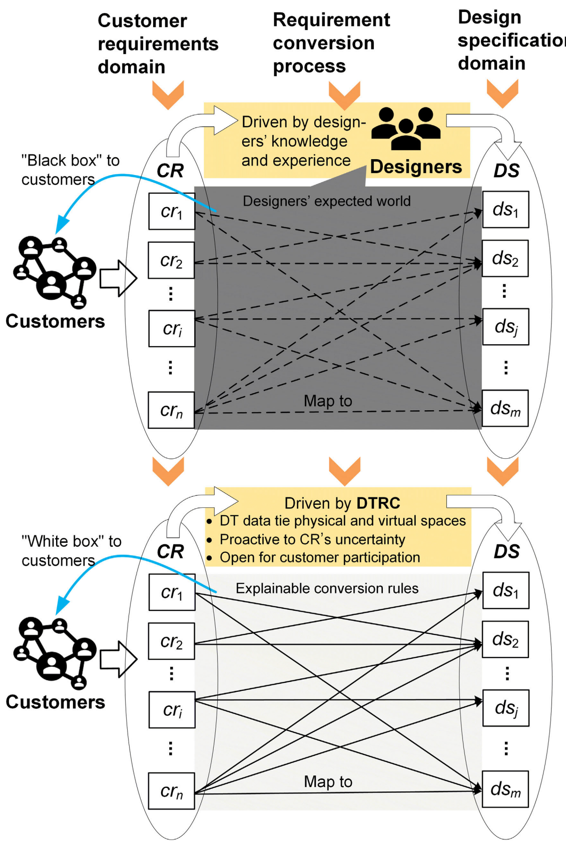


FIGURE 1. Problem formulation of requirement conversion.

[14] model. The RC process is conducted in the designers' expected world and is a black box to customers, which limits customer participation.

Therefore, this paper aims at dealing with the problems mentioned above through analysis of dynamic dependency relationships between Ω_{CR} and Ω_{DS} by DT, as well as conversion rules mining in the RC process. The proposed DTRC aims to discover and visualize conversion rules depending on the customer groups rather than the individual customer, as well as DT data tying physical and virtual worlds rather than only the static domain knowledge. DTRC is characterized by the following

1) DTRC simulates sufficient data via virtual models mirroring the physical space, compensating for the absence of real-world data.
2) DTRC visualizes the RC process, which is a white box to customers, and can promote customer participate at the beginning of product customized design.
3) DTRC enables the co-evolution between physical and virtual spaces, predictable and proactive to the uncertain and diversified customer requirements in future.

B. FRAMEWORK FORMULATION

DT for product customization can be likened to a complete living body regarding the whole PLC. DT is depicted as a five-dimensional composition expression,

$$M_{DT} = \{PE, VE, Ss, DD, CN\}, \tag{2}$$

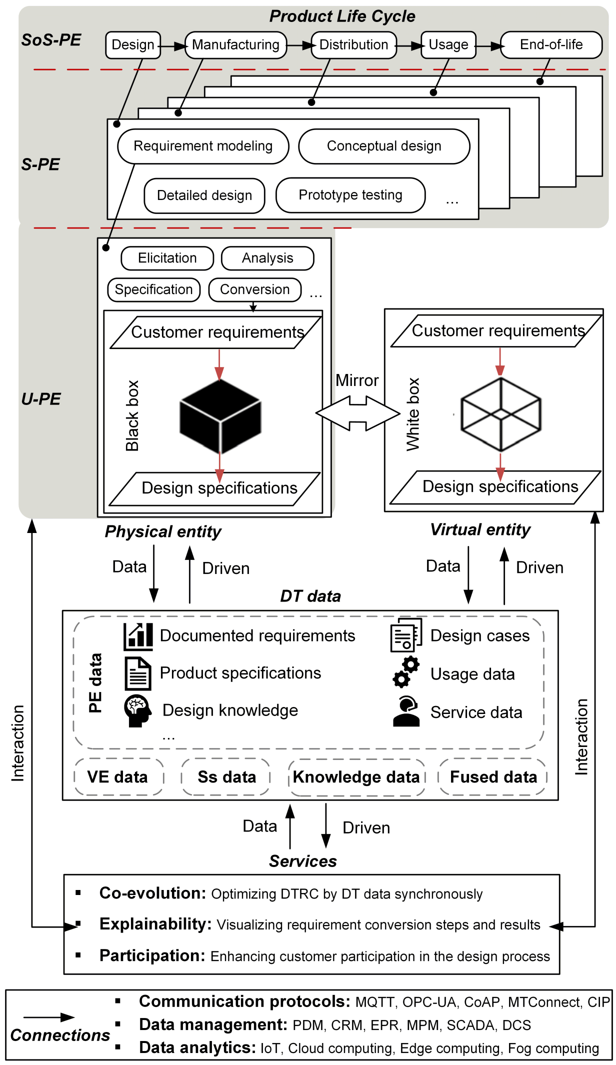


FIGURE 2. Digital twin driven requirement conversion framework.

where PE is the physical entity, VE is the virtual entity, Ss stands for services provided by DT, DD refers to DT data, and CN is the connections among the previous four parts. VE is the mirror image of PE regarding the whole PLC, equivalent to different organs in the living body, where DD is the blood, CN is the blood vessels, and Ss is the functions of organs. Evidently, a single organ can share blood throughout the living body. DTRC is equivalent to a single organ in DT for product customization, as shown in Fig. 2.

- *Physical entity (PE)*
  PE for product customization consists of all stages of PLC. PE can be divided into three levels: system-to-system level (SoS-PE), system level (S-PE), and unit level (U-PE). PLC can be divided into five sequential SoS-PEs [20]: design, manufacturing, distribution, usage, and end-of-life. SoS-PE for design consists of some S-PEs, such as requirement modeling, conceptual design, detailed design, and prototype testing. S-PE for requirement modeling is then divided into some U-PEs, e.g., elicitation, analysis, specification, and conversion.

U-PE for RC (i.e., PE in DTRC) is a realistic RC behavior driven by designers' domain knowledge in physical space, which is an expected world to designers and a black box to customers.

- *Virtual entity (VE)*
  VE in DTRC is the mirror image of RC behavior in physical space, including the behavior model and rule model. The behavior model is the real-time response of RC process, focusing on the interactions among customer requirements at different time scales (external environment) and the driving force of the RC (internal mechanism). The rule model includes mainly the evaluation and evolution models established following the laws of RC in physical space. Stakeholders are enabled to interact directly with VE in a virtual environment with high authenticity and feed the improvements of VE into PE with real-time feedback.
- *Services (Ss)*
  Ss in DTRC includes internal and external services. Internal services encapsulate different tools, components, and modules to support DTRC working. External services are provided to meet stakeholders' demands for DTRC. Designers input customer requirements into DTRC and then obtain accurate design specifications, while DTRC can optimize RC results from ever-changing customer requirements every time. Customers can clarify the logic and steps of the RC process from DTRC and participate in the product customized design.
- *DT data (DD)*
  DD is collected from both physical and virtual spaces, including PE data (DD_p), VE data (DD_v), Ss data (DD_s), knowledge data (DD_k), and fused data (DD_f).

  1) DD_p is the data collected from all four SoS-PEs, mainly including real-world historical "$\boldsymbol{CR}-\boldsymbol{DS}$" data and product operation and maintenance data.
  2) DD_v is the simulated data augmented by VE to compensate for the absence of real-world "$\boldsymbol{CR}-\boldsymbol{DS}$" data.
  3) DD_s is the data collected from Ss, including RC methods, models, algorithms, and running log in internal services and new "$\boldsymbol{CR}-\boldsymbol{DS}$" data addressed by DTRC in external services.
  4) DD_k represents the domain knowledge used in the DTRC development.
  5) DD_f is deployed for the former four data convergence, integration, and analytics. For example, product operation and maintenance data from DD_p need preprocessing to support the self-reinforcement of DTRC.
- *Connections (CN)*
  There are six types of CN among PE, VE, Ss, and DD. CN is then introduced from three stages of DTRC.

  1) Installation stage: VE simulates DD_v by DD_p provided from PE. DD_f integrating DD_v and DD_p drives the DTRC installation.

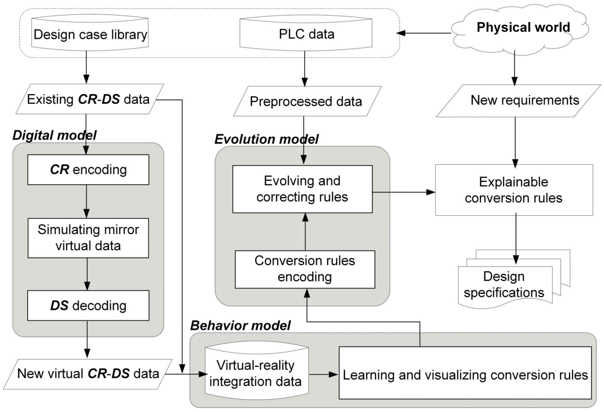


FIGURE 3. Tri-model-based approach for DTRC development.

  2) Running stage: DD_p provided from PE is preprocessed by DD and then passed into VE, driving the DTRC evolution synchronously by the internal services in Ss.
  3) Service stage: PE provides the new customer requirements to VE. The RC steps and results are then presented to the stakeholders by the external services in Ss

  CN can be empowered by a number of technologies, such as communication protocols, data management, data analytics, to ensure the development of DTRC.

IV. TRI-MODEL-BASED APPROACH FOR DTRC DEVELOPMENT

DTRC is a complex digital object driven by DT data with visualization and evolution capability, and is predictable and proactive to the dynamic environment, which is achieved by a tri-model-based approach (i.e., digital model, behavior model, and evolution model, see Fig. 3). As a virtual-reality integration technology, DTRC can compensate for the absence of real-world RC data via the digital model. The behavior model, driven by the virtual-reality integration RC data, can mirror and visualize the RC rules from the physical behavior. The evolution model optimizes the RC rules synchronously by the physical PLC data.

A. DIGITAL MODEL

A digital model is used to simulate virtual data by the real-world physical "$\boldsymbol{CR}-\boldsymbol{DS}$" data. Hence, the virtual data should be as accurate as possible. The digital model in this paper is based on the neural network (NN), with excellent generalization ability and fast simulation speed [34], as shown in Fig. 4.

The input and output of the digital model are $\boldsymbol{CR}$ and $\boldsymbol{DS}$, respectively. The encoding of $\boldsymbol{CR}$ and decoding of $\boldsymbol{DS}$ are introduced in Table 3, which is classified into continuous and categorical variables. Neurons in the input and output layers of the NN-based digital model are denoted as δ and ω, respectively. δ and ω for continuous variables are calculated by normalization and renormalization; while binary coding is used for categorical variables.

TABLE 3. Encoding and decoding in NN-based digital model.

Classification	Number of neurons	Encoding of cr_i	Decoding of ds_j
Continuous variable	1	$\delta_1 = \frac{cr_i - \min\{\Omega^i_{CR}\}}{\max\{\Omega^i_{CR}\} - \min\{\Omega^i_{CR}\}}$	$ds_j = \left(\max\left\{\Omega^j_{DS}\right\} - \min\left\{\Omega^j_{DS}\right\}\right) \cdot \omega_1 + \min\left\{\Omega^j_{DS}\right\}$
Categorical variable	$\lceil \log_2 N \rceil$ with N categories	for $k = 1, k \le \lceil \log_2 N \rceil, k++$ $\delta_k \leftarrow cr_i \bmod 2$; $cr_i \leftarrow cr_i \vert 2$; end	$ds_j \leftarrow \sum_{k=1,2,\ldots,\lceil \log_2 N \rceil} \omega_k 2^{k-1}$

The NN-based digital model is a multilayer feedforward neural network trained by the error backpropagation algorithm. The training process repeats two stages, signal feedforward and error backpropagation, until the successive iterations no longer produce better results. In the signal feedforward stage, the signal is transmitted from the input layer to the hidden layer for processing, and then to the output layer. The output value is compared with the expected value to calculate the error. In the error backpropagation stage, the cost function of error is used to update the weights and biases by gradient descent. The training of the digital model can be regarded as the process of reducing errors, as shown in Algorithm 1.

In Algorithm 1, cost function J is obtained by adding regularizer $\lambda\Omega(\theta)$ to the loss function $E(\hat{\boldsymbol{\omega}}, \boldsymbol{\omega}) = -\sum_{j=1}^{m}\left[\left(1-\hat{\omega}_j\right)\log_2\left(1-\omega_j\right) + \hat{\omega}_j \log_2 \omega_j\right]$, where λ is the hyper-parameter balancing the relative contribution of the norm penalty term Ω and the loss function E, and θ denotes the network parameters. f is the activation function, and the network can be initialized by Xavier [35].

Algorithm 1 Development of NN-Based Digital Model

Require: Depth l of the network, weight $\boldsymbol{W}^i$ and bias $\boldsymbol{b}^i$ of the i^{th} layer of the network, $i = 1, 2, \ldots, l$, input $\boldsymbol{\delta}$, and output $\boldsymbol{\omega}$

(1) Signal feedforward

$\boldsymbol{h}^0 = \boldsymbol{\delta}$
for $k = 1, 2, \ldots, l$ **do**
  $\boldsymbol{a}^k = \boldsymbol{b}^k + \boldsymbol{W}^k \boldsymbol{h}^{k-1}$
  $\boldsymbol{h}^k = f(\boldsymbol{a}^k)$
end for
$\hat{\boldsymbol{\omega}} = \boldsymbol{h}^l$
$J = E(\hat{\boldsymbol{\omega}}, \boldsymbol{\omega}) + \lambda\Omega(\theta)$

(2) Error backpropagation

$\boldsymbol{g} \leftarrow \nabla_{\hat{\boldsymbol{\omega}}} J = \nabla_{\hat{\boldsymbol{\omega}}} E(\hat{\boldsymbol{\omega}}, \boldsymbol{\omega})$
for $k = l, l-1, \ldots, 1$ **do**
  $\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{a}^k} J = \boldsymbol{g} \odot f'\left(\boldsymbol{a}^k\right)$
  $\nabla_{\boldsymbol{b}^k} J = \boldsymbol{g} + \lambda \nabla_{\boldsymbol{b}^k} \Omega(\theta)$
  $\nabla_{\boldsymbol{W}^k} J = \boldsymbol{g}\left(\boldsymbol{h}^{k-1}\right)^T + \lambda \nabla_{\boldsymbol{W}^k} \Omega(\theta)$
  $\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{h}^{k-1}} J = \left(\boldsymbol{W}^k\right)^T \boldsymbol{g}$
end for

(3) Termination condition

$t = 0$
while $\Delta J > \varepsilon$ **do**
  $\boldsymbol{b}_{t+1} = \boldsymbol{b}_t - \eta \nabla_{\boldsymbol{b}_t} J$
  $\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \eta \nabla_{\boldsymbol{W}_t} J$
  $\Delta J = J_{t+1} - J_t$
  $t = t + 1$
end while

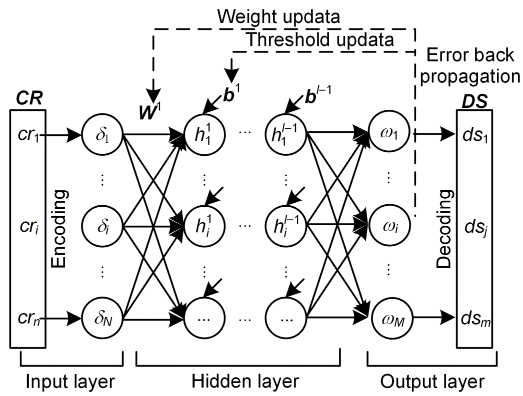


FIGURE 4. NN-based digital model.

The virtual data can be obtained by the digital model with the unconstrained input (***CR***). The virtual data is simulated based on the real-world data. Ω^i_{CR} can be divided into K_i components by clustering cr_i, and each cluster center is denoted as $cr^j_i, j = 1, 2, \ldots, K_i$. Hence, the input cr'_i of digital model could be random one out of $\{cr^j_i | i = 1, 2, \ldots, n, j = 1, 2, \ldots, K_i\}$. Digital model is able to simulate a total of $\prod_{i=1}^{n} K_i$ "$\boldsymbol{CR} - \boldsymbol{DS}$" data.

B. BEHAVIOR MODEL

The behavior model is driven by the virtual-reality integration dataset, which is composed of virtual data, real-world data, and noisy data. Real-world data needs reduction to avoid data redundancy. Mean-shift algorithm is used for data reduction, without initializing the number of clusters [36], as shown in Algorithm 2.

In Algorithm 2, bandwidth $\sigma > 0$, a distance function $d(\cdot,\cdot)$ is applicable to any pair of points, and a threshold $\varepsilon > 0$ is larger than the diameter of each component but smaller than the distance between any two components. With the Gaussian kernel, $\{\boldsymbol{CR}_i\}_{i=1}^N$ can be clustered into K components. The representative of component k is $\boldsymbol{c}_k$, which is a point in that component. Hence, N real-world data is reduced into K data. Noisy data can improve the robustness of the behavior model, by selecting some non-clustering center points from the real-world data reduction.

The behavior model is characterized as explainability. The behavior model in this paper is based on the decision tree, decomposing the complex decision-making process into a

Algorithm 2 Real-World Data Reduction

Require: $\{\boldsymbol{CR}_i\}_{i=1}^{N} \in \mathbb{R}^{N\times n}$
for $i = 1, 2, \ldots, N$ **do**
  $\boldsymbol{x} \leftarrow \boldsymbol{CR}_i$
  repeat
    $\forall i : p(\boldsymbol{x}) \leftarrow \dfrac{\exp\left(-\frac{1}{2}\left\|(\boldsymbol{x}-\boldsymbol{CR}_i)/\sigma\right\|^2\right)}{\sum_{i'=1}^{N}\exp\left(-\frac{1}{2}\left\|(x-FR_{i'})/\sigma\right\|^2\right)}$
    $\boldsymbol{x} \leftarrow \sum_{i=1}^{N} p(i|\boldsymbol{x})\boldsymbol{CR}_i$
  until stop
  $z_i \leftarrow \boldsymbol{x}$
end for
$K \leftarrow 1, \boldsymbol{c}_1 \leftarrow z_1$
for $i = 2, 3, \ldots, N$ **do**
  for $k = 1, 2, \ldots, K$ **do**
    if $d\,(z_i, \boldsymbol{c}_k) < \varepsilon$ **then**
      assign z_i to component k; **break**
    end if
  end for
  if z_i is not assigned **then**
    $K \leftarrow K+1, \boldsymbol{c}_k \leftarrow z_i$
  end if
end for
return $K, \{\boldsymbol{c}_k\}_{k=1}^{K}$

set of rules understood by natural language. C4.5 is used to develop the behavior model, which is probably the best-known tree algorithm by employing an entropy-based criterion [37], called information gain ratio, in order to select the best attribute to create a node in the tree.

Behavior model is composed by m trees, where the corresponding training sets are $S^j = \{\boldsymbol{CR}_i - ds_j\}_{i=1}^{N}$, $j = 1, 2, \ldots, m$. The attributes values of ds_j can be regarded as the labels of $\boldsymbol{CR}$. There are N "$\boldsymbol{CR}-\boldsymbol{DS}$" data in the virtual-reality integration dataset S.

$Tree^j$ is constructed based on C4.5 algorithm and dataset $S^j, j = 1, 2, \ldots, m$. The entropy of S^j is given by

$$Entropy(S^j) = -\sum_{c=1}^{l} -p_c \log_2 p_c, \tag{3}$$

where p_c is the proportion of S^j associated with the c^{th} label and l is the total number of labels.

The information gain of cr_i in S^j is defined as

$$Gain(S^j, cr_i) = Entropy(S^j) - \sum_{v=1}^{T} \frac{\left|S_v^j\right|}{\left|S^j\right|} Entropy(S_v^j), \tag{4}$$

where T is the number of different values in the domain of cr_i and S^j is then divided into T subsets. S_v^j is the v^{th} subset of S^j and $|\cdot|$ is the number of data in the dataset.

The information gain ratio of cr_i is equal to the ratio of information gain ($Gain$) to split information ($SplitInform$):

$$GainRatio(S^j, cr_i) = \frac{Gain(S^j, cr_i)}{SplitInform(S^j, cr_i)}. \tag{5}$$

$SplitInform(S^j, cr_i)$ is a penalty for cr_i that divide S^j into very small subsets to improve generalization ability of the C4.5 algorithm, and is given by

$$SplitInform(S^j, cr_i) = \sum_{v=1}^{T} -\frac{\left|S_v^j\right|}{\left|S^j\right|} \cdot \log_2 \frac{\left|S_v^j\right|}{\left|S^j\right|}. \tag{6}$$

The development of the behavior model, denoted as $M_{initial}$, is shown in Algorithm 3, where $M_{initial}$ is composed by m trees ($Tree^j, j = 1, 2, \ldots, m$). Each tree is constructed by C4.5 function. By selecting the best attribute cr_{best} to create a node in the tree, whose $GainRatio$ value is maximum, $Tree^j$ can be constructed by recursion until $S_v^j = \varnothing$.

C. EVOLUTION MODEL

The evolution model is developed for DTRC self-reinforcement, driven by the preprocessed physical PLC data, called the sequential constraint dataset. By monitoring PE data, possible problems in RC and unreasonable design specifications would be found to promote the next new customer requirements conversion. For example, the change of force over time in a certain part of the product in the usage stage of PLC can be monitored by force sensors, then the conversion rules related to this force (specific design specifications) are corrected by DTRC.

The evolution model in this paper is based on the genetic algorithm (GA), providing a possibility to seek the optimal strategy for DTRC, as shown in Fig. 5. The sequential constraint dataset is transmitted batch to batch over time, consistent with the evolution of population from generation

Algorithm 3 Development of Tree-Based Behavior Model

Require: Virtual-reality integration dataset S
(1) $\boldsymbol{M}_{\textbf{initial}}$ **modeling**
for $j = 1, 2, \ldots, m$ **do**
  $S^j \leftarrow$ Discretization of continuous variables in S^j
  $Tree^j \leftarrow C4.5(S^j)$
end for
$M_{initial} = \{Tree^j | j = 1, 2, \ldots, m\}$
(2) Function C4.5($\boldsymbol{S^j}$**)**
$cr_{best} = \left\{ cr_i \,\middle|\, \max_{cr_i \in S^j} \left(GainRatio(S^j, cr_i)\right) \right\}$
$Tree^j \leftarrow$ Create a decision node that tests cr_{best} in the root
$S_v^j \leftarrow$ Induced sub-dataset from S^j based on cr_{best}
for all S_v^j **do**
  $Tree_v^j = C4.5(S_v^j)$
  Attach $Tree_v^j$ to the corresponding branch of $Tree^j$
end for
return $Tree^j$

to generation in GA [38]. DTRC is ensured to keep evolving with the evolution model and the updating sequential constraint dataset. The key steps of the evolution model are as follows.

- *Initialization and encoding*
For $Tree^j$ ($j = 1, 2, \ldots, m$) in $M_{initial}$, the initial population is generated using N-fold cross validation by the virtual-reality integration dataset S. Individual in the population is the RC rules set $\{Rule_i\}_{i=1}^{R}$ generated from one tree. Rule is denoted as " IF A_1 and A_2 and ... and A_n, THEN C," where $A_i = \{cr_i = value\}$ is the judgment and $C = \{ds_j = value\}$ is the output. The encoding of $Rule$ is $\{x_i\}_{i=1}^{n+1}$, where x is $value$ in $Rule$. If A_i does not exist, $x_i = *$. Parameters used in the evolution model are summarized in Table 4.
- *Fitness evaluation*
Genetic fitness is calculated from two domains: the accuracy and the complexity of individual, which is given by

$$Fitness = w_1\tau_1 + \frac{1}{w_2\tau_2 + w_3\tau_3}, \quad (7)$$

where w_1, w_2, and w_3 are the corresponding impact factor, $w_1 + w_2 + w_3 = 1$, and $0 < w_1, w_2, w_3 < 1$.
τ_1 is the accuracy of individual to the sequential constrained dataset,

$$\tau_1 = \frac{Number\ of\ correctly\ predicted\ cases}{Number\ of\ cases\ in\ sequential\ constrained\ dataset}. \quad (8)$$

τ_2 denotes the complexity of chromosome of individual,

$$\tau_2 = \frac{Number\ of\ chromosomes\ in\ individual}{Maximum\ number\ of\ chromosomes\ in\ population}. \quad (9)$$

τ_3 denotes the complexity of gene of individual,

$$\tau_3 = \frac{Number\ of\ gene\ in\ individual}{Maximum\ number\ of\ gene\ in\ population}. \quad (10)$$

- *Genetic operators*
    1) Selection: During each successive generation, the probability of an individual being selected to breed a new generation is $p_i = Fitness_i / \sum_{i=1}^{N} Fitness_i$. There exist N individuals in each generation.
    2) Crossover: Genetic information located in the same loci is swapped between the two chromosomes in one individual.
    3) Mutation: Generating a random variable, including deletion, for a specific loci in a chromosome.
    4) Recombination: Chromosomes with the same phenotype and the same genetic information except one loci t are recombined into a new chromosome with $x_t = *$, meeting the following condition: x_t of these chromosomes covers all possible values or x_t of one of these chromosomes equals $*$.

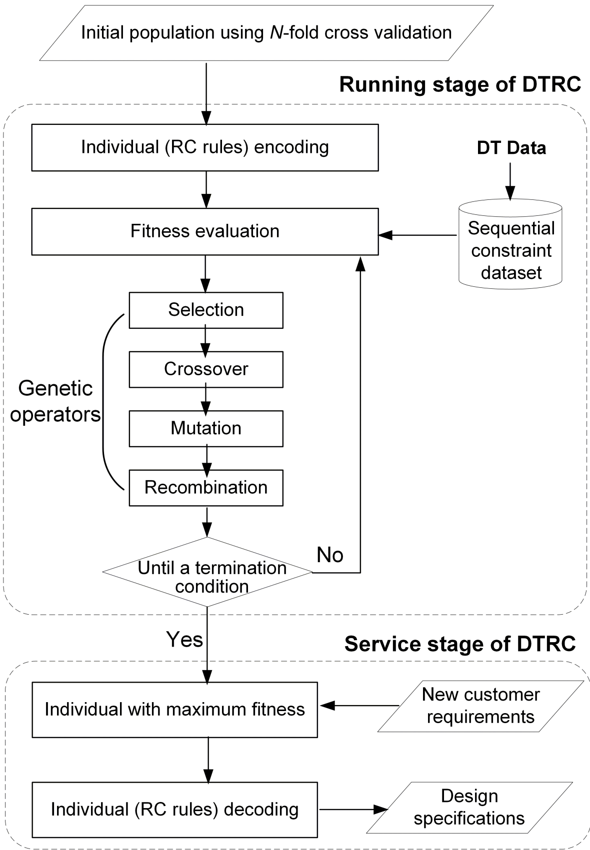


FIGURE 5. GA-based evolution model.

TABLE 4. Parameters used in evolution model.

Parameter	Description
Population	N trees
Individual	RC rules set generated from one tree
Chromosome	$Rule$ from RC rules set
Gene	Judgment A in $Rule$
Genetic information	x_i in $Rule$ encoding, $i = 1, 2, \ldots, n$
Phenotype	Output C in $Rule$, as same as x_{n+1} in $Rule$ encoding
Loci	Subscript of A_i in $Rule$, as same as subscript of x_i in $Rule$ encoding
Deletion	$*$ in $Rule$ encoding

- *Termination condition*
PE data from PLC is time-varying as same as the sequential constrained dataset is updated over time, with the fitness of each individual in the population changing accordingly. The change of the sequential constrained dataset is also beneficial to jump out of the local optimum during the evolution process. This evolution process of DTRC is repeated until a termination condition has been reached. Terminating conditions of the evolution model are:
    1) Fixed number of generations reached.
    2) The highest-ranking solution's fitness is reaching or has reached a plateau such that successive iterations no longer produce better results.
    3) DTRC provides external services to stakeholders.

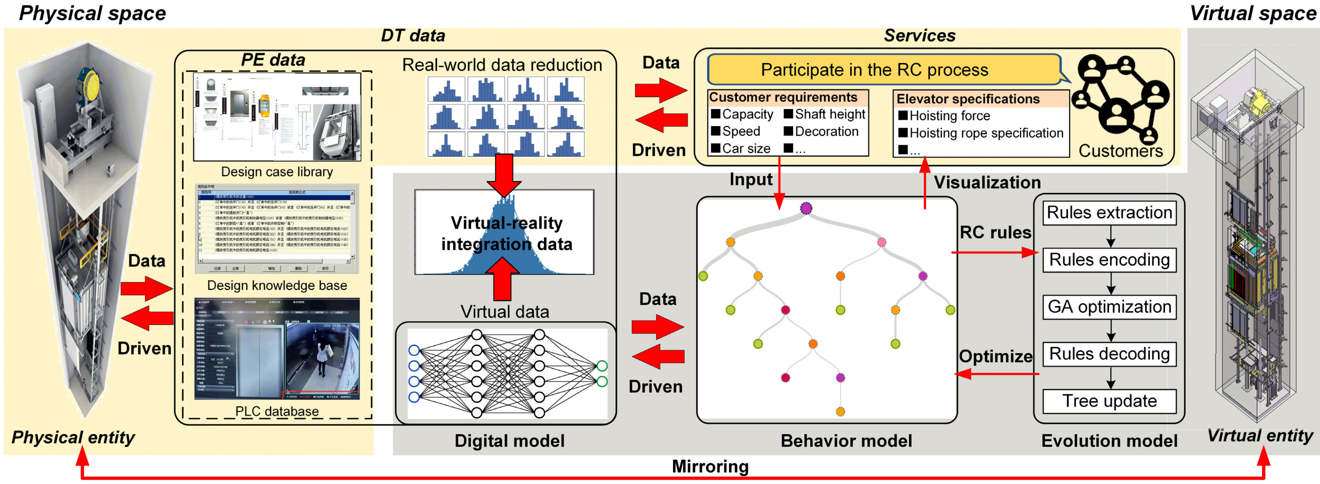


FIGURE 6. DTRC for elevator customized design.

V. CASE STUDY: DTRC FOR ELEVATOR CUSTOMIZATION

Elevators are highly customized industrial products for specific modern buildings. Increasing demands and intense market competition for various elevators call for elevator manufacturers more intelligent in design, more flexible in manufacturing, and more personalized in service [4]. During the design stage in elevator customization, DTRC is employed to facilitate the conversion from new customer requirements to elevator design specifications, meanwhile addressing the existing limitations in RC. As shown in Fig. 6, DTRC for elevator customization is developed by the mirroring between PE and VE, including DD, Ss, and CN. PE data in DD includes the design knowledge base, design cases library, and PLC database. The digital model simulates "$\boldsymbol{CR}-\boldsymbol{DS}$" virtual data, compensating for the absence of real-word "$\boldsymbol{CR}-\boldsymbol{DS}$" data extracted from design cases library. The behavior model is driven by the virtual-reality integration "$\boldsymbol{CR}-\boldsymbol{DS}$" data, which visualizes physical RC behavior and promotes customer participation. The evolution model is driven by the preprocessed PLC data, which reflects the real-time conditions throughout the elevator life cycle and enables DTRC to optimize RC rules and results in the future. Development of DTRC can be outlined in three stages, installation, running, and service, detailed in the subsections below.

A. INSTALLATION STAGE OF DTRC

The real-world "$\boldsymbol{CR}-\boldsymbol{DS}$" data is extracted from the design case library in an elevator manufacturer in China, with a total of 19 939 valuable data. Due to the complex dependency relationships within and between customer requirements and design specifications in real-world RC for elevator customized design, this case study simplifies the RC parameters to derive the theoretical aspects in a logical manner [13]. The simplified customer requirements for RC have five dimensions, shaft height, capacity, speed, car width, and car depth. Design specifications in this case study take hoisting force (continuous variable) and hoisting rope specification (categorical variable) as examples. Hoisting rope specification is standardized by GB8903–2005 [39], with five common class labels in this case study, 9.5NAT8 × 25Fi+FC, 10NAT8 × 19S+FC, 12.7NAT8 × 19S+FC, 16NAT8 × 19S+IWR, and 17.5NAT8× 19W+FC.

An NN-based digital model is developed for each design specification. The numbers of neurons in input and output layers are 5 and 1, respectively, for hoisting force, whereas 5 and 3 neurons in input and output layers are set for hoisting rope specification. The number of hidden layers is 2. The digital model is trained by the real-world "$\boldsymbol{CR}-\boldsymbol{DS}$" dataset with 10-fold cross validation. Fig. 7 presents the influence of the different numbers of neurons in the hidden layers on the accuracy of virtual data, and (11) describes the accuracy of the simulation by root-mean-square error (RMSE). The smaller the RMSE value is, the more similar the virtual data is to the real-world data.

$$\begin{cases} RMSE_0 = \sqrt{\dfrac{1}{|M|}\displaystyle\sum_{i=1}^{|M|}\left(\widehat{ds}_i - ds_i\right)^2}, \\ RMSE_1 = \sqrt{\dfrac{1}{|M|}\displaystyle\sum_{i=1}^{|M|} 1_{\{\widehat{ds}_i \neq ds_i\}}}, \end{cases} \tag{11}$$

where M is the validation set to the digital model. ds in RMSE_0 and RMSE_1 are continuous and categorical variables, respectively. $\widehat{ds}$ is the simulated virtual data, while ds is the real-world data.

As shown in Fig. 7, the red rectangle area is the parameters selected for the numbers of neurons in hidden layers, while ensuring the accurate simulation and the acceptable training time. The selected topologies of digital model for hoisting force and hoisting rope specification are "5–10–10–1" and "5–100-100–3", respectively, with $\mathrm{RMES}_0 = 0.1445$ for hoisting force and $\mathrm{RMES}_1 = 0.1632$ for hoisting rope specification.

Virtual data in S is simulated by the distribution of real-world $\boldsymbol{CR}$ data, as shown in Fig. 8. The input of the digital

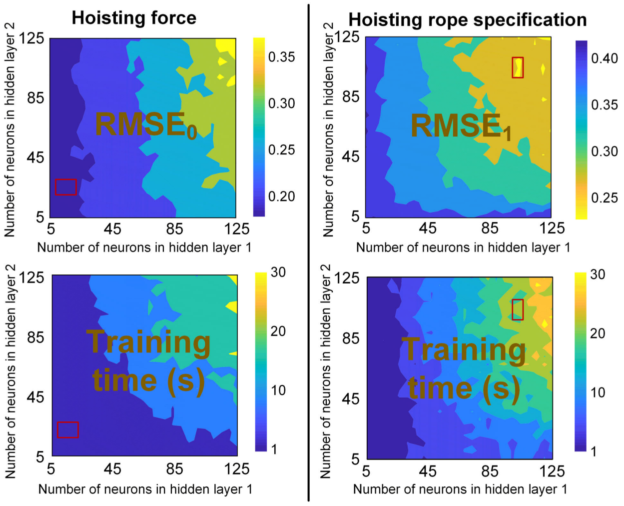


FIGURE 7. Parameters selection for digital model.

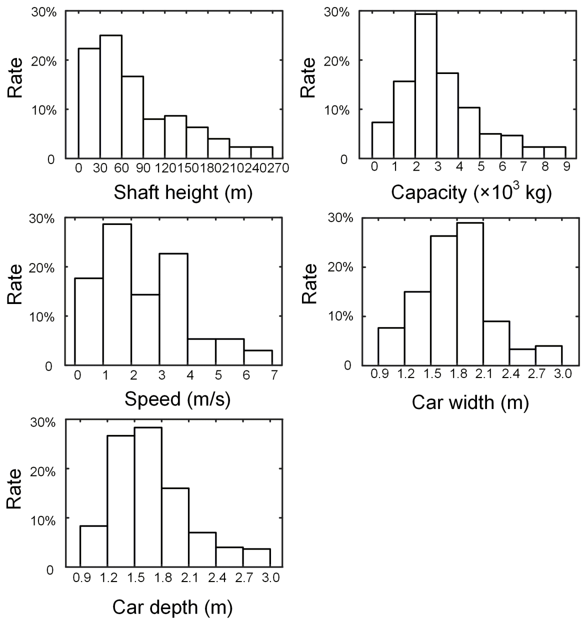


FIGURE 8. Distribution of each attribute in customer requirements.

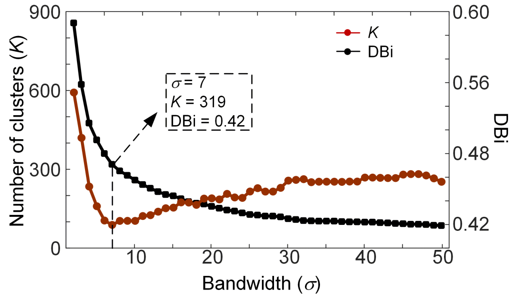
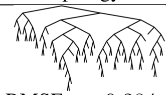
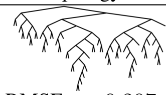
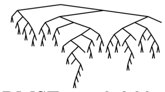
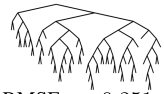
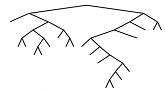
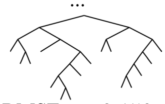


FIGURE 9. Real-world data reduction by a mean-shift algorithm.

TABLE 5. Evolution process of DTRC.

Hosting force		Hosting rope specification	
Epoch	Topology	Epoch	Topology
0	 $\mathrm{RMSE}_0 = 0.284$	0	 $\mathrm{RMSE}_1 = 0.297$
1	 $\mathrm{RMSE}_0 = 0.272$	1	 $\mathrm{RMSE}_1 = 0.268$
2	 $\mathrm{RMSE}_0 = 0.251$	2	 $\mathrm{RMSE}_1 = 0.237$
...	...	...	...
22	 $\mathrm{RMSE}_0 = 0.195$	19	 $\mathrm{RMSE}_1 = 0.196$
...	...	...	...
50	 $\mathrm{RMSE}_0 = 0.135$	32	 $\mathrm{RMSE}_1 = 0.164$
...	...	...	...
60	 $\mathrm{RMSE}_0 = 0.128$	43	 $\mathrm{RMSE}_1 = 0.159$

model covers every cluster of each attribute in ***CR***. Digital model can simulate 27 783 virtual "$\boldsymbol{CR}-\boldsymbol{DS}$" data.

Real-world data reduction in S with different bandwidth σ is shown in Fig. 9. As σ increases, the number of clusters decreases. Clustering is evaluated by the Davies-Bouldin index (DBi). There are 319 real-world reduced data with the smallest DBi value when $\sigma = 7$.

Noisy data in S is obtained by randomly selecting 98 non-clustering center points from the real-world data reduction. The behavior model is driven by S with 28 200 virtual-reality integration "$\boldsymbol{CR}-\boldsymbol{DS}$" data. $M_{initial}$ developed by the behavior model is simplified as the trees of Epoch 0 in Table 5.

B. RUNNING STAGE OF DTRC

The evolution model is operating in the running stage of DTRC, which is driven by the sequential constrained data preprocessed by monitoring the operation and maintenance conditions of 291 elevators within one month. The evolution process of DTRC is summarized in Table 5. The initial trees from the behavior model have complex topologies and are inaccurate in predicting sequential constrained data. GA in the evolution process considers the accuracy and complexity of trees at the same time, pruning the trees partly. RMSE_0 for hoisting force decreases from 0.284 in Epoch 0 to 0.128 in Epoch 60, with a reduction in RMSE of 54.9%. RMSE_1 for hoisting rope specification decreases from 0.297 in Epoch 0 to 0.159 in Epoch 43, with a reduction in RMSE of 46.5%. Updating RC rules with the time-varying sequential constrained data obtained from real-world PLM, DTRC enables the co-evolution between PE and VE by feeding the improved RC rules into PE.

To test the performance of the proposed tri-model-based DTRC, we set up a comparative experiment to transform customer elevator requirements into two design specifications.

TABLE 6. Performance of different methods for RC.

Model	Method	RMSE	
		Hosting force	Hosting rope specification
Uni-model-based	NN	0.146	0.163
	Tree	0.306	0.376
Du-model-based	NN+Tree	0.164	0.191
	Tree+GA	0.284	0.297
Tri-model-based	NN+Tree+GA	**0.128**	**0.159**

The compared methods include uni-model-based RC, du-model-based RC, and tri-model-based DTRC. Uni-model-based RC is driven by the single model, NN or Tree, whereas du-model-based RC combines two basic models, NN+Tree or Tree+GA. Tri-model-based DTRC integrates digital model, behavior model, and evolution model, which is denoted as NN+Tree+GA.

The experimental dataset is the real-world "$\boldsymbol{CR-DS}$" data and sequential constrained data, where 100 "$\boldsymbol{CR-DS}$" data are randomly sampled for validation and the rest data is used for running compared methods. The evaluation metrics used in the experiment is the conversion accuracy (RMSE), as the same form as (11), where $\widehat{ds}$ is the conversion data obtained by different methods and ds is the real-world validation data. Performance of different methods for RC is shown in Table 6, where RMSE results are averaged over 10 independent realizations.

In the uni-model-based RC, NN is a traditional data-driven model for RC with higher accuracy against other single models. However, the unexplainability of NN limits customer participation, where the RC process is a black box to customers. Tree makes up for the shortcoming of NN, but the accuracy is the lowest in comparison. Hence, uni-models can be integrated and evolved for better performance in RC.

In the du-model-based RC, NN+Tree is a static model compared with DTRC. Although the accuracy of NN+Tree is acceptable, the performance will drop significantly with the changeable customer requirements and product upgrades in the future. The solution is to redevelop the RC model periodically. Tree+GA is predictable and proactive to a dynamic environment, but with low accuracy in RC. The driving force of Tree+GA is only the real-world data, without virtual data simulated by the digital model. The basic model (Tree) has deviations caused by the single-source data, and the effect of optimization (GA) is not obvious, indicating that the insufficient data inhibits the accuracy of RC models.

The tri-model-based DTRC is characterized as explainability and proactivity compared against the uni-model-based RC (NN). NN+Tree+GA outperforms other methods, with average reductions in RMSE of 57.9%, 50.7%, and 19.4% against Tree, Tree+GA, and NN+Tree.

C. SERVICE STAGE OF DTRC

To assist stakeholders in executing RC in elevator customized design, DTRC is configured as a module in a prototype system (see Fig. 10) created by Java, MySQL, and Python. DTRC is a tool for conversion in requirement modeling, where visualization, interaction, and optimization engines provide services of DTRC.

As shown in Fig. 10, the real-world design specifications for the customized elevator are various. Taking the conversion from customer requirements to hoisting rope specification as an example, the right area of Fig. 10 visualizes the RC rules. The input new customer requirements are 21 m shaft height, 2000 kg capacity, 1.02 m/s speed, 1.55 m car width, and 2.03 m car depth. The conversion path is highlighted in the RC tree, where customer can instinctively understand each conversion step from requirements to specification. The conversion path is also shown in the user interface. The conversion result for hoisting rope specification is 12.7NAT8 $\times$ 19S+FC, which is viable to this elevator customized design evaluated by designers. Meanwhile, the conversion rules are optimized and updated continuously in DTRC. This demonstrative case highlights the feasibility of the proposed DTRC and promotes customer participation, which is powerful in addressing major challenges for RC in elevator customized design.

D. DISCUSSIONS

This case study demonstrates the feasibility of the proposed DTRC in facilitating RC in smart customized design. The RC results are optimized by the sequential constrained dataset, in the other words, the big data throughout PLC is applied to optimize the important activity in the customized design. DTRC is helpful to deepen designers' understandings of the expected and real worlds in the design process. For instance, designers would compare the differences of hoisting force between the actual operation and the design specifications, to improve the RC rules. Meanwhile, customers can participate in the RC process via DTRC, which is an important characteristic of smart customization. Customers can clarify their own requirements by understanding the RC steps and results, and interact directly and independently with the designers to improve the satisfaction with the final design scheme.

The ability to respond to real-time developments in physical space is not essential to DTRC compared against other DT applications in the manufacturing or usage stages of PLC. For instances, the hoisting force of an elevator can be monitored and stored by force sensors. However, an elevator is not a continuous running industrial product. The hoisting force of an elevator is time-varying depending on different load and load distribution each time it runs. The sequential constrained dataset for optimizing hoisting force in RC, constructed by the peak value from the monitoring force values for one week, is reasonable. Hence, the real-time capability of PE data transmission is not strict but necessary to DTRC.

Despite those advantages, the proposed DTRC still has some limitations. For instance, the design specifications in elevator customized design are assumed as independent. The behavior model in DTRC can be advanced by other visualization models, considering the dependency relationships within design specifications. Meanwhile, an elevator, a special industrial product, occurs with the specific building. The

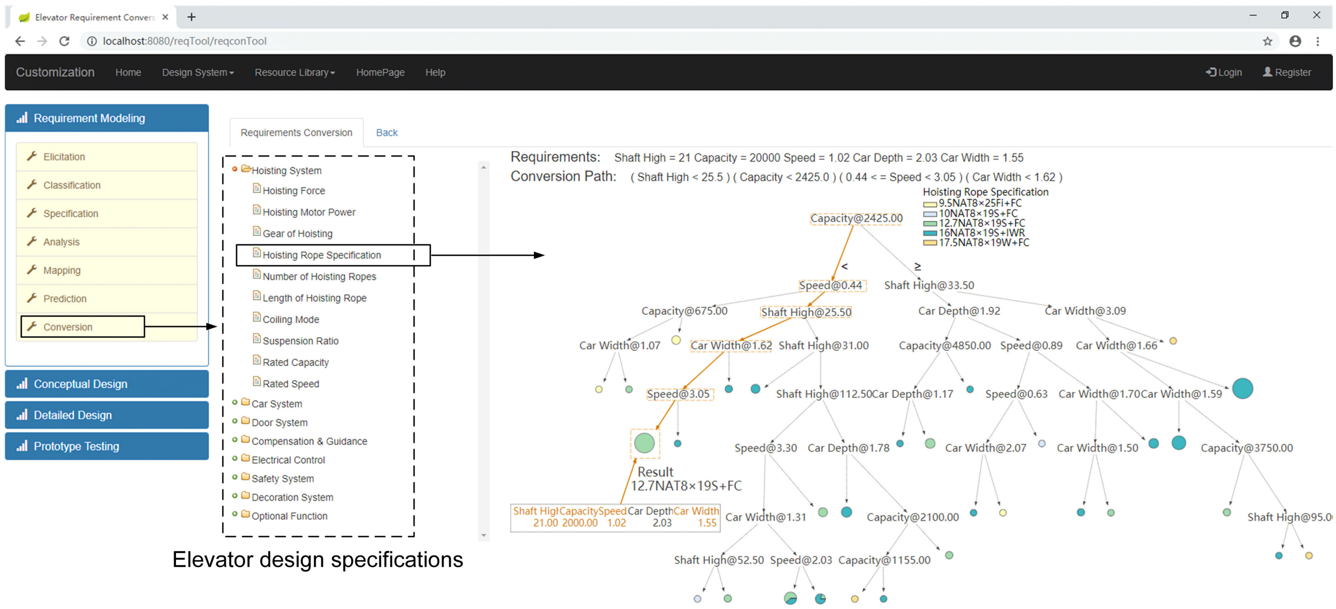


FIGURE 10. User interface for DTRC.

environment model, high-fidelity replication of the building environment (e.g., building purposes, passenger flow, weather, and passenger identities), is considered in some DT applications [4], [40], [41]. Although these DTs are mostly applied in the usage stage of the elevator life cycle (e.g., noise control and energy efficiency service), it is worth thinking about integrating environment models in DTRC in the design stage.

VI. CONCLUSION

DTRC is proposed to address the major challenges for RC in product customized design, including single-source driving force, passivity to diversified requirements, and limitation to customer participation. The novelty and innovation of the paper concentrate on the following aspects:

1) To the best of authors' knowledge, it is the first time to enhance RC with DT in product customized design. A comprehensive DTRC framework is detailed introduced with the five-dimension structure, i.e., PE, VE, DD, Ss, and CN.
2) A tri-model-based approach for DTRC development is proposed. To solve the single-source driving force, NN-based digital model simulates the virtual data to compensate for the absence of real-world data. To solve the limitation to customer participation, tree-based behavior model visualizes the RC behavior and rules via virtual-reality integration data. To solve passivity to diversified requirements, GA-based evolution model optimizes the RC rules constrained by the preprocessed physical PLC data. These three models can be integrated to solve the existing challenges in RC.
3) A case study for RC in elevator customized design validates the feasibility and effectiveness of DTRC. Experimental results show that DTRC outperforms other model-based RC approaches in terms of conversion accuracy, DTRC visualizes the conversion path oriented to customers, and DTRC is proactive to the dynamic customer requirements via co-evolution between PE and VE.

However, several limitations of DTRC should be considered as well. Firstly, the tri-model can be advanced to improve the accuracy of RC. Secondly, DTRC aims at the single design specification without considering the dependency relationships within specifications. Finally, preprocessing methods of PLC data applied to optimize the RC rules can be pursued in future work.

REFERENCES

[1] S. Zhang, J. Xu, H. Gou, and J. Tan, "A research review on the key technologies of intelligent design for customized products," *Engineering*, vol. 3, no. 5, pp. 631–640, Oct. 2017.
[2] F. Tao, Q. Qi, A. Liu, and A. Kusiak, "Data-driven smart manufacturing," *J. Manuf. Syst.*, vol. 48, pp. 157–169, Jul. 2018.
[3] C. Zhang, D. Chen, F. Tao, and A. Liu, "Data driven smart customization," *Procedia CIRP*, vol. 81, pp. 564–569, Jun. 2019, doi: 10.1016/j.procir.2019.03.156.
[4] X. Wang, Y. Wang, F. Tao, and A. Liu, "New paradigm of data-driven smart customisation through digital twin," *J. Manuf. Syst.*, vol. 58, pp. 270–280, Jan. 2021.
[5] Y. Gu, S. Zhang, L. Qiu, Z. Wang, and L. Zhang, "A layered KNN-SVM approach to predict missing values of functional requirements in product customization," *Appl. Sci.*, vol. 11, no. 5, p. 2420, Mar. 2021.
[6] W. Song, "Requirement management for product-service systems: Status review and future trends," *Comput. Ind.*, vol. 85, pp. 11–22, Feb. 2017.
[7] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital twin in industry: State-of-the-art," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2405–2415, Apr. 2019.
[8] Y. Yin, K. E. Stecke, and D. Li, "The evolution of production systems from industry 2.0 through industry 4.0," *Int. J. Prod. Res.*, vol. 56, nos. 1–2, pp. 848–861, Jan. 2018.
[9] X.-Y. Shao, Z.-H. Wang, P.-G. Li, and C.-X.-J. Feng, "Integrating data mining and rough set for customer group-based discovery of product configuration rules," *Int. J. Prod. Res.*, vol. 44, no. 14, pp. 2789–2811, Jul. 2006.

[10] X. Geng, X. Chu, D. Xue, and Z. Zhang, ''An integrated approach for rating engineering characteristics' final importance in product-service system development,'' *Comput. Ind. Eng.*, vol. 59, no. 4, pp. 585–594, Nov. 2010.

[11] C. Lee, B. Song, and Y. Park, ''Design of convergent product concepts based on functionality: An association rule mining and decision tree approach,'' *Expert Syst. Appl.*, vol. 39, no. 10, pp. 9534–9542, Aug. 2012.

[12] K.-C. Wang, ''A hybrid Kansei engineering design expert system based on grey system theory and support vector regression,'' *Expert Syst. Appl.*, vol. 38, no. 7, pp. 8738–8750, Jul. 2011.

[13] L. Yu, L. Wang, and J. Yu, ''Identification of product definition patterns in mass customization using a learning-based hybrid approach,'' *Int. J. Adv. Manuf. Technol.*, vol. 38, nos. 11–12, pp. 1061–1074, Oct. 2008.

[14] W. Song, X. Ming, and Y. Han, ''Prioritising technical attributes in QFD under vague environment: A rough-grey relational analysis approach,'' *Int. J. Prod. Res.*, vol. 52, no. 18, pp. 5528–5545, May 2014.

[15] X.-J. Ma, G.-F. Ding, S.-F. Qin, R. Li, K.-Y. Yan, S.-N. Xiao, and G.-W. Yang, ''Transforming multidisciplinary customer requirements to product design specifications,'' *Chin. J. Mech. Eng.*, vol. 30, no. 5, pp. 1069–1080, Sep. 2017.

[16] T. Beernaert and L. Etman, ''Multi-level decomposed systems design: Converting a requirement specification into an optimization problem,'' in *Proc. Design Soc., Int. Conf. Eng. Design*, Delft, The Netherlands, 2019, pp. 3691–3700.

[17] X. Li, C.-H. Chen, P. Zheng, Z. Wang, Z. Jiang, and Z. Jiang, ''A knowledge graph-aided concept–knowledge approach for evolutionary smart product–service system development,'' *J. Mech. Des.*, vol. 142, no. 10, Oct. 2020, Art. no. 101403.

[18] J. Cong, C.-H. Chen, and P. Zheng, ''Design entropy theory: A new design methodology for smart PSS development,'' *Adv. Eng. Informat.*, vol. 45, Aug. 2020, Art. no. 101124.

[19] Q. Qi and F. Tao, ''Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison,'' *IEEE Access*, vol. 6, pp. 3585–3593, Jan. 2018.

[20] J. Stark, *Product Lifecycle Management*. Cham, Switzerland: Springer, 2016.

[21] K. Y. H. Lim, P. Zheng, and C.-H. Chen, ''A state-of-the-art survey of digital twin: Techniques, engineering product lifecycle management and business innovation perspectives,'' *J. Intell. Manuf.*, vol. 31, pp. 1313–1337, Aug. 2020.

[22] F. Tao, F. Sui, A. Liu, Q. Qi, M. Zhang, B. Song, Z. Guo, S. C.-Y. Lu, and A. Nee, ''Digital twin-driven product design framework,'' *Int. J. Prod. Res.*, vol. 57, no. 12, pp. 3935–3953, 2019.

[23] J. Guo, N. Zhao, L. Sun, and S. Zhang, ''Modular based flexible digital twin for factory design,'' *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 3, pp. 1189–1200, Mar. 2019.

[24] J. Leng, D. Yan, Q. Liu, H. Zhang, G. Zhao, L. Wei, D. Zhang, A. Yu, and X. Chen, ''Digital twin-driven joint optimisation of packing and storage assignment in large-scale automated high-rise warehouse product-service system,'' *Int. J. Comput. Integr. Manuf.*, pp. 1–18, Sep. 2019, doi: 10.1080/0951192X.2019.1667032.

[25] Q. Liu, J. Leng, D. Yan, D. Zhang, L. Wei, A. Yu, R. Zhao, H. Zhang, and X. Chen, ''Digital twin-based designing of the configuration, motion, control, and optimization model of a flow-type smart manufacturing system,'' *J. Manuf. Syst.*, vol. 58, pp. 52–64, Jan. 2021.

[26] Q. Liu, H. Zhang, J. Leng, and X. Chen, ''Digital twin-driven rapid individualised designing of automated flow-shop manufacturing system,'' *Int. J. Prod. Res.*, vol. 57, no. 12, pp. 3903–3919, May 2018.

[27] B. Schleich, N. Anwer, L. Mathieu, and S. Wartzack, ''Shaping the digital twin for design and production engineering,'' *CIRP Ann.*, vol. 66, no. 1, pp. 141–144, Jan. 2017.

[28] J. Dias-Ferreira, L. Ribeiro, H. Akillioglu, P. Neves, and M. Onori, ''BIOSOARM: A bio-inspired self-organising architecture for manufacturing cyber-physical shopfloors,'' *J. Intell. Manuf.*, vol. 29, no. 7, pp. 1659–1682, Oct. 2018.

[29] K. Y. H. Lim, P. Zheng, C.-H. Chen, and L. Huang, ''A digital twin-enhanced system for engineering product family design and optimization,'' *J. Manuf. Syst.*, vol. 57, pp. 82–93, Oct. 2020.

[30] J. Leng, D. Yan, Q. Liu, K. Xu, J. Zhao, R. Shi, L. Wei, D. Zhang, and X. Chen, ''Manuchain: Combining permissioned blockchain with a holistic optimization model as bi-level intelligence for smart manufacturing,'' *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 1, pp. 182–192, Jan. 2020.

[31] P. Zheng, T.-J. Lin, C.-H. Chen, and X. Xu, ''A systematic design approach for service innovation of smart product-service systems,'' *J. Cleaner Prod.*, vol. 201, pp. 657–667, Nov. 2018.

[32] P. Zheng, X. Xu, and C.-H. Chen, ''A data-driven cyber-physical approach for personalised smart, connected product co-development in a cloud-based environment,'' *J. Intell. Manuf.*, vol. 31, no. 1, pp. 3–18, Jan. 2020.

[33] B. Stackpole. (2015). *Digital Twins Land a Role in Product Design*. [Online]. Available: https://www.digitalengineering247.com/article/digital-twins-land-a-role-in-product-design/

[34] F. Li, J. M. Zurada, Y. Liu, and W. Wu, ''Input layer regularization of multilayer feedforward neural networks,'' *IEEE Access*, vol. 5, pp. 10979–10985, Jun. 2017.

[35] X. Glorot and Y. Bengio, ''Understanding the difficulty of training deep feedforward neural networks,'' *J. Mach. Learn. Res.*, vol. 9, pp. 249–256, May 2010.

[36] C. Hennig, M. Meila, F. Murtagh, and R. Rocci, *Handbook of Cluster Analysis*. New York, NY, USA: CRC Press, 2015.

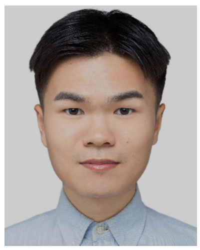
[37] J.-S. Lee, ''AUC4.5: AUC-based C4.5 decision tree algorithm for imbalanced data classification,'' *IEEE Access*, vol. 7, pp. 106034–106042, Jul. 2019.

[38] S. Jun and S. Lee, ''Learning dispatching rules for single machine scheduling with dynamic arrivals based on decision trees and feature construction,'' *Int. J. Prod. Res.*, pp. 1–19, Mar. 2020, doi: 10.1080/00207543.2020.1741716.

[39] *Steel Wire Ropes for Elevators*, document GB 8903–2005, 2005.

[40] M. M. Gichane, J. B. Byiringiro, A. K. Chesang, P. M. Nyaga, R. K. Langat, H. Smajic, and C. W. Kiiru, ''Digital triplet approach for real-time monitoring and control of an elevator security system,'' *Designs*, vol. 4, no. 2, p. 9, Apr. 2020.

[41] R. Parmar, A. Leiponen, and L. D. Thomas, ''Building an organizational digital twin,'' *Bus. Horizons*, vol. 63, no. 6, pp. 725–736, Nov. 2020.



YE GU was born in Jiangsu, China, in 1995. He received the B.S. degree from the School of Mechanical Engineering, Dalian University of Technology, in 2016. He is currently pursuing the Ph.D. degree with Zhejiang University. His research interests include product digital design and smart customization.

SHUYOU ZHANG was born in Zhejiang, China, in 1963. He received the M.S. degree from the Department of Mechanical Engineering, Zhejiang University, in 1991, and the Ph.D. degree from the State Key Laboratory of Computer-Aided Design and Computer Graphics, Zhejiang University, in 1999. He is currently a Professor with Zhejiang University. He engaged in the research work for the technology of product digital design, technology of design and stimulation for complex equipment's, engineering and computer graphics, and the key technology for enterprise information. His research interests include technology of product digital design, technology of design and stimulation for complex equipment's, engineering and computer graphics, and key technology for enterprise information. He is an Assistant Director of the Computer Graphics Professional Committee, China Engineering Graphic Society, a member of the Product Digital Design Professional Committee, and a Chairman of the Zhejiang Engineering Graphic Society.

LEMIAO QIU received the Ph.D. degree from the Department of Mechanical Engineering, Zhejiang University, Hangzhou, China, in 2008. His research interests include mechanical design theories, digital design and manufacture, and production informatization.