machines

Article

Digital-Twins-Driven Semi-Physical Simulation for Testing and Evaluation of Industrial Software in a Smart Manufacturing System

Keqiang Cheng [1,2,*], Qiang Wang [2], Dongyu Yang [2], Qingyun Dai [3] and Meilin Wang [1]

[1] School of Information Engineering, Guangdong University of Technology, Guangzhou 510006, China; wml@gdut.edu.cn
[2] China Electronic Product Reliability and Environmental Testing Research Institute, Guangzhou 510610, China; wangqiang@ceprei.com (Q.W.); yangdy@ceprei.com (D.Y.)
[3] School of Electronics and Information, Guangdong Polytechnic Normal University, Guangzhou 510665, China; dqy@gpnu.edu.cn
* Correspondence: chengkeqiang@ceprei.com

Abstract: To satisfy the needs of the individualized manufacturing of products, the smart manufacturing system (SMS) is frequently reconfigured. To quickly verify the reliability and adaptability of industrial software in reconfiguring the SMS for new or upgraded product orders, a semi-physical simulation method for testing and evaluation of industrial software is proposed based on digital-twins-driven technology. By establishing a semi-physical simulation model of SMS, the reliability and robustness of the software system are quickly verified by running industrial software in various manufacturing scenarios. In this paper, the key technologies to carry out semi-physical simulation testing and evaluation of industrial software for SMSs are expounded in detail, including how to synchronize cyber and physical systems, how to conduct semi-physical accelerated simulation testing, and how to identify defects quickly in industrial software used in actual production environments. By establishing a semi-physical simulation production line model for stepper motors, the effectiveness and practicality of the proposed approach are verified, and the testing verification time of industrial software is significantly reduced. Finally, the robustness of the industrial software for SMS is further verified by conducting fault injection testing, so as to provide implications for fault prognostics or fault-prevention research.

Keywords: digital twin; industrial software; semi-physical simulation; accelerated testing; fault injection

check for updates

Citation: Cheng, K.; Wang, Q.; Yang, D.; Dai, Q.; Wang, M. Digital-Twins-Driven Semi-Physical Simulation for Testing and Evaluation of Industrial Software in a Smart Manufacturing System. *Machines* **2022**, *10*, 388. https://doi.org/10.3390/machines10050388

Academic Editor: Zhuming Bi

Received: 7 April 2022
Accepted: 16 May 2022
Published: 18 May 2022

1. Introduction

Industrial software is the brain and nerves of smart manufacturing. The industrial software of a smart manufacturing system (SMS) is generally used for industrial control, production, and storing and processing production-related data. Its reliability and stability are extremely crucial and directly related to the economy, efficiency, and production safety. To adapt to the urgent needs for digital transformation and upgrading of manufacturing enterprises, the production system urgently needs to be transformed. Meanwhile, the traditional workshop production process is not transparent, resulting in production process control being seriously restricted by empirical knowledge. When upgrading the traditional workshop or deeply optimizing the workshop production process, how to evaluate whether the transformation solutions provided by service providers are effective, as well as how to choose the optimal solution from multiple solutions, are the more concerning issues of manufacturing enterprises. The industrial software of a SMS plays a pivotal role; its software quality directly affects the extent to which the above problems are solved, so we should pay attention to the testing and evaluation of industrial software.

The industrial software of a SMS has distinctive industry characteristics. It is closely related to the operating environment, where different production processes require different industrial software to support it. When using the traditional method to test, it is always difficult to achieve the desired results, as there are generally certain differences between the testing environment of nonuser sites and the production sites of real users [1]. If the testing is conducted in the actual operating scenarios, it will take a large test workload, a long time, a high test-cost, and the risk coefficient is also high. Along with the development of software testing and simulation technology, some scholars introduce some physical models, and the rest are described by mathematical models [2,3]. Yang et al. [4] carried out multi-fixed wing UAV guidance and formation testing using a semi-physical simulation method. Krzysztof et al. [5] used hardware-in-the-loop simulation technology to predict vibration monitoring during the high-speed milling process. Quérel et al. [6] developed a semi-physical model to predict the NOx emissions generated by diesel engines. Xiong et al. [7] proposed a kind of semi-physical simulation optimization method for the sintering flow in the steel production process, debugging relevant parameters on the simulation platform to reduce debugging risks. To promote the development of heavy haul trains, the running state of the long marshaling train is simulated on the semi-physical simulation platform [8]. Ying et al. [9] established a semi-physical braking intelligent test system to study the factors affecting the braking technology of freight trains. Chien et al. [10] developed a scheme of a semi-physical verification platform to meet the requirement of performance evaluation on the light-screen array measurement system. It can be seen from the above references that carrying out semi-physical simulation tests can effectively build simulation scenarios and shorten the development cycle. However, there are still some problems with the current semi-physical simulation technology. For example, it is difficult to build high-precision simulation scenarios; the data used for testing cannot run synchronously with the physical production line.

There are many other factors that affect the reliability of industrial software besides the operating environment, such as the quality problems of the software system itself and the influence brought by external network attacks. To improve the quality problems of software systems, it is necessary to focus on the whole life-cycle management of software development and improve the effectiveness of software testing. In addition, some innovative initiatives are used to enhance industrial software for cyber-attacks. Elsisi, M. et al. [11,12] proposed an integrated IoT-architecture-based deep neural network in the online detection system of automated guided vehicles and the power transformer to defend against cyber-attacks. This paper mainly focuses on the establishment of high precision simulation environment, testing, and evaluation of the industrial software.

In recent years, with the rapid development of multidisciplinary modeling and simulation technology, the concept and application of digital twin has become a research hotspot. Leng et al. [13,14] proposed a digital twins-driven manufacturing cyber-physical system for parallel control of a smart workshop. When carrying out industrial software simulation testing, if the digital twin model can be constructed to interact with the system software in real time and realize high-fidelity testing, the testing time and cost can be effectively shortened. This paper uses digital twin technology to conduct testing and evaluation industrial software based on the semi-physical simulation system. The rest of the paper is organized as follows. Section 2 reviews the related work of the development of industrial software testing technology and applications of digital twin in the manufacturing domain to identify the research gaps. Section 3 presents the framework of the digital twins-driven cyber-physical system model for industrial software testing and evaluation. Section 4 details three enabling technologies that are key for engineering the proposed model. Section 5 conducts experiments on the stepping-motor manufacturing system and discusses related issues. Finally, Section 6 draws a conclusion, followed by discussions on a future research direction.

2. Related Work

Industrial software reliability testing of SMS is used to verify whether the system meets the specified requirements under the setting conditions, as well as to assess the deviation of expected results from the actual results. Due to the uncertainty of industrial software requirements, the subjective limitations of developers, and the complexity of manufacturing systems, defects will inevitably occur in the development process of industrial software [15]. In addition, with long-term operation, the system performance deteriorates, and the occupied resources increase; aging software can cause failure or downtime [16]. To avoid the phenomenon of software-system aging, Huang et al. [17] proposed a software regeneration technique which used periodic clearing of an internal state or operating environment to avoid sudden failure caused by aging, but the regeneration frequency is not easy to grasp. Some scholars proposed prediction and analysis of system resource consumption based on the information of system operation parameters, but it was difficult to capture the hidden laws of aging performance parameters [18,19]. To enhance the reliability of industrial software, slow down the speed of software aging, and better service manufacturing, Jeff et al. [20] proposed strengthening the system software simulation testing by establishing a high-fidelity-test running environment. In the testing process, to save personnel, time, and hardware resources, Just et al. [21] proposed the use of automated testing methods, but this is a complex task involving not only the execution of test cases, but also the generation of appropriate input values and the evaluation of the corresponding outputs. As industrial software tends to become larger and more complex, Stahl et al. [22] proposed continuous integration testing for large and complex software with multiple dependencies. Although frequent software testing at various stages of the development cycle can improve testing efficiency and predictability, the overall testing time is too long and costly. Currently, with the rapid development of individualized production, the existing reliability testing and evaluation techniques for manufacturing systems software can no longer meet the requirements of development.

Digital twin is a simulation technology that integrates multi-physical quantities, multi-scales, multi-probabilities, and realizes the process of simultaneous evolution of digital twin and the whole life-cycle of the physical device through high-fidelity virtual mapping [23,24]. The concept of digital twin was initially applied in the aerospace field [25]. Now, the application of digital twin has been extended to smart manufacturing, vehicle testing, equipment preventive maintenance, throughout all stages of the product's lifecycle [26]. Liu and Leng et al. [27,28] proposed the configuration motion control optimization design model based on digital twin for early detection of design defects in SMS. Ge et al. [29] proposed the use of digital twin technology for testing and verification of self-driving vehicles in a limited environment, which can realistically simulate complex road scenarios. Angjeliu [30] used digital twin technology for the maintenance and conservation of Milan Cathedral to predict and assess structural development trends by building a high-precision structural twin model of the building. Leng et al. [31] proposed establishing the digital twin model of a manufacturing workshop on the upper layer of the server and put forward the ManuChain framework to innovate the large-scale personalized production mode. Digital twin technology was applied to nuclear power equipment operation services to predict the remaining life of equipment by diagnosing and evaluating system performance [32]. Digital twin has been described as "the re-engineering of structural life prediction and management" by building ultra-high-fidelity models to predict the life of partial aircraft components and the entire aircraft [33]. Tao et al. [34] proposed the use of wind engine digital twin models to drive failure prediction and health management, which can effectively achieve the interaction and integration of twin models and entities. Since 2020, the rampant spread of COVID-19 has seriously affected human life and economic activities. Leng et al. [35] proposed an open process SMS remote semi-physical commissioning method based on digital twin, solving the obstacles of remote commissioning.

With the integration and development of simulation technology of industrial Internet, big data, and artificial intelligence, digital twin technology will significantly promote the development of changes in the whole life-cycle of smart manufacturing. The smart workshop is the main carrier of smart manufacturing. Tao et al. [36] proposed a method to construct a digital twin model of the smart workshop and studied the operation mechanism and implementation method. Under the individualized design requirements, Liu et al. [37,38] adopted a rapid individualized design method based on digital twin to achieve dynamic multi-objective optimization of the manufacturing system. Guo et al. [39] proposed a digital twin model of an assembly graduate manufacturing system to organize production activities through design, installation and logistics tickets for the characteristics and workflow of assembly islands, effectively enhancing the visibility of shop-floor management. Industrial software is the core of smart manufacturing, manipulating the data of its whole life-cycle of products, including its reliability and stability, which directly affect the quality and efficiency of the products. Using digital twin technology, vulnerabilities and deficiencies can be discovered during the development and testing phase of industrial software without having to wait for the actual production operation phase in order to run tests.

3. Framework of Digital Twins Smart Production Line for Running Industrial Software

This paper attempts to establish the digital twin model of the smart production line to study the reliability testing and evaluation of industrial software. The digital twin system replaces the traditional physical production line to provide a high simulation and safe operating environment for the industrial software, avoiding possible damage to the real production equipment during performance testing. By using virtual clock simulation acceleration, the digital twin system can realize high-speed message queue feedback and reduce useless waiting time. In this way, it can help enterprises effectively find and improve performance bottlenecks in industrial software and improve production efficiency.

The digital twin model of a smart production line is based on the attributes of the physical entity of the manufacturing system in virtual space, to realize the mapping of the whole life-cycle of the entity. To conduct industrial software testing, the digital twin model should not only meet the functions of normal interaction production with industrial software, but also cover other functions, such as equipment state monitoring and faults alarming. Therefore, it is necessary to supplement the abnormal state of equipment in a digital twin model to trigger the occurrence of abnormal events. The digital twin model structure of the smart production line for running industrial software is shown in Figure 1. In this framework, the discrete event simulation of the smart production line is accelerated and tested by constructing the 3D structure, communication data, and communication interfaces of the actual physical model. The smart production line-management system interacts with the digital twin model through supervisory control and data acquisition (SCADA), including issuing business processes and feeding back model status data.

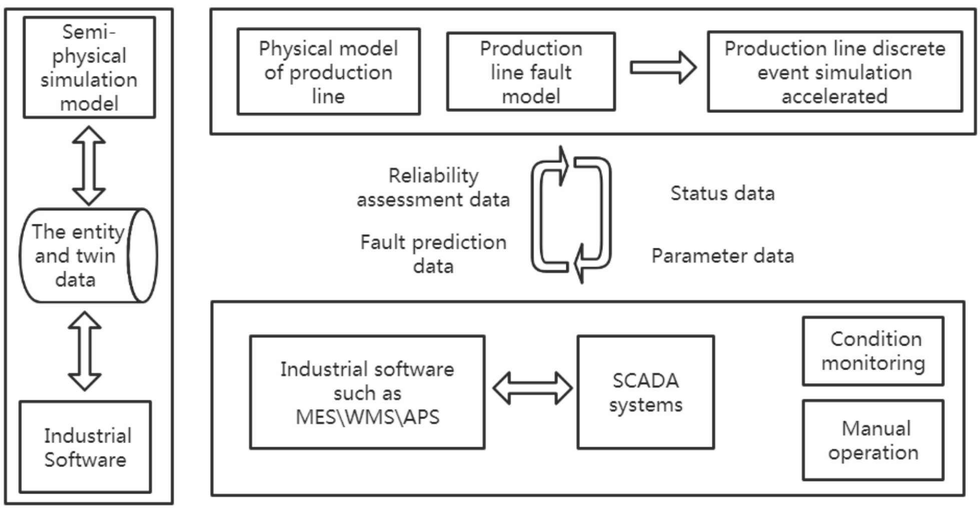

Figure 1. Digital twin smart production line model framework for running industrial software.

4. Key Enabling Technologies

4.1. Accelerated Semi-Physical Simulation Testing Method

The smart production line is a typical discrete event system. After constructing the digital twin model of the smart production line, the discrete event simulation acceleration method is applied to realize the acceleration effects of the digital twin model to execute the production process and shorten the reliability testing evaluation time. Combined with the characteristics of the smart production line, the process interaction method [40] is proposed as the modeling method for the discrete-event simulation system. The simulation testing is carried out by boosting the speed of running the simulation clock of the system.

4.1.1. Discrete Event Process Interaction Method

The state of a discrete event system changes at a discrete time-point, and the evolution of the system over time is represented by a series of system images. For the image of time t, it includes not only the system state at time t, but also the current table of all active events and the state information of each event during the propulsion of the system. The formal theory of discrete event dynamic system proposed by Zeigler can constitute the concept of hierarchical and modular abstract simulator [41], which becomes the theoretical basis of simulation system software development and simulation modeling. Its definition is as follows:

$$\varphi = \langle \vartheta, \zeta, \beta, \delta, \upsilon, \kappa a \rangle$$

Among which a discrete event system φ is expressed as the logical set of external event ϑ, output event ζ, sequential state β, state transition description function δ, output function υ, and time-advance function (κa).

According to the idea of the process interaction method, the digital twin models of each equipment and product in the smart production line are regarded as entities in the discrete event system simulation, where the product is a temporary entity and the equipment is a permanent entity. With the dynamic execution of the simulation system, temporary entities are continuously created and reached under the role of permanent entities, which eventually leave the system after interacting with it to complete all activities. The process interaction method implements the relevant processes by setting and executing the future event table and the current event table. The future event table includes the records of events occurring at a future moment, as well as the records of events that have been postponed at the present stage, but the next execution time has been determined. Each record contains information about the current position, the next position, and priority flags. The flow chart of the process interaction method is shown in Figure 2.

4.1.2. Smart Manufacturing System Software Simulation Accelerated Testing

While using the smart production line digital twin model to carry out accelerated testing, the system software to be tested directly interacts with the production line digital twin model following the actual business process data. The digital twin model contains a production-related event queue that records the sequence of events and the execution time. To accurately calculate the total testing time, the simulation acceleration testing uses a fixed-step time-advance mechanism for simulation clock acceleration. For example, a 100-time acceleration multiplier means that, in reality, the simulation clock advances by 1 s every 10 milliseconds to accelerate the execution of the event queue. The interaction process between the system software under testing and some digital twin models and their events is shown in Figure 3.

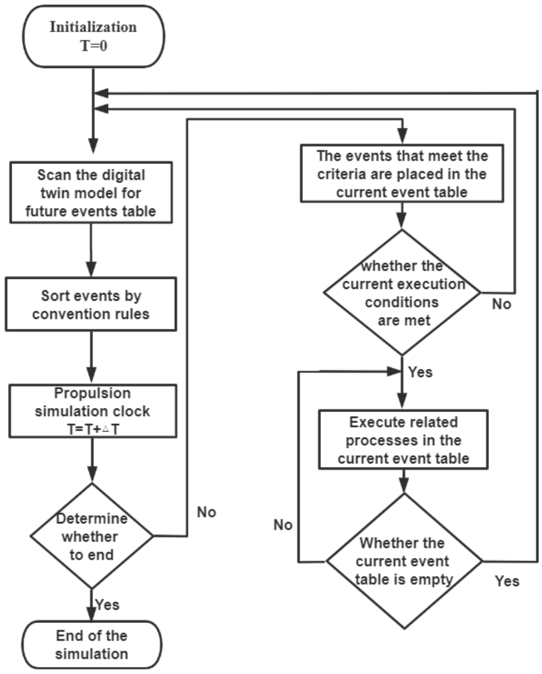

Figure 2. Flow chart of process interaction method.

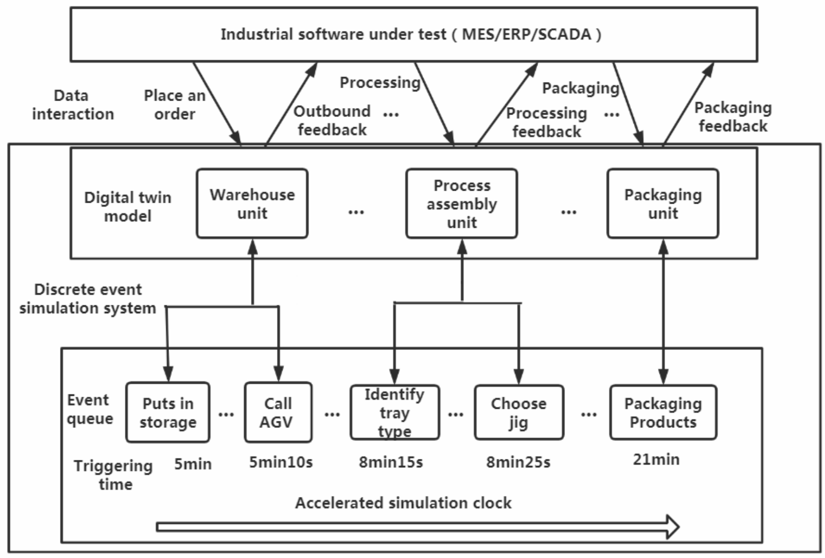

Figure 3. Schematic diagram of manufacturing system software simulation acceleration test.

From the above schematic diagram, it can be seen that the SMS software simulation acceleration test is mainly realized to accelerate the execution speed of the software in the workshop layer. The software to be tested interacts with the digital twin model according to the actual business process. To shorten the response time of the software system waiting for the equipment and improve the communication frequency between the software system and the digital twin equipment, the virtual events in the simulation system can be accelerated to speed up the execution and response of the digital twin device. In the test solution, the reduction of the reliability evaluation verification time is ultimately achieved by reducing the completion time of each business process in the digital twin system.

4.2. Establishing the Digital Twin Model and Testing

To carry out industrial software simulation testing of a manufacturing system, it is necessary to build a highly simulated manufacturing scene. Using the virtual simulation software system, a variety of smart manufacturing scenarios for CPS systems can be built by invoking the internal digital model library. The digital simulation production line is based on the related data of the whole life-cycle of products, which can evaluate and optimize the plant planning, construction, and operation of the production line. The steps of establishing digital twin model of the smart product line by using simulation software such as Demo3D and Flexsim are as follows.

(a) According to the production and processing characteristics of the production line, the 3D model of the equipment is constructed by combining the physical structure of the key equipment. (b) The communication protocol, API interface, and communication address is determined according to the device user manual or description document. (c) The data content of communication between production-line equipment and system software during operation is determined through combination with business processes and functions. (d) The main events in the production process are identified. (e) The connection between the events and the digital twin model of the production line is determined.

Through the above work, the model is basically constructed. Then, the simulation clock acceleration value is set. Selecting the appropriate time-step Δt (in milliseconds) based on the time spent in the actual operation of the production line as the increment when the simulation clock is pushed forward; each advance step is processed as follows.

(1) If no event occurs during the current time, the simulation clock is advanced one unit time-step Δt.

(2) If several events occur during the current time, they are all considered to occur at that moment. Follow the sequence of the events and event queues determined in the above steps.

Assuming that the simulation clock advance times per unit-time is n, the simulation acceleration rate v (times/s) is obtained by the following formula.

$$v = \frac{n \quad \times \quad \Delta t \quad \times \quad 1000}{t} \tag{1}$$

The software to be tested is deployed on the testing server and connected to the digital twin model of the production line equipment; the communication address is configured. According to the user's manual, the instructions for a production run of the software to be tested are determined, and a script is recorded based on the instructions with testing tools. Finally, the digital twin model of the production line is run on the simulation software and the testing script is run cyclically. Relevant information needs to be recorded, such as the system status of the test server at certain time intervals and the time between each system downtime and recovery. At the end of the test, the system status of the test server is recorded and the software running time and the system downtime are counted.

Assuming that the running time of the software to be tested is $t_{runtime}$ and the downtime of the software is $t_{dowmtime}$, the actual mean time between failures ($MTBF$) of the software under test is obtained from the following formula:

$$MTBF = \frac{\sum t_{runtime}(\mathrm{h})}{v(\mathrm{n/s}) \times 3600} \tag{2}$$

The mean time to repair ($MTTR$) of the software to be tested is obtained from the following equation:

$$MTTR = \frac{\sum t_{dowmtime}(\mathrm{h})}{v(\mathrm{n/s}) \times 3600} \tag{3}$$

Then, the reliability A of the software to be tested is:

$$A = \frac{MTBF}{MTBF + MTTR} \tag{4}$$

Downtimes (in hours) of the software to be tested in one year is:

$$Downtimes(\mathrm{h}) = 365 \times 24\ \mathrm{h} \times (1 - A) \tag{5}$$

4.3. Fault-Injection Testing of Smart Manufacturing System Software

In order to adapt to the production needs of new products or to improve production efficiency, enterprises often adjust the amount of equipment or add relevant functions. The reliability and robustness of the software system need to be verified before the industrial software is put into production, to avoid losses due to the failures of industrial software after actual production. Software fault-injection simulates the occurrence of software faults by modifying program execution statements and adding or deleting data or other forms. Fault injection testing of industrial software for SMS is used to verify whether the reliability of the system meets the design requirements and the fault-tolerant design of the system under testing functions is effective when various possible hardware and software defects are activated. By performing fault injection on the digital twin model, hardware damage to the physical production line equipment can be avoided and the testing can also be verified. In the testing system, combined with the fault knowledge-base information, by inputting the modified parameters, the tested software system is run in the digital twin model to check its reliability and fault tolerance, as well as whether the fault can be triggered accurately.

5. Experiments and Discussion

This paper explores the digital twin testing and evaluation method of industrial software for SMS by taking a stepper motor production line as an example. The stepper motor semi-physical simulation production line adopts a combination of virtual and real mode. The physical production line consists of classical discrete manufacturing links in stepper motor assembly and processing. The virtual simulation factory environment is built with the whole production line of the entity as the scenes. So, the virtual factory and the physical production line interact synchronously through digital twin technology.

5.1. Overview of the Demonstrative Semi-Physical Motor Assembly Line

Taking the stepper motor production line as the industrial scene, a semi-physical smart production plant simulation system based on stepper motor processing and assembly will be built. The system structure is shown in Figure 4. In the whole system, the SCADA system is the central hub for data exchange. By installing a variety of sensors on the key equipment of the physical production line, real-time data collection is realized and transmitted to the digital smart-service platform for decision analysis through MES system and SCADA. The physical production line and the virtual production line exchange production-related data through the SCADA system to achieve mutual control. The digital smart service platform and MES system send orders to each station's equipment and the online digital twin modes through SCADA system. Meanwhile, the physical and virtual production lines feed production status data to the digital smart-service management platform. The test control engine collects data from the virtual and physical production lines and detects the production line operation status in real time. At the system terminal, the physical line bus control PLC collects process data from each device and transmits the data to the SCADA system. In addition, the system can transfer the data to the finite element simulation module, and the simulation results will guide the process optimization through the MES system.

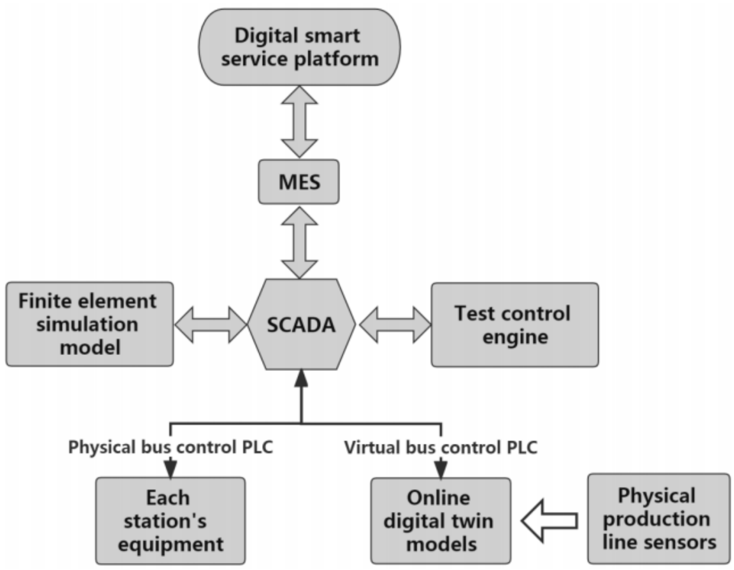

Figure 4. Stepper motor processing assembly testing production line system architecture.

For the stepper motor production line, a semi-physical simulation smart production line based on digital twin is constructed according to the steps described in Section 4.2. Firstly, the 3D models are constructed by combining the physical structure of key equipment such as AGV cars, packaging units, warehouse units, and processing units. The size, position, and shape of the 3D model are in the same proportion as the actual equipment. Then, we determine the communication protocols, interfaces, and communication addresses between the equipment and the software used in the actual production line and set the same protocols and addresses in the model. Thirdly, the specific data content of the production line equipment communicating with the system software or other equipment during operation should be determined. After that, the main events in the production process, such as order placement, material preparation, and processing should be determined. Finally, depending on the production process, the carrier of each event occurrence is determined. The semi-physical simulation model of the production line of the stepper motor is finally constructed, as shown in Figure 5. In the constructed semi-physical simulation production line of the stepper motor, some units are virtual, such as front-end cap group disk unit and rotor group disk unit; some units are physical entities, such as intelligent processing, assembly, inspection, packaging unit, etc. Each station has an RFID, and according to the characteristics of the equipment, temperature, pressure, and electromagnetic sensors are arranged on the 3D printer, loading and unloading robot, and inspection equipment, respectively. Each component unit is controlled and dispatched by the MES system, which ultimately realizes the orderly production of motor products.

The MES software executes the production process in the digital twin production line according to the relevant signals. Each device in the production line has its own status, such as idle, fault, and running. When MES delivers a task, it sends the start signal to the corresponding device. After the device accepts the signal, it modifies its own status to working and then starts to execute the simulation task according to the preset action model. After the task is executed, the state is changed to completed. The status information can be fed back to MES by the device itself or read by MES. Finally, the simulation device can be controlled by the MES system according to the process flow or directly by the MES system through signals.

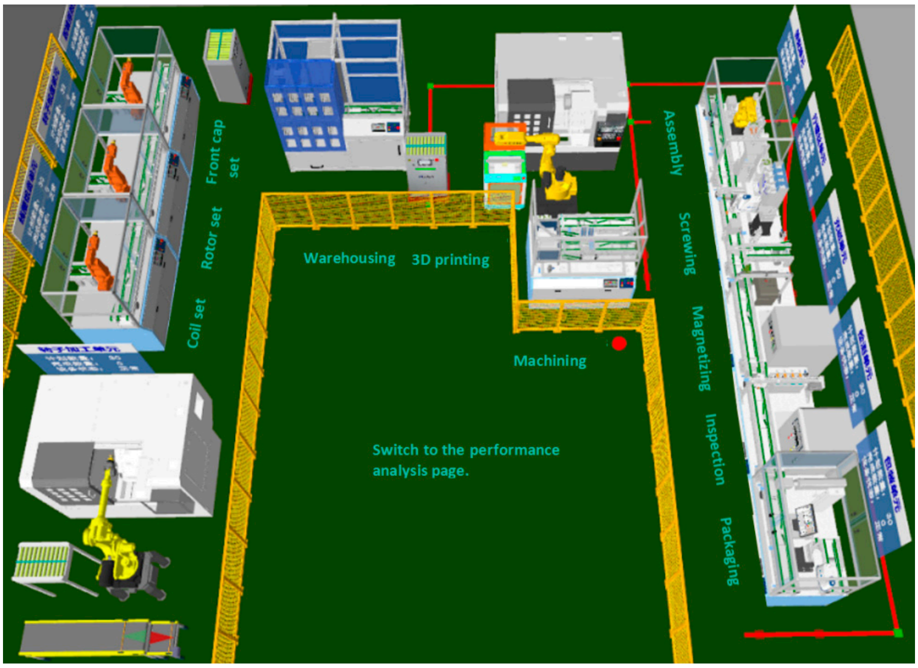

Figure 5. Semi-physical simulation model of stepper motor production line.

5.2. The Digital-Twin-Driven Semi-Physical Testing and Evaluation of Industrial Software

This paper provides a highly simulated and safe operating environment for industrial software with a digital twin system, avoiding possible damage to real production equipment during performance testing and evaluation. The indexes that reflect the performance of industrial software include transaction processing-time, communication interaction time, maximum concurrency, and so on. Through studying the server memory usage and communication interaction time, the reliable performance of the system is reflected. The number of communication interactions refers to the communication frequency between the software system and the production line equipment, which is closely related to the execution time of each business process. When conducting accelerated tests of industrial software, the reliability assessment verification of industrial software is achieved by accelerating the corresponding speed of digital twin devices and improving the execution time of virtual events. The stability of the memory usage also reflects the reliability of the system performance. When industrial software is accelerated and there is no problem with the software design, the memory usage will keep stabilizing after a certain period of time.

To conduct this experiment, there is an assumption that the functions of the system to be tested are strongly correlated with the components of the production line. That is, when performing production tasks, the MES system sends production instructions to the production line through SCADA, and the production line then feeds the results back to the MES system to complete the relevant production requirements. The running scenario of the stepper motor production line is mainly composed of three servers, which run the simulation production line, SCADA system, and MES system, respectively. The simulation line and SCADA use the OPCUA protocol to communication; the SCADA and MES systems use the HTTP protocol, and the whole system runs in a LAN.

The MES software is run on the digital twin model of the stepper motor production line. Then, simulation acceleration testing is carried out using the discrete event process interaction method to accelerate the event execution progress and shorten the system software reliability measurement time by advancing the simulation clock. In the test, we mainly monitor the servers running the MES system software and obtain the test results by calculating the server occupancy rate at different acceleration multipliers. After several tests, the acceleration running speed of the system software in the stepper motor digital twin line model is maintained at a maximum of 200 times operation due to the limitation of the hardware performance of the simulation server. The number of times that the system software communicates with the production line every 5 min under each speed is shown in Figure 6. The system memory occupation of the server where the system software of the workshop is located at different speeds is shown in Figure 7.

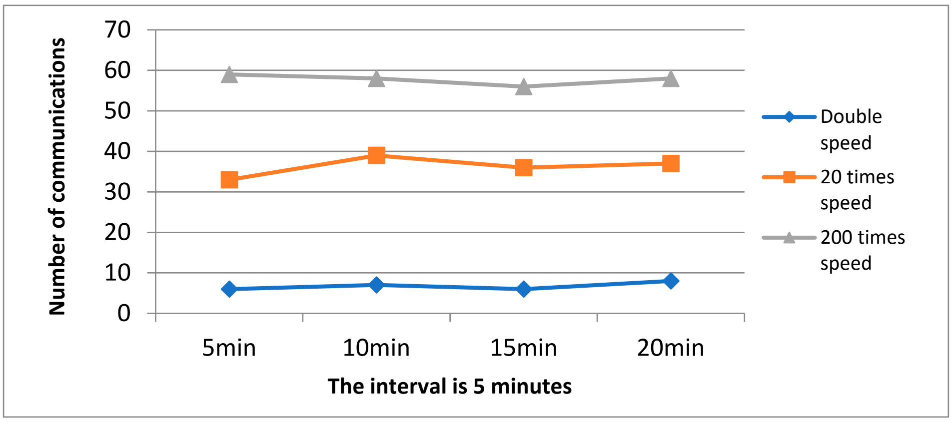

Figure 6. Communication time between system and production line under different acceleration test conditions.

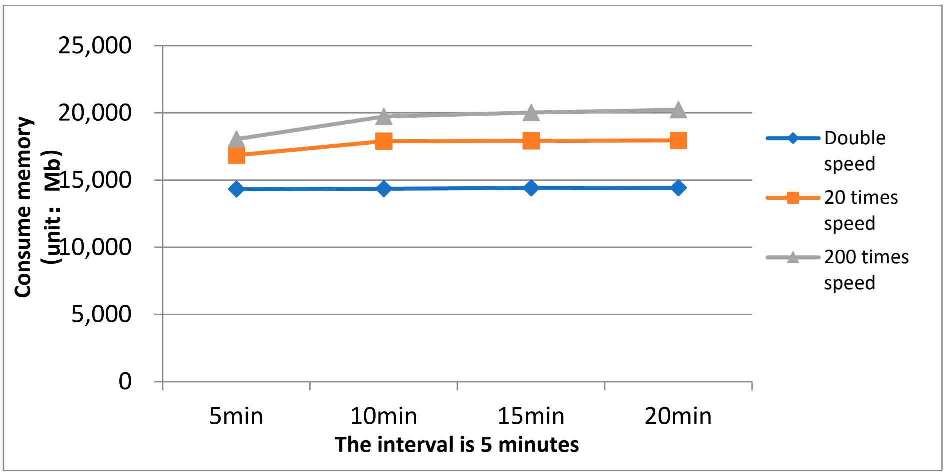

Figure 7. System memory usage during acceleration tests at different multipliers (unit: Mb).

From the testing results, the number of communications between the system software and the production line and the memory consumption of the server varies significantly under the effect of different acceleration multipliers. It can be seen that with the speed multiplier of 1, 20, and 200 times, the memory occupied by the running system software is significantly increased and the number of communications is increased with the increase in the acceleration multiplier in the same testing time. Increasing the system running speed is equivalent to accelerating the aging speed of the system software. So, through accelerated testing, some factors affecting software reliability, such as the exhaustion of operating system resources, fragmentation, and the accumulation of errors also accelerate accordingly, which helps to uncover the vulnerabilities existing in the system software in a shorter period. In addition, when conducting acceleration tests, the acceleration multiplier is mainly limited by the degree of simulation, and the higher the degree of simulation, the closer it is to the real physical equipment. When the degree of simulation is higher, more action logic and fault types need to be included, and the hardware requirements for the server running the simulation are higher.

In this case, assuming that a year in reality is the testing cycle and the acceleration multiplier of the simulation acceleration testing is 200 times, the year within the simulation clock is 43.8 h in reality. Throughout the testing process, the system software has errors in the processing of the AGV trolley signal variables due to the scheduling process, which eventually causes the AGV trolley to stop running at the warehouse unit shipping gate. After restarting the software system to clear the data, the software operates normally and the realistic time spent to restart and clear the data is 24 min; that is, the total failure time $t_{failtime}$ is 0.4 h. The *MTBF* and *MTTF* of the software are calculated by the following formula.

$$MTBF = \frac{\sum t_{runtime}}{\sum n_{fail}} = \frac{365 \times 24}{1} = 8760 \text{ (hours per time)}$$

$$MTTF = \frac{\sum t_{runtime} - t_{failtime}}{\sum n_{fail}} = \frac{365 \times 24 - 0.4}{1} = 8759.6 \text{ (hours per time)}$$

5.3. Performance Evaluation under Fault Injection

To further test and evaluate the fault tolerance and robustness of the industrial software of SMSs, the sensitivity of the software system fault triggering is verified by setting the error instruction under the known fault type. In this study, a fault-injection test is conducted for the digital twin model of a stepper motor production line; the developed test system is shown in Figure 8a. In the fault knowledge base, the digital twin device parameters are modified according to known fault types and injected into the execution system to trigger faults. To verify the fault tolerance and reliability of the industrial software to be tested, the digital twin system is executed to verify whether the model can accurately trigger the fault.

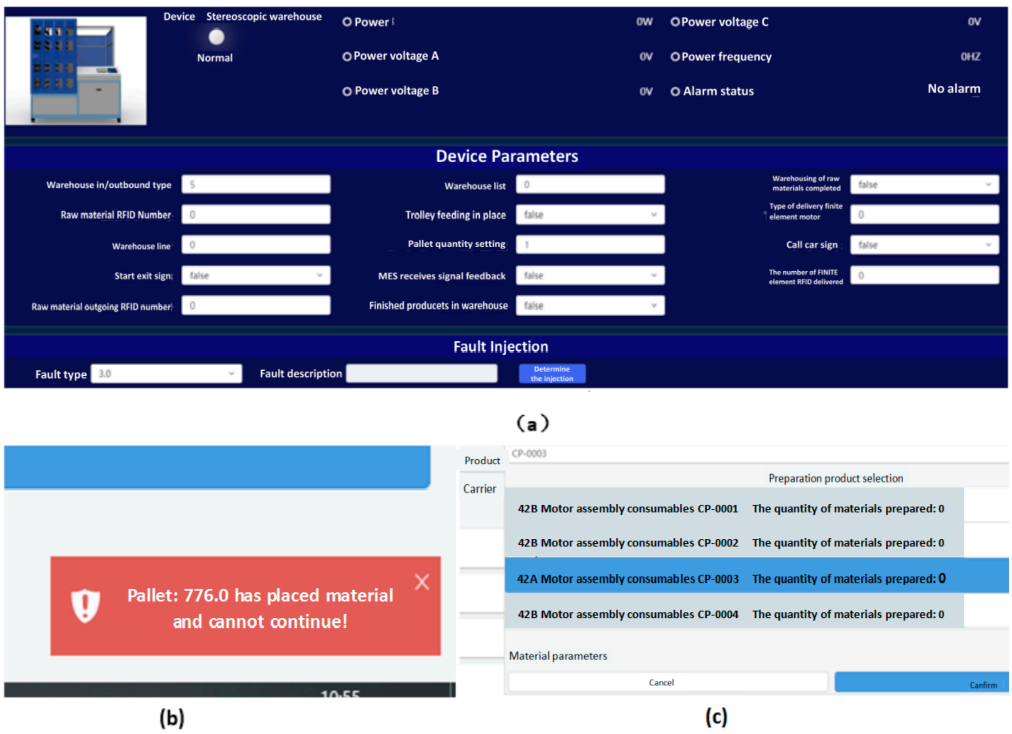

Figure 8. (**a**) Fault injection system; (**b**) pallet material error reported, and (**c**) the actual situation.

When carrying out the fault injection of the digital twin model, the RFID of the pallet of raw materials into the warehouse is manually set to 776.0. Then, the software to be tested issues a preparation command to perform the warehouse preparation operation in the digital twin model, and the system prompts an error: "Pallet: 776.0 has placed material and cannot continue!", as shown in Figure 8b. Actually, there are no pallets in the system and in the warehouse. To check whether the software to be tested reflects the fault in time when it occurs, the system background data are reviewed and it is found that the system has detected the fault and the corresponding fault type after the occurrence of the preparation fault, as shown in Figure 8c. Then, the corresponding measure—namely stopping the preparation—is carried out. This shows that the software under test is fault tolerant and reliable for the preparation process.

6. Conclusions

This paper proposes a digital-twins-based semi-physical simulation testing and evaluation method for industrial software of SMS and discusses the application of digital twin in accelerated testing of the industrial software system of smart manufacturing. By analyzing the characteristics of discrete SMS, the discrete event process-interaction method is proposed to enhance the acceleration effect of the digital twin production line executing the production process in order to achieve the purpose of quickly evaluating the software reliability and robustness of SMS. Through a case study of a stepper motor manufacturing system, a generalized method of digital-twin-based semi-physical simulation testing is proposed, which can meet the accelerated testing requirements of industrial software in a variety of scenarios. The method studied in this paper can save time and cost, improve the accuracy of testing, and can find defects in the trial run phase of industrial software, which reduces the business risk of industrial software. At the same time, the method can also be used for production line renovation and upgrade of intelligent workshops and trial run analysis of new products before they are put into production. In the future, we will consider optimizing the simulation system to further improve the acceleration multiplier. At the same time, more performance indicators will be considered to carry out testing of complex software systems to comprehensively evaluate the software performance.

Author Contributions: K.C. was responsible for literature search, experimental design, and manuscript writing. Q.W. and D.Y. were involved in experimental design, data analysis, and manuscript review. Q.D. and M.W. were involved in study design, data analysis, etc. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Key R&D Program of China (no. 2018YFB1802403), National Nature Science Foundation of China (no. U1701266) and Guangdong Provincial Key Laboratory under Grant (no. 2018B030322016).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Zhao, Y.; Xia, M.; Chu, J. Designing and management of test data on spot testing for process industrial software. *J. Zhongyuan Univ. Technol.* **2018**, *29*, 58–62.
2. Cai, A.; Jiang, Z.; Guo, S. Development Status and Trends of Hardware-in-the-Loop Simulation Technology in Industry. *Aerosp. Control* **2018**, *36*, 52–56.
3. Yu, X.X.; Wang, D.D.; Zhao, Z. Research Progress of Semi-physical Verification Technology Based on Photoelectric Sensing. In *Semi-Physical Verification Technology for Dynamic Performance of Internet of Things System*; Springer: Singapore, 2019. [CrossRef]
4. Yang, J.; Thomas, A.G.; Singh, S.; Baldi, S.; Wang, X. A semi-physical platform for guidance and formations of fixed-wing unmanned aerial vehicles. *Sensors* **2020**, *20*, 1136. [CrossRef] [PubMed]
5. Kaliński, K.J.; Galewski, M.A. Vibration surveillance supported by hardware-in-the-loop simulation in milling flexible workpieces. *Mechatronics* **2014**, *24*, 1071–1082. [CrossRef]
6. Quérel, C.C.; Grondin, O.O.; Letellier, C. Semi-physical mean-value NOx model for diesel engine control. *Control Eng. Pract.* **2015**, *40*, 27–44. [CrossRef]
7. Xiong, X.; Xu, H.; Wu, M.; Lai, X. Cloud manafacturing simulation platform for sintering production process control. *Comput. Intergrated Manuf. Syst.* **2012**, *18*, 1627–1636.
8. Liu, Y.; Zhou, K.; Wan, J.; Wan, G.; Tong, M. A Novel Semi-physical Simulation Platform for Train Braking System Based on Cascade Control Strategy. In Proceedings of the 2021 Photonics & Electromagnetics Research Symposium (PIERS), Hangzhou, China, 21–25 November 2021; pp. 144–148.

9. Ying, Z.D.; Wan, G.C.; Liu, W.J.; Tong, M.S. Simulation modeling and interface parameter design of the semi-physical braking intelligent test system. *Int. J. Numer. Model.* **2019**, *32*, e260. [CrossRef]
10. Chen, D.; Ni, J.; Bai, L.; Chen, D. Evaluation method for the performance of light screen array measurement system based on semi-physical simulation. *Optik Int. J. Light Electron Opt.* **2019**, *178*, 884–891. [CrossRef]
11. Elsisi, M.; Tran, M.Q.; Mahmoud, K.; Mansour, D.E.A.; Lehtonen, M.; Darwish, M.M. Effective IoT-based Deep Learning Platform for Online Fault Diagnosis of Power Transformers against Cyberattack and Data Uncertainties. *Measurement* **2022**, *190*, 110686. [CrossRef]
12. Elsisi, M.; Tran, M.Q. Development of an IoT Architecture Based on a Deep Neural Network against Cyber Attacks for Automated Guided Vehicles. *Sensors* **2021**, *21*, 8467. [CrossRef]
13. Leng, J.; Zhang, H.; Yan, D.; Liu, Q.; Chen, X.; Zhang, D. Digital twin-driven manufacturing cyber-physical system for parallel controlling of smart workshop. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 1155–1166. [CrossRef]
14. Leng, J.; Liu, Q.; Ye, S.; Jing, J.; Wang, Y.; Zhang, C.; Zhang, D.; Chen, X. Digital twin-driven rapid reconfiguration of the automated manufacturing system via an open architecture model. *Robot. Comput. Integr. Manuf.* **2020**, *63*, 101895. [CrossRef]
15. Famelis, M.; Salay, R.; Chechik, M. Partial models: Towards modeling and reasoning with uncertainty. In Proceedings of the 34th International Conferences on Software Engineering, Zurich, Switzerland, 2–9 June 2012; IEEE Press: Piscataway, NJ, USA, 2012; Volume 1, pp. 573–583.
16. Yan, Y. *A Novel Variance Analysis of Software Aging Problems*; IET Software: London, UK, 2018.
17. Huang, Y.; Kintala, C.; Kolettis, N.; Fulton, N.D. Software rejuvenation: Analysis, module and applications. In Proceedings of the 25th International Symposium on Fault-Tolerant Computing, Pasadena, CA, USA, 27–30 June 1995; IEEE Computer Society Press: Piscataway, NJ, USA, 1995; pp. 381–390.
18. Jia, S.; Hou, C.; Wang, J. Software Aging Analysis and Prediction in a Web Server Based on Multiple Linear Regression Algorithm. In Proceedings of the 9th IEEE International Conference on Communication Software & Networks, Guangzhou, China, 6–8 May 2017; IEEE Press: New York, NY, USA, 2017; pp. 1452–1456.
19. Araujo, J.; Matos, R.; Maciel, P.; Vieira, F.; Matias, R.; Trivedi, K.S. Software Rejuvenation in Eucalyptus Cloud Computing Infrastructure: A Method Based on Time Series Forecasting and Multiple Thresholds. In Proceedings of the 3rd International Workshop on Software Aging and Rejuvenation, Hiroshima, Japan, 29 November–2 December 2011; IEEE Press: Piscataway, NJ, USA, 2011; pp. 38–43.
20. Jeff, T. *Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement*; John Wiley & Sons: Hoboken, NJ, USA, 2005.
21. Just, R.; Schweiggert, F. Automating unit and integration testing with partial oracles. *Softw. Qual. J.* **2011**, *19*, 753–769. [CrossRef]
22. Stahl, D.; Bosch, J. Modeling continuous integration practice differences in industry software development. *J. Syst. Softw.* **2014**, *87*, 48–59. [CrossRef]
23. Tao, F.; Cheng, J.; Qi, Q.; Zhang, M.; Zhang, H.; Sui, F. Digital twin-driven product design, manufacturing and service with big data. *Int. J. Adv. Manuf. Technol.* **2018**, *94*, 3563–3576. [CrossRef]
24. Glaessgen, E.; Stargel, D. The digital twin paradigm for future NASA and US air force vehicles. In Proceedings of the 53rd Structures, Structural Dynamics and Materials Conference, Honolulu, HI, USA, 23–26 April 2012; AIAA Press: Reston, VA, USA, 2012; pp. 2–14.
25. Rajratna, K.; Vikas, B.; Santosh, J. Digital twin: Manufacturing excellence through virtual factory replication. *Glob. J. Eng. Sci. Res.* **2014**, 6–15.
26. Negri, E.; Fumagalli, L.; Macchi, M. A Review of the Roles of Digital Twin in CPS-based Production Systems. *Procedia Manuf.* **2017**, *11*, 939–948. [CrossRef]
27. Liu, Q.; Leng, J.; Yan, D.; Zhang, D.; Wei, L.; Yu, A. Digital twin-based designing of the configuration, motion, control, and optimization model of a flow-type smart manufacturing system. *J. Manuf. Syst.* **2021**, *58*, 52–64. [CrossRef]
28. Leng, J.; Wang, D.; Shen, W.; Li, X.; Liu, Q.; Chen, X. Digital twins-based smart manufacturing system design in Industry 4.0: A review. *J. Manuf. Syst.* **2021**, *60*, 119–137. [CrossRef]
29. Ge, Y.; Wang, Y.; Han, Q. Test method of connected and automated vehicles based on digital twin. *ZTE Technol. J.* **2020**, *26*, 25–29.
30. Angjeliu, G.; Coronelli, D.; Cardani, G. Development of the simulation model for Digital Twin applications in historical masonry buildings: The integration between numerical and experimental reality. *Comput. Struct.* **2020**, *238*, 106282. [CrossRef]
31. Leng, J.; Yan, D.; Liu, Q.; Xu, K.; Zhao, J.; Shi, R. ManuChain: Combining Permissioned Blockchain With a Holistic Optimization Model as Bi-Level Intelligence for Smart Manufacturing. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *50*, 182–192. [CrossRef]
32. Oluwasegun, A.; Jung, J. The application of machine learning for the prognostics and health management of control element drive system. *Nucl. Eng. Technol.* **2020**, *52*, 2262–2273. [CrossRef]
33. Euegel, E.J.; Ingraffea, A.R.; Eason, T.G.; Spottswood, S.M. Reengineering Aircraft Structural Life Prediction Using a Digital Twin. *Int. J. Aerosp. Eng.* **2011**, *2011*, 154798.
34. Tao, F.; Zhang, M.; Liu, Y.; Nee, A.Y.C. Digital twin driven prognostics and health management for complex equipment. *CIRP Ann.* **2018**, *67*, 169–172. [CrossRef]
35. Leng, J.; Zhou, M.; Xiao, Y.; Zhang, H.; Liu, Q.; Shen, W. Digital twins-based remote semi-physical commissioning of flow-type smart manufacturing systems. *J. Clean. Prod.* **2021**, *306*, 127278. [CrossRef]

36. Tao, F.; Zhang, M. Digital Twin Shop-Floor: A New Shop-Floor Paradigm Towards Smart Manufacturing. *IEEE Access* **2017**, *5*, 20418–20427. [CrossRef]
37. Liu, Q.; Zhang, H.; Leng, J.; Chen, X. Digital twin-driven rapid individualised designing of automated flow-shop manufacturing system. *Int. J. Prod. Res.* **2018**, *57*, 3903–3919. [CrossRef]
38. Zhang, H.; Liu, Q.; Chen, X.; Zhang, D.; Leng, J. A Digital Twin-Based Approach for Designing and Multi—Objective Optimization of Hollow Glass Production Line. *IEEE Access* **2017**, *5*, 26901–26911. [CrossRef]
39. Guo, D.; Zhong, R.Y.; Lin, P.; Lyu, Z.; Rong, Y.; Huang, G.Q. Digital twin-enabled Graduation Intelligent Manufacturing System for fixed-position assembly islands. *Robot. Comput. Integr. Manuf.* **2020**, *63*, 101917. [CrossRef]
40. Goo, B.; Chung, H.; Han, S. Layered discrete event system specification for a ship production scheduling model. *Simul. Model. Pract. Theory* **2019**, *96*, 101934. [CrossRef]
41. Zeigler, B.P.; Herbert, K.; Tag, G. *Theory of Modeling and Simulation*, 2nd ed.; Academic Press: Cambridge, MA, USA, 2000.