

Article

A PEI Simulation Method for Process Manufacturing

Xiaobin Tang ^{1,2} , Meng Yan ¹, Wenfeng Xu ¹, Gaoping Xu ^{1,*}  and Yize Sun ¹

¹ College of Mechanical Engineering, Donghua University, 2999 North Renmin Road, Songjiang District, Shanghai 201620, China; xiaobintang@dhu.edu.cn (X.T.); yanmeng0112@163.com (M.Y.); 15366552692@163.com (W.X.); sunyz@dhu.edu.cn (Y.S.)

² College of Electronic Engineering, Chengdu Technological University, No. 1, Section 2 Zhongxin Avenue, Pidu District, Chengdu 611730, China

* Correspondence: xgp@dhu.edu.cn

Abstract

In response to the growing complexity of modern process manufacturing systems, this paper proposes a novel simulation framework named the Process–Equipment–In-Process State (PEI) simulation method, which introduces a unified and structured approach to modeling multi-stage industrial processes. Unlike conventional simulation approaches that rely on ad hoc or loosely organized modules, the PEI method decomposes the simulation system into three core and interoperable modules: Process Structure (P), Equipment Behavior (E), and In-Process State (I). This modular abstraction facilitates the decoupling of model logic. It also enables a structure-driven simulation execution mechanism. In this structure, the process topology governs task scheduling; equipment models translate control inputs into physical conditions; and state models simulate material evolution accordingly. A complete simulation case involving water mixing, heat exchange, and slurry transformation demonstrates the method’s capability to support traceable state evolution, logical task flow, and extensible model binding. The results demonstrate that the proposed method enables module decoupling, clear simulation pathways, and traceable state changes, providing effective support for structured modeling and behavioral evolution analysis in process manufacturing.

Keywords: process manufacturing; PEI simulation method; structured simulation; module decoupling; process-driven simulation



Academic Editors: Dan Zhang and Iqbal M. Mujtaba

Received: 2 July 2025

Revised: 26 September 2025

Accepted: 28 September 2025

Published: 30 September 2025

Citation: Tang, X.; Yan, M.; Xu, W.; Xu, G.; Sun, Y. A PEI Simulation Method for Process Manufacturing. *Processes* **2025**, *13*, 3148. <https://doi.org/10.3390/pr13103148>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Process manufacturing is widely applied in industries such as chemicals, pharmaceuticals, food, energy, and fiber production. Its typical characteristics include continuous material flow, tightly coupled equipment, complex system structures, and strong dynamic nonlinearity [1–3]. In the face of multi-source heterogeneous devices, time-varying boundary conditions, and cross-level process logic, traditional static modeling and offline computation approaches have become inadequate for meeting the modern demands of process industries in terms of state awareness, performance prediction, and scheduling optimization [4,5].

In recent years, research in simulation and modeling for process manufacturing has shown a trend toward multi-dimensional integration, primarily in the following areas.

First, research has focused on soft sensing and state modeling technologies. To address the challenge of estimating key variables that are difficult to measure directly under real

operating conditions, researchers have proposed various data-driven state estimation methods. For example, soft sensors based on Long Short-Term Memory (LSTM) and attention mechanisms have demonstrated high prediction accuracy in industrial applications [6,7], and sequence modeling methods that integrate convolutional structures with self-attention have also shown excellent performance and generalization capabilities [8]. In addition, multi-modal, real-time updating models that fuse statistical rules and machine learning approaches have found applications in complex processes [9–11]. However, these methods mainly focus on single-point prediction and lack system-level structural integration [12].

Second, research on equipment behavior modeling and process control has also drawn significant attention. Hybrid modeling and model predictive control (MPC) approaches are widely used to construct device response mechanisms. For instance, the work by Ahmed employed MPC to enhance performance in crude oil processing systems [13], while Asensi et al. proposed a multi-variable control strategy for thermal flow control systems, improving robustness in temperature regulation [14]. Other practical cases include gas blending stations [15] and Dynamic Matrix Control (DMC) reaction systems [16]. Despite these advances in device-level control, collaborative simulation across systems remains underdeveloped.

Third, in the development of system-level process modeling platforms, mainstream industrial tools such as Simulink, Modelica, and Sim3Tanks provide modular support for industrial modeling [17,18]. The Process Analysis, Retrofit and Optimization in the Context of Control (PAROC) framework proposed by Pistikopoulos integrates modeling languages, schedulers, and optimizers to support process-control integrated modeling [19]. However, most of these platforms focus on modeling syntax or solving algorithms and lack a unified structural definition for simulation task organization and model data flow scheduling [20–22].

Fourth, the rise of digital twin (DT) technologies in the process industry has also attracted growing interest. Recent studies have explored how to integrate industrial data, modeling systems, and real-time interaction to construct digital models with mapping, prediction, and feedback capabilities [2,5,23]. For example, Mozharovskii and Shevlyagina proposed a sensor twin framework that integrates prediction and visualization [24]. Some research also focuses on coupling digital twins with controllers [25,26]. However, most existing systems are still limited to visualization or early warning functions and lack the ability to organize simulation tasks from a process modeling logic perspective [27,28].

In addition, some studies have focused on integrated modeling for specific processes, such as CO₂ capture and conversion systems [4], fiber spinning and drying systems [29], and dimethyl ether (DME) synthesis processes [30]. However, these approaches generally lack reconfigurability and the ability to support multi-module coupling, which limits their applicability across diverse industrial scenarios. Recent review articles have also emphasized that future research in process simulation should focus more on system-level modeling structures, modular coupling mechanisms, and organizational models for cross-equipment coordination [31–33].

Most existing simulation tools in process manufacturing are developed within commercial platforms such as Aspen Plus, Tecnomatix Plant Simulation, or gPROMS. These platforms are usually tailored to specific domains. For example, Aspen Plus focuses on chemical thermodynamics, Tecnomatix on discrete logistics, and gPROMS on equipment-level process control. However, they are not well suited for generalized simulation of multi-stage production processes that involve structural logic, equipment behavior, and evolving material states.

In summary, although existing studies have contributed substantial advancements in model construction, optimization algorithms, and simulation platforms, several critical

challenges remain unresolved: (1) the lack of a structured mapping between simulation tasks, equipment behavior, and state evolution; (2) the absence of a standardized mechanism for scheduling model execution and coordinating data flow across different modules; and (3) the unclear coupling between process structure hierarchies and system response logic, which hinders the scalability and reusability of simulation systems.

To address these challenges, this paper proposes a structured simulation method for process manufacturing, referred to as the PEI simulation method. This method constructs a simulation system architecture based on three fundamental modules: P (the Process Structure module), which defines simulation nodes and scheduling sequences; E (the Equipment Behavior module), which models the input–output response logic of equipment; and I (the In-Process State module), which simulates the evolution of state variables.

These modules are organized in a modular and schedulable manner, where simulation tasks are driven by the process structure and executed in a step-by-step sequence. This structure enables clear task orchestration, data flow routing, and model coordination across multiple stages of the simulation process. It should be noted that the present prototype assumes that basic data acquisition and control interfaces are already available, and the focus of this work is on validating the structural feasibility of the PEI simulation workflow. The development of a fully autonomous Industry 4.0/5.0-style system with self-correction and feedback control is left for future research.

The remainder of this paper is organized as follows. Section 2 introduces the architecture of the proposed PEI simulation method, including the definitions and boundaries of the P, E, and I modules as well as the execution workflow. Section 3 presents a case implementation and discusses the simulation results. Section 4 concludes the paper with a summary of contributions, limitations, and directions for future work.

2. Architecture of the PEI Simulation Method

Figure 1 illustrates the production workflow of a typical process manufacturing system. In such systems, the complete production process is usually composed of multiple processing steps connected in a predefined sequence. Each step is typically equipped with one or more devices responsible for executing specific operations. After the execution of each step, the in-process material is inspected to collect key process state data and evaluate the effectiveness of the operation. This type of production process exhibits distinct structural continuity and stage-wise evolution, forming a complex system characterized by multi-stage interactions and multi-parameter coupling. To ensure traceability and consistency, each process step is coupled with a specific equipment unit and followed by inspection. The final inspection leads to the release of the finished product, completing the process flow. This schematic structure serves as the abstract foundation for the PEI simulation organization.

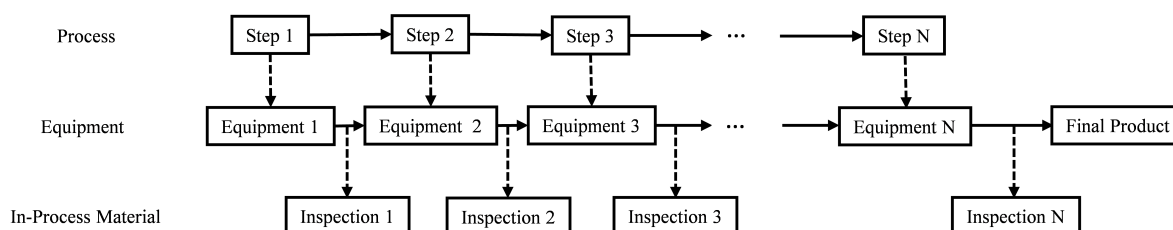


Figure 1. The production flow of a typical process manufacturing system.

The “step–equipment–inspection” configuration is commonly observed in industrial production lines, reflecting the continuity, structured execution, and stage-wise transformation inherent to process manufacturing. However, conventional simulation approaches tend to treat each step as an isolated modeling object. As a result, the operational logic of

equipment, material state transitions, and process orchestration are often modeled independently, lacking a unified framework for describing their interdependencies. This separation limits the ability to analyze multi-step interactions or predict outcomes under complex operating conditions.

To overcome these limitations, this paper proposes an architecture illustrated in Figure 2, referred to as the PEI simulation method. It introduces a structured organization comprising three cooperating modules: a Process Structure module (P), which defines the sequence and dependencies of the process steps; an Equipment Behavior module (E), responsible for translating control inputs into physical operating conditions; and an In-Process State module (I), which simulates the transformation of material properties under the given process environment.

Each simulation node, corresponding to a process step, is managed by the P module, which activates the relevant E and I modules assigned to that step. Within a node, the E module generates the local process conditions, which are then used by the I module to compute material state transitions. This hierarchical and modular execution mechanism, as visualized in Figure 2, enables PEI to support multi-stage simulations with explicit structure, logical task flow, and traceable state evolution across the process chain. In this architecture, Figure 2 visualizes a representative simulation node, highlighting the typical internal coupling of control and monitoring parameters via the E and I modules.

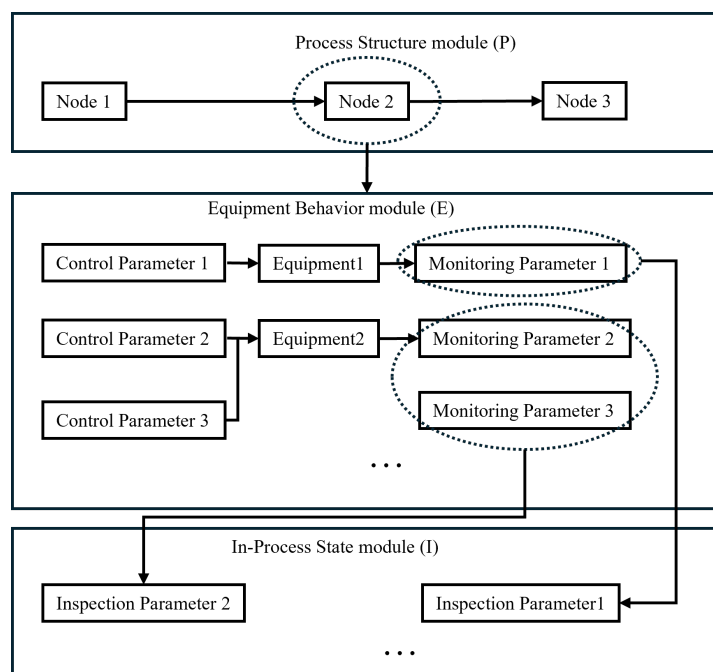


Figure 2. The architecture of the PEI simulation method. This diagram illustrates the internal logic of a typical simulation node, combining equipment control and in-process state evolution.

2.1. Definitions and Boundaries of the PEI Modules

The Process Structure module (P) defines the connectivity of steps, execution sequence, and data dependencies within the overall production workflow. Conceptually, it functions as a structured process graph or a directed scheduling network that determines how simulation tasks are organized and in what order nodes are activated. The P module itself does not perform any physical modeling or numerical computation; rather, it serves as the scheduling center of the system, managing the flow of simulation tasks and activation of simulation nodes. Its boundary lies at the organizational level of process topology, without involving equipment behavior or material state modeling.

The Equipment Behavior module (E) characterizes the relationship between control operations and the physical response of equipment. It establishes a mapping from control inputs to equipment responses, capturing how the system reacts under given operational conditions. The E module serves as the bridge between control settings and the physical process environment, transforming setpoints into physical parameters such as temperature, pressure, or flow rate. The modeling strategies for E modules may include various structures such as: single-input single-output (SISO), multi-input single-output (MISO), control loop modeling, data-driven models (e.g., regression, random forest, neural networks), or physics-based formulations. Inputs to the E module consist of control or setpoint variables, while outputs are the corresponding physical responses. Its scope is limited to equipment-level behavior and does not cover material transformation or the structural logic of the process. In practical industrial systems such as those governed by Distributed Control Systems (DCSs), the E module interfaces with two categories of parameters: control parameters and monitoring parameters. Control parameters refer to adjustable input signals, such as valve openings, power settings, or motor speeds, that define how equipment is operated. Monitoring parameters, by contrast, are sensor-acquired feedback values, including flow rate, temperature, pressure, or equipment current, which reflect the real-time operating state of the device. This distinction aligns with real-world engineering practices, where control signals drive the system's behavior, and monitoring signals are used for validation, supervision, or as inputs to state estimation models. Within the PEI framework, both parameter types are encapsulated by the E module, enabling accurate reflection of equipment behavior and providing critical support for downstream in-process state modeling.

The In-Process State module (I) simulates the evolution of material properties under given process conditions. It is the core modeling component responsible for reflecting process effects and evaluating product-related indicators. The goal of the I module is to derive the target state of in-process materials at each simulation step, based on initial material conditions and the external process environment. Inputs to the I module include the following: (1) the output state from the upstream I module (i.e., the result of the previous process step), and (2) the response output from the current step's E module (i.e., the process condition). These two factors jointly determine the transformation path of materials within the current equipment, forming a "raw state–process condition–final state" modeling chain. Common I module types include single-variable prediction models, multi-variable coupling models, dynamic evolution models based on control inputs, data-driven soft sensors, and simplified mechanistic models. The output of the I module is the updated in-process material state at the end of the current step, serving as a critical indicator for assessing the effectiveness of control strategies, process configurations, and structural flow logic.

2.2. Module Collaboration Mechanism and Simulation Execution Workflow

The PEI simulation method organizes the execution of simulation tasks through the Process Structure module (P), which determines the step-by-step scheduling of the entire simulation process. Within each step, the system sequentially invokes the Equipment Behavior module (E) and the In-Process State module (I). These modules collaborate through data transfer: the E module generates the local process conditions, while the I module simulates material state transitions under those conditions.

In each process step, the E module is first called to compute the equipment response based on the specified control parameters. The resulting process conditions then serve as part of the input to the I module. Upon receiving both the equipment response and the upstream material state, the I module performs the prediction and update of the current material properties. The simulation proceeds step by step, with the E and I modules

operating collaboratively within each node. The output of the I module is passed to the next step, enabling a progressive evolution of process states throughout the simulation. The overall execution sequence is controlled by the P module, continuing until the final step is completed. At each step, the E and I modules are executed in a synchronized manner, ensuring consistent data dependencies and preventing race conditions.

To realize this structured execution, the PEI simulation method follows the simulation workflow illustrated in Figure 3. The system first uses the P module to identify the current simulation node and retrieve its assigned parameters. Then, the E module computes the equipment response from the input settings, and this response is passed to the I module along with the upstream material state. The I module calculates the updated material state, which is then written into the simulation state cache. The P module checks whether the current node is the final step. If not, it proceeds to the next node and repeats the process. If it is the final step, the simulation terminates and outputs the complete set of simulation results.

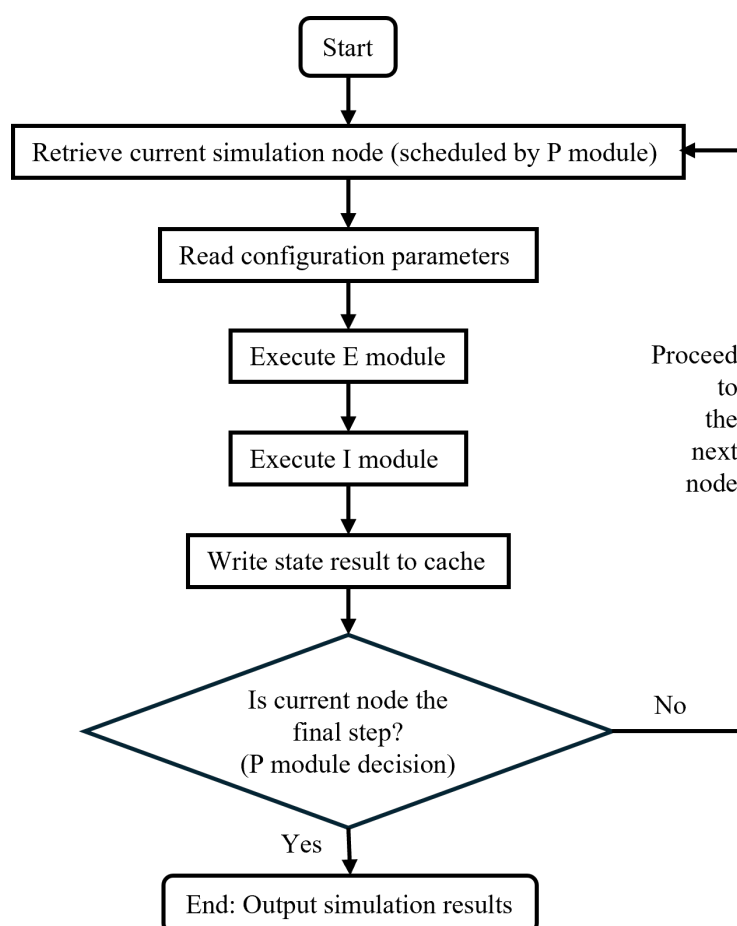


Figure 3. Execution workflow of the PEI simulation method.

3. Results and Discussion

This section presents a validation scenario and discusses the execution feasibility and structural properties of the PEI simulation method. Both simulation results and methodological considerations are analyzed to demonstrate applicability and future extensibility.

To verify the structural organization and execution feasibility of the PEI simulation method, a simplified simulation scenario is constructed involving the following process chain: cold/hot water control → thermal mixing → slurry state evolution. This process exhibits clear stepwise segmentation and state transfer characteristics, making it suitable

for demonstrating the collaboration among the Process Structure, Equipment Behavior, and In-Process State modules.

The complete simulation consists of three sequential steps:

- **Step 1: Cold and hot water regulation**
The control parameters are the valve openings for cold and hot water. The objective is to adjust the mixed temperature of the thermal water. This step is associated with the E module, which computes the resulting water temperature based on the valve settings (equipment response variable). No in-process material state is updated in this step.
- **Step 2: Heat exchange between thermal water and equipment**
The input control parameter is the thermal water temperature, and the equipment response is the temperature (or heat flux) transferred within the heat exchanger. The E module simulates the heat conduction behavior of the exchanger, while the I module predicts the temperature rise of the slurry material under the given thermal conditions.
- **Step 3: Slurry state evolution**
No new control parameters are introduced. This step inherits the equipment response and the slurry state from the previous step. The I module is used to predict final product quality indicators such as viscosity, particle size distribution, or homogeneity, based on the received thermal conditions and exposure duration.

Throughout the simulation, the P module governs the execution flow, linking the three steps and ensuring that simulation tasks proceed in the correct order. Before each step is executed, the P module schedules the corresponding simulation node, invokes the associated E and I modules, and updates the system state before proceeding to the next node. The current simulation prototype is implemented in Python, using modular functions to represent equipment and state models. This allows flexible model binding and supports cross-platform deployment.

3.1. Model Library Construction and Binding Mechanism

To support the modeling needs of equipment behavior and in-process material states across multiple process steps, the PEI simulation method adopts a modular library-based approach to organize both E modules (equipment models) and I modules (in-process models). Each model is implemented as an independent function or submodule and is bound to simulation nodes in the process structure, yielding a clear mapping between simulation nodes and model instances.

In the current example, the model library includes the following categories:

- **E Module Library:** This includes the cold/hot water mixing temperature model and the heat exchanger thermal conduction model.
 - The mixing model accepts two control inputs (valve openings) and outputs the temperature of the mixed thermal water stream.
 - The heat exchanger model takes the setpoint thermal water temperature as input and computes the temperature transmitted to the slurry within the device.
- **I Module Library:** This includes the slurry heating model and the slurry state evolution model.
 - The heating model predicts the temperature increase of the slurry under given thermal conditions.
 - The evolution model estimates quality-related properties such as viscosity and particle size distribution based on the exposure time and process conditions.

These models are constructed based on empirical data or fitted rules, and they also support replacement with physics-based or data-driven alternatives.

- **Model Binding Mechanism:**
Before simulation begins, the system reads the process structure definition (e.g., configuration files or mapping tables) to bind each simulation step (e.g., Step 1, Step 2) to the designated E and I model names. During execution, the simulation engine dynamically invokes the correct model instances based on the current step ID, enabling task-level model scheduling and module reuse.

This mechanism not only supports the flexible composition of multiple steps and multiple models but also facilitates later-stage model extension, replacement, and version control, making it adaptable to various simulation requirements in complex process manufacturing scenarios.

3.2. Simulation Execution Example

To validate the PEI method's ability to support collaborative simulation under structured process control, a three-step case study was conducted using a predefined set of input parameters. The objective is to demonstrate the end-to-end execution, from control input to equipment response and finally to in-process state evolution. Note that the parameters and results presented here are illustrative and not derived from real industrial data. They are intended to showcase the organization and execution mechanism of the proposed simulation method.

- **Step 1: Cold/hot water regulation**
The control parameters are set to 40% opening for the cold water valve and 70% for the hot water valve. The system invokes the bound cold/hot water mixing model, which calculates a mixed thermal water temperature of 53.38 °C. This temperature is treated as the equipment response output, stored in the state cache, and passed to the next step.
- **Step 2: Heat exchange between thermal water and equipment**
The computed thermal water temperature is passed into the heat exchanger conduction model, which outputs a material-side heating temperature of 50.71 °C. The system then invokes the slurry-heating model, which predicts a new slurry temperature of 45.50 °C based on the initial slurry temperature of 27.00 °C and the heating condition. This value is recorded as the output in-process state for the current step.
- **Step 3: Slurry state evolution**
Finally, the slurry state evolution model is executed using the previously calculated slurry outlet temperature (45.50 °C) and an initial viscosity of 2.30 Pa·s. Under the given thermal condition and processing time, the model predicts a final viscosity of 1.67 Pa·s.

After the simulation is completed, all outputs are written to a structured result record. The actual console output (Figure 4) and its structured summary (Table 1) provide a stepwise illustration of the run, from the control inputs, through the equipment responses, to the in-process state updates at each step. Together, they demonstrate the input–output coupling across modules under process-structured control and confirm the feasibility of the PEI method for module decoupling, path control, and traceable state evolution.

The overall simulation accuracy at this stage depends on the predictive accuracy of each trained E/I model. A systematic assessment of error propagation across the PEI chain is left for future work.

```

=====
Simulation Start
=====

Cold valve opening (%): 40
Hot valve opening (%): 70
Slurry initial temp (°C): 27
Viscosity initial (Pa·s): 2.3
[Step 1] Mixed thermal water temp           = 53.38 °C
[Step 2] Material-side heating temp          = 50.71 °C
[Step 3] Slurry heating → slurry temp (out) = 45.50 °C
[Step 4] Slurry viscosity (out)              = 1.67 Pa·s

=====
Simulation Finished
=====

Inputs:
cold_open=40.00 %, hot_open=70.00 %
slurry_in=27.00 °C, viscosity_in=2.30 Pa·s

Outputs:
mixed_temp           = 53.38 °C
hex_material_temp    = 50.71 °C
slurry_temp_out      = 45.50 °C
viscosity_out        = 1.67 Pa·s

```

Figure 4. Console run of the PEI prototype, demonstrating stepwise simulation execution and state evolution.

Table 1. Structured result record of the three-step case under the PEI execution.

Step	Control Input (to E)	Equipment Response	Material State (Before → After)
1. Cold/Hot Mixing	Valve = 40%/70%	Mixed water = 53.38 °C	– → –
2. Heat Exchange	Water = 53.38 °C	Heating temp = 50.71 °C	Slurry temp = 27.00 °C → 45.50 °C
3. Slurry Evolution	–	–	Viscosity = 2.30 Pa·s → 1.67 Pa·s

3.3. Model Realization Strategy

The core innovation of the PEI simulation method is its structured organization and modular coordination, which distinguish it from conventional modeling approaches. Each Equipment Behavior (E) or In-Process State (I) module is only required to complete a defined input-to-output mapping, and its internal implementation can be flexibly selected according to the application scenario.

Depending on data availability and engineering requirements, users may adopt one of the following modeling strategies:

- Physics-based modeling, such as heat balance equations, thermal transfer formulas, or empirically derived correlations.
- Data-driven modeling, including regression models, random forests, or neural networks.
- Hybrid modeling approaches, where basic physical structures are retained and enhanced using data-fitting or optimization techniques.

This strategy decouples model construction from the simulation framework, enabling a simulation mechanism that is interchangeable, extensible, and schedulable. For example, in the slurry heating module, one can either apply a first-order dynamic model to approximate the thermal process or use a neural network trained on pilot data. Both models can be seamlessly integrated into the PEI framework without affecting the simulation logic.

Thus, the PEI method focuses on standardizing simulation task scheduling rather than constraining the modeling approach. This flexibility ensures broad engineering adaptability and provides space for the integration of advanced algorithms.

Implementation in a simplified process scenario demonstrates that the PEI method offers a clear execution logic and strong engineering applicability, providing a robust foundation for tasks such as process validation, parameter prediction, and state analysis in process manufacturing systems. Future work will focus on enhancing model accuracy, enabling adaptive configuration of process flows, and integrating with real-world industrial data to further strengthen the method's industrial adaptability and intelligent potential.

Despite its structural clarity and scheduling flexibility, the current implementation of the PEI simulation method also faces several limitations. First, the method assumes the availability of well-defined equipment and in-process models, which may not be readily available in early-stage applications. Second, the dependency on manually defined task sequences could limit scalability for highly dynamic systems. Third, the absence of feedback loops in the current prototype limits its ability to support real-time process correction or adaptation. Future work will aim to address these limitations by integrating automated model generation, real-time data adaptation, and process feedback mechanisms. Additionally, efforts will be made to enhance scalability, enabling the PEI framework to support more complex industrial systems involving multiple units, parallel task flows, or distributed simulations.

4. Conclusions

This paper presents the PEI simulation method for process manufacturing. It abstracts the production workflow into three core modeling entities: Process Structure, Equipment Behavior, and In-Process State. By employing a modular architecture and structure-driven execution mechanism, the method establishes a full process simulation framework that is composable, scalable, and schedulable. It supports node-level model binding, structure-level execution control, and state-level traceability, addressing the limitations of traditional approaches where process flows are fragmented, model coupling is unclear, and organizational mechanisms are lacking.

Despite these advantages, the current implementation also presents several limitations. The method assumes the availability of well-defined equipment and in-process models, which may not be readily available during early-stage system development. Additionally, task sequencing is manually configured, limiting scalability in highly dynamic environments. The current prototype lacks real-time feedback integration, which restricts its use in adaptive simulation or closed-loop control. Future efforts will focus on incorporating automated model generation, feedback-driven execution logic, and broader compatibility with industrial data sources to improve adaptability and deployment readiness.

Author Contributions: Conceptualization, X.T. and M.Y.; methodology, X.T.; software, W.X.; validation, W.X. and M.Y.; formal analysis, X.T.; investigation, X.T. and W.X.; resources, G.X. and Y.S.; data curation, X.T.; writing—original draft preparation, X.T.; writing—review and editing, X.T., W.X., and G.X.; visualization, X.T.; supervision, Y.S.; project administration, G.X.; funding acquisition, G.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science and Technology Department of Yibin grant No. JB2024002.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding authors.

Acknowledgments: The authors would like to express their sincere gratitude to Yibin Grace Group Co., Ltd., for providing access to production facilities and valuable technical support, which greatly contributed to the development of this research.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Liu, Y.; Jia, M.; Xu, D.; Yang, T.; Yao, Y. Physics-guided graph learning soft sensor for chemical processes. *Chemom. Intell. Lab. Syst.* **2024**, *249*, 105131. [\[CrossRef\]](#)
2. Shin, Y.; Smith, R.; Hwang, S. Development of model predictive control system using an artificial neural network: A case study with a distillation column. *J. Clean. Prod.* **2022**, *277*, 124124. [\[CrossRef\]](#)
3. Chinpravit, J.; Panjapornpon, C. Model predictive control of vinyl chloride monomer process by Aspen Plus Dynamics and MATLAB/Simulink co-simulation approach. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Phuket, Thailand, 22–24 April 2020.
4. Hernández, E.; Belinchón, A.; Pachón, E.R.; Navarro, P.; Palomar, J. Toward Sustainable and Cost-Effective CO₂ Conversion Processes to Propylene Carbonate Based on Ionic Liquids. *Adv. Sustain. Syst.* **2022**, *6*, 2200384. [\[CrossRef\]](#)
5. Uhlemann, T.H.-J.; Lehmann, C.; Steinhilper, R. The digital twin: Realizing the cyber-physical production system for industry 4.0. *Procedia CIRP* **2017**, *61*, 335–340. [\[CrossRef\]](#)
6. Ma, L.; Zhao, Y.; Wang, B.; Shen, F. A multistep sequence-to-sequence model with attention LSTM neural networks for industrial soft sensor application. *IEEE Sens. J.* **2023**, *23*, 10801–10813. [\[CrossRef\]](#)
7. Boskabadi, M.R.; Murugaiah, M.; Nielsen, T.R.; Sivaram, A.; Sin, G.; Mansouri, S.S. Virtual Sensor for Sustainable Large-Scale Process Monitoring. *Ind. Eng. Chem. Res.* **2025**, *64*, 3902–3917. [\[CrossRef\]](#)
8. González-Herbón, R.; González-Mateos, G.; Rodríguez-Ossorio, J.R.; Prada, M.A.; Morán, A.; Alonso, S.; Fuertes, J.J.; Domínguez, M. Assessment and deployment of a LSTM-based virtual sensor in an industrial process control loop. *Neural Comput. Appl.* **2025**, *37*, 10507–10519. [\[CrossRef\]](#)
9. Wang, J.; Qiu, K.; Wang, R.; Zhou, X.; Guo, Y. Development of soft sensor based on sequential kernel fuzzy partitioning and just-in-time relevance vector machine for multiphase batch processes. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–10. [\[CrossRef\]](#)
10. Li, J.; Wang, J.; Sui, E.; Wang, W.; He, R. Soft sensor development based on hybrid modeling with ensemble learning for multimode batch processes. *IEEE Sens. J.* **2025**, *25*, 15588–15597. [\[CrossRef\]](#)
11. Hussain, M.; Zabiri, H.; Uddin, F.; Yusup, S.; Tufa, L.D. Pilot-scale biomass gasification system for hydrogen production from palm kernel shell (part B): Dynamic and control studies. *Biomass Convers. Biorefin.* **2023**, *13*, 8173–8195. [\[CrossRef\]](#)
12. Haque, M.E.; Palanki, S. Advanced Process Control Strategies for Efficient Methanol Production from Natural Gas. *Processes* **2025**, *13*, 424. [\[CrossRef\]](#)
13. Ahmed, O.H.K. Enhancement of Al Dar Crude Oil Pre-Processing Using Model Predictive Control. Master's Thesis, Al-Neelain University, Khartoum, Sudan, 2022.
14. Asensi Martínez, D. Study of Multivariable Control Techniques Using Process Simulators and MATLAB/Simulink. Master's Thesis, Universidad de Alicante, Raspeig, Spain, 2024.
15. Reguig, A.A.; Sari, B.; Berber, S. Modelling and control of a gas blending station: A real industrial case study. In Proceedings of the 19th International Multi-Conference on Systems, Sétif, Algeria, 6–10 May 2022.
16. Tian, Y.; Pappas, I.; Burnak, B.; Katz, J.; Pistikopoulos, E.N. Simultaneous design & control of a reactive distillation system—a parametric optimization & control approach. *Chem. Eng. Sci.* **2021**, *230*, 116232.
17. Farias, A.O.; Queiroz, G.A.C.; Bessa, I.V.; Medeiros, R.L.P.; Cordeiro, L.C.; Palhares, R.M. Sim3tanks: A benchmark model simulator for process control and monitoring. *IEEE Access* **2018**, *6*, 62234–62254. [\[CrossRef\]](#)
18. Efheij, H.; Albagul, A.; Albraiki, N.A. Comparison of model predictive control and PID controller in real time process control system. In Proceedings of the 2019 19th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), Sousse, Tunisia, 24–26 March 2019.
19. Pistikopoulos, E.N.; Diangelakis, N.A. Towards the integration of process design, control and scheduling: Are we getting closer? *Comput. Chem. Eng.* **2016**, *91*, 85–92. [\[CrossRef\]](#)
20. Rodriguez, C.; Mhaskar, P.; Mahalec, V. Linear hybrid models of distillation towers. *Comput. Chem. Eng.* **2023**, *171*, 108160. [\[CrossRef\]](#)
21. Wei, B.; Tan, S.; Zhang, Q.; Zhou, H. A hybrid soft sensor for key product yield of FCC unit based on deep learning framework driven by data and process mechanism. *Chem. Eng. Res. Des.* **2024**, *202*, 429–443. [\[CrossRef\]](#)
22. Zhu, X.; Rehman, K.U.; Bo, W.; Shahzad, M.; Hassan, A. Data-driven soft sensor model based on deep learning for quality prediction of industrial processes. *SN Comput. Sci.* **2021**, *2*, 40. [\[CrossRef\]](#)
23. Kim, C.; Shah, M.; Sahlodin, A.M. Design of multi-loop control systems for distillation columns: Review of past and recent mathematical tools. *Chem. Prod. Process Model.* **2022**, *17*, 171–197. [\[CrossRef\]](#)
24. Mozharovskii, I.; Shevlyagina, S. A hybrid approach to soft sensor development for distillation-in-series plant under input data low variability. *Meas. Sci. Technol.* **2024**, *35*, 076211. [\[CrossRef\]](#)
25. Curreri, F.; Patanè, L.; Xibilia, M.G. RNN-and LSTM-based soft sensors transferability for an industrial process. *Sensors* **2021**, *21*, 823. [\[CrossRef\]](#)

26. Sharma, M.; Verma, P.; Mathew, L. Design of an intelligent controller for a process control system. In Proceedings of the 2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH), Greater Noida, India, 3–5 February 2016.
27. Alsanousie, A.A.; Elsamni, O.A.; Attia, A.E.; Elhelw, M. Transient and troubleshoots management of aged small-scale steam power plants using Aspen Plus Dynamics. *Energy* **2021**, *223*, 120079. [[CrossRef](#)]
28. Ferro, V.R.; Ruiz, E.; De Riva, J.; Palomar, J. Introducing process simulation in ionic liquids design/selection for separation processes based on operational and economic criteria through the example of their regeneration. *Sep. Purif. Technol.* **2012**, *97*, 195–204. [[CrossRef](#)]
29. Dong, G.; Zhang, D.; Yue, J.; Wu, L.; Han, L.; Zhang, Z.; Qian, X. Progress of air flow field and fiber motion analysis in textile manufacturing process: A review. *Text. Res. J.* **2024**, *94*, 1581–1612. [[CrossRef](#)]
30. Zhang, J.; Yu, G.; Bo, C.; Li, J. Data-driven dynamic modeling and fault diagnosis of dimethyl oxalate industrial production process. In Proceedings of the 2024 6th International Conference on Data-driven Optimization of Complex Systems (DOCS), Hangzhou, China, 16–18 August 2024.
31. Mohindru, P. Review on PID, fuzzy and hybrid fuzzy PID controllers for controlling non-linear dynamic behaviour of chemical plants. *Artif. Intell. Rev.* **2024**, *57*, 97. [[CrossRef](#)]
32. Yandrapu, V.P.; Kanidarapu, N.R. Process design for energy efficient, economically feasible, environmentally safe methyl chloride production process plant: Chlorination of methane route. *Process Saf. Environ. Prot.* **2021**, *154*, 360–371. [[CrossRef](#)]
33. Bedia, J.; Ruiz, E.; de Riva, J.; Ferro, V.R.; Palomar, J.; Rodriguez, J.J. Optimized ionic liquids for toluene absorption. *AIChE J.* **2013**, *59*, 1648–1656. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.