Article

Digital Twin and Reinforcement Learning-Based Resilient Production Control for Micro Smart Factory

Kyu Tae Park [1,2], **Yoo Ho Son** [1], **Sang Wook Ko** [2] **and Sang Do Noh** [1,*]

1 Department of Industrial Engineering, Sungkyunkwan University, Suwon-si 16419, Korea; parkkyutae0201@gmail.com (K.T.P.); sonyooho92@gmail.com (Y.H.S.)
2 Digital Factory Solution R&D Center, MICUBE Solution Inc., Seoul 06719, Korea; swko@micube.co.kr
* Correspondence: sdnoh@skku.edu; Tel.: +82-31-290-7603; Fax: +82-31-290-7610

Abstract: To achieve efficient personalized production at an affordable cost, a modular manufacturing system (MMS) can be utilized. MMS enables restructuring of its configuration to accommodate product changes and is thus an efficient solution to reduce the costs involved in personalized production. A micro smart factory (MSF) is an MMS with heterogeneous production processes to enable personalized production. Similar to MMS, MSF also enables the restructuring of production configuration; additionally, it comprises cyber-physical production systems (CPPSs) that help achieve resilience. However, MSFs need to overcome performance hurdles with respect to production control. Therefore, this paper proposes a digital twin (DT) and reinforcement learning (RL)-based production control method. This method replaces the existing dispatching rule in the type and instance phases of the MSF. In this method, the RL policy network is learned and evaluated by coordination between DT and RL. The DT provides virtual event logs that include states, actions, and rewards to support learning. These virtual event logs are returned based on vertical integration with the MSF. As a result, the proposed method provides a resilient solution to the CPPS architectural framework and achieves appropriate actions to the dynamic situation of MSF. Additionally, applying DT with RL helps decide what-next/where-next in the production cycle. Moreover, the proposed concept can be extended to various manufacturing domains because the priority rule concept is frequently applied.

Keywords: digital twin; production control; micro smart factory; modular manufacturing system; resilience; reinforcement learning

1. Introduction

Personalized production has become the core paradigm in manufacturing research owing to the need for highly diversified products [1,2]. Customized products with affordable quality, cost, and delivery can be manufactured via this production process to meet customer requirements [1–5]. To realize this personalized production, the following three limitations need to be addressed: access, cost, and performance hurdles [2,3,6–8]. Among these hurdles, cost and performance are closely correlated. The access hurdle pertains to the difficulty in accurately judging customer needs through customer interaction; cost hurdle includes increase in cost due to more complex manufacturing systems; and performance hurdle involves performance degradation caused by the complexity of the production process, dynamic situation, and increased preparation time [2,3,6–10]. Additionally, personalized production needs to employ make to order (MTO) entirely or partly. Because the MTO production environment cannot handle inventory, which allows managing fluctuations within certain margins, it is necessary to address these limitations [7,10,11].

Modular manufacturing systems (MMSs) enable the management cost hurdles, and the concept of resilience helps overcome the performance hurdles [10,12–14]. The realization of MMS is expected to restructure the manufacturing system rapidly and easily and enable personalized production of highly diversified products [14]. In addition, the

MMS has an advantage in terms of the cost of product change in production and is suitable for frequent changes related to production [14,15]. Moreover, the elements in the physical work center are managed by a module with independent functional units [14]. Resilience is a characteristic that corresponds to robustness and prevents the degradation of performance indicators. Impermissible events that can cause bullwhip and ripple effects are managed under the resilient production control [10,12,15]. Resilience needs to satisfy five core functional requirements for handling the events: (1) action selection, (2) key performance indicator (KPI) measurement, (3) monitoring, (4) fluctuation notification, and (5) adjustment [12]. In the production control perspective, requirements 1, 2, and 5 correspond to achieving resilience [10].

By configuring the MMS for personalized production, the micro smart factory (MSF) produces personalized products [2,5]. The MSF acts as the work center for production of personalized products that are requested by factory as a service (FaaS) platform [2,5,16]. Additionally, the MSF operates with cyber-physical production system (CPPS) to achieve resilience [7]. CPPS establishes and revises the production plan and schedule, provides time-machine monitoring, and extracts off-line programming (OLP) codes to the physical MSF [5,7]. These technical functionalities satisfy the five core requirements for achieving resilience and support the production control [7].

Although the MSF overcomes the cost hurdle and CPPS solves the performance hurdle in the personalized production perspective, there are still limitations that need to be solved. MMS often reconstructs its configuration to accommodate product change and the control-related characteristics [14,17,18]. To enable efficient production operation of MMS, these limitations must be overcome [17,18]. In addition, three functional requirements, as mentioned above, are necessary to achieve resilience [10]. Thus, the production control of MMS needs to select an action with restructured configuration, measure the KPI, and adjust the rule for selecting action with revised plans and schedules. In particular, adjusting the rule to select action needs to be executed when the impermissible events are detected, and the reactive plan and schedule are established for handling such events.

To consider the abovementioned roles, the dispatching rule for production control is also changeable. It is necessary to enable efficient production even after a restructured product in production is changed, and production plan and schedule are revised [10,14,17,19]. However, the core functional requirements for resilience cannot be achieved by the traditional heuristic rule [19]. Therefore, a novel method for the production control of MMS needs to be proposed. The reinforcement learning (RL) technique enables the adjustment of parameters by repeating episodes to learn the policy network [19–21]. This technique is often applied to select a robust action in response to stochastic arrival by replacing the dispatching rule in a work center [19,22]. To support the adjustment process without user intervention, a model that reflects the accurate systematic behavior of a work center is required [19,23,24].

To overcome the limitations of CPPS in MSF, we propose a digital twin (DT) and RL-based design. This method establishes and adjusts the required parameters for production control in MSF when they need to be revised. The RL supports this establishment and adjustment of parameters through the learning process. Moreover, the KPI, which represents the configuration of the physical work center, reflecting functional units, and synchronizing parameters, properties, and current status, can be measured based on the DT simulation. Furthermore, the asset administration shell (AAS) model is applied and inherited by this method for interoperability between DT and RL. The core contents of this study are as follows:

1. The technical requirements for designing the DT and RL-based production control methods to achieve resilience are defined. To define these requirements, the general process of FaaS platform and the existing research studies on FaaS and MSF are analyzed.
2. The CPPS architectural framework that includes the proposed method is revised and proposed. In this CPPS architectural framework, the essential components for

enabling the proposed method are also suggested. These components coordinate with the components for the proposed method.

3. The policy network is designed to provide the appropriate action to the specific state for maximizing the reward. The action is defined with the concept of priority rule for the efficient replacement of the existing dispatching rule in MSF. Further, the dispatching rule is designed to ensure robustness and resilience upon changes to configuration and production operation.

4. Horizontal coordination, which is the service composition between the technical functionality of DT application and RL technique, is designed to enable the RL policy network for MSF. This coordination considers the advanced characteristics of DT that can reflect the current status of MSF. Moreover, the advantage of RL, which includes efficient adjustment for production control, is also reflected in the design.

5. An industrial case study in MSF is performed to verify and validate the proposed method. Three experiments related to the industrial case study are conducted to confirm whether technical requirements are satisfied.

2. Research Background

2.1. Cyber-Physical Production System and Digital Twin

A cyber-physical system (CPS) advances processes in the physical world by approaching, processing, analyzing, and utilizing data through the internet-based connection between the physical world and virtual components [25–28]. Thus, a CPS can be defined as "a physical and engineering system that monitors, controls, coordinates and integrates physical elements by utilizing computing and communication technologies" [26]. Furthermore, a CPPS is a CPS that enhances efficiency of production process of a manufacturing system. A CPPS is defined as:

> "a physical and engineered system, which aggregates resources, equipment and products by interacting between the physical and the cyber world. This system utilizes knowledge about the overall product lifecycle to improve the efficiency of the production process. Here, the interface between the physical and cyber world is used to monitor, control, coordinate, and integrate resources, equipment, and products. Knowledge about the product lifecycle is applied for the operation of CPPS in an appropriate way for a specified time scale. In addition, heterogeneous advanced engineering applications can improve the value added by the operation of CPPS" [26,29,30].

The above definition indicates that any study on CPPS must focus on the composition and interoperation of a complex system, and that the modularization and interoperability of technology and applications with various levels, layers, and scopes are core issues [29,31,32]. This SoS perspective is related to many issues in architectural design and can follow modular architectural design [29,33]. A modular architecture consists of modules with one or several distinct functions that are connected through a simple interface. The overall system behavior is implemented based on the interactions through this interface, which can be loosely or tightly coupled [29,34,35].

DT is an advanced virtual factory that represents a heterogeneous configuration, reflects the functional units, and synchronizes information objects. The advanced attributes of a DT can improve management accuracy and decision-making efficiency. As a core technology of CPPS, DT can be used to achieve cyber-physical integration for work-center-level design and operation. A DT has the following advanced characteristics in comparison with the traditional simulation model [23,36–42]:

*   automatic creation of DT with predefined configurations and functional units,
*   transmission or reception of information from physical assets through vertical integration,
*   advanced process that applies horizontal coordination to advanced engineering applications, and
*   repeated derivation of performance indicators for prediction and diagnosis.

A DT application is a software component for creating, synchronizing, and utilizing a DT. A virtual representation of a DT application (VREDI) is an asset description that supports vertical integration and horizontal coordination. VREDI considers four core advanced characteristics for applying a DT to a work-center-level asset administration shell (AAS), which includes DT-based technical functionality and the concepts of type and instance. The DT virtual representation is an asset description that abstracts the input to the DT application, thereby realizing an object through component-manager-enabled aggregation. The operation module runs the actual DT application; it runs with the DT engine and uses virtual representation-based objects as the input. The DT engine runs according to the creation, synchronization, and utilization procedures of the operation module; therefore, it must be appropriately designed or selected to achieve the required technical functionality. The configuration data library (CDL) stores the composition of the resources for an accurate and quick site simulation; the composition of the resources is divided into base model, metadata, and logic. The logic includes the element logic for simulating the behavior of the elements and the systematic logic for representing the policy between the elements [9–11,19].

The procedures for the operation module can be defined as follows: for procedure creation, the CDL and DT information object is taken as the input to represent the configuration and reflect the functional units of the physical asset. This includes resource-centric, process-centric, and hybrid creations. In the synchronization procedure, information is mapped to the represented configuration and the reflected functional units according to the DT information schema. This includes steps such as snapshot and footprint synchronizations. In the utilization procedure, the technical functionality of the DT is realized through two detailed steps: execution and post-processing. This includes steps such as virtual commissioning, prognostic simulation, reactive simulation, and synchronization-based representation [10,11].

2.2. Asset Administration Shell

The AAS is a key concept of the reference architectural model industry (RAMI) 4.0 in the Industry 4.0 (I4.0) policy devised in [43–45]. RAMI 4.0 is a three-dimensional model that reflects technical and economic attributes; it simply shows the main aspects of different stakeholders and outlines the guidelines for three axes and the required technical functionality. The three axes are the hierarchy level, value stream, and layer [44,46–48]. The hierarchy level is used to assign functions to the components. The value stream allows classification based on the current state of the life cycle, which is divided according to the type and instance. Layers are used to address concerns regarding the interoperability and common understanding of syntax and semantics from different perspectives; they serve as an interface between the physical and cyber worlds.

The core components of an AAS are virtual representation and technical functionality. The 'manifest' is the metadata, and the 'component manager' supports information management to enable loosely coupled integration with the service-oriented architecture (SOA). The most important feature of AAS is that it realizes I4.0 components with various hierarchy levels [44,45]. An AAS can use a web service to refer to the information and functions of another AAS. In addition, a high level of decentralization and object-orientation allows an AAS to dynamically integrate small amounts of information. Factories that become I4.0 components can be accessed and utilized even if they do not match the descriptions and functionalities of their subunits (i.e., equipment and products) [44,45,48].

In this study, an AAS was applied as a reference model to achieve a high level of interoperability and efficient information management between the DT and heterogeneous components. The key characteristics of the SOA principle in AAS were used to support service composition for DT and RL-based resilient production control with loosely coupled integration. Further, the component-manager-enabled support of vertical integration and horizontal coordination establishes robust and efficient RL-based production control. The

application of this AAS concept to the proposed method enables the development and operation quality of the target physical asset.

3. Cyber-Physical Production System for Resilient Personalized Production

FaaS is a service platform and model that supports personalized production. The main purpose of the FaaS platform is to overcome access, cost, and performance hurdles. This platform has six sequential processes to produce and deliver personalized products to the end-customer: (1) the end-customer provides the computer-aided design (CAD) file of the product and requests production order. (2) Based on the CAD file, the engineering experts consult and revise the design of the product. (3) The essential parts are procured from the suppliers. (4) According to the final design of personalized product and the procured parts, the MSF produces the product. (5) The product is shipped after the production operation ends. (6) The final product is delivered to the end-customer [2,5].

To ensure successful operation of the FaaS platform, studies have been conducted to address the three limitations specified above. To solve the access hurdle, the customers and engineering experts interact through a web client in steps 1–2 of the abovementioned process. In a previous study, the CAD model was uploaded to derive a bill of materials (BOM) from a client [5,49]. Furthermore, 3D printing machines have been proposed to address the cost hurdle in step 4. This is because several different products can be more easily produced via the proposed method than the traditional mold manufacturing method. Thus, the MSF is included as the work center for step 4 of generation process of FaaS platform. In addition, the MSF in FaaS allows for post-processing rather than only providing outputs; thus, it can be configured to generate products based on customer requirements with limited facilities [5,16].

Several studies have been conducted to mitigate the performance hurdle. Kang et al. [16] used the DT to improve the layout and logistics of the MSF so that the transport robots can produce a variety of products and respond to different scenarios. Park et al. [2] implemented a DT through vertical integration between factory sites and information systems. This enabled time-machine monitoring of the entire MSF, which includes past tracking, *real-time monitoring*, and future predictions. In our previous work, the CPS service composition was studied in terms of an SoS rather than as a stand-alone application. Five service-composition-based technical functionalities for problem solving in MSF were defined. *Production planning and scheduling*, and *automated execution* are the technical functionalities performed in the production operation planning stage. The remaining technical functionalities are included in the production execution stage, also referred to as the instance stage. The criteria for determining work-center-level abnormalities include determining if the due date is being met and if there are any problems with specific performance indicators [7]. These criteria form part of the *abnormal situation notification*. The five service-composition-based technical functionalities, implemented using DT through horizontal coordination, which is one of the requirements for DT in MSF, are as follows:

- *Production planning and scheduling*: It involves determining the production plan based on orders that are input/fed from the FaaS service platform.
- *Automated execution*: It involves deriving and executing OLP instructions for executing the production plan.
- *Real-time monitoring*: It involves synchronization of the MSF status to support the user's decision-making.
- *Abnormal situation notification*: It involves providing notifications of the detected events, such as quality defects, equipment failures, and work-center-level abnormal situations.
- *Dynamic response*: The technical functionality involves deriving and executing alternatives after the occurrence of work-center-level abnormal situations.

The schematic configuration of an MSF is shown in Figure 1. It consists of seven process modules and two types of material handling robots (MHRs). The seven process modules perform additive manufacturing, fumigation, polishing, inspection, packaging, and assembly processes. Furthermore, modules performing the assembly process are divided into two types: Assembly No. 1 with a three-axis robot and Assembly No. 2 with a six-axis robot [5,7,16]. These modules can be controlled by a platform based on IoT devices or middleware [2,5,16]. The two MHRs perform material handling operations in each station, and the six-axis handler executes the production plan according to the first come first served schedule. Further, the tower handler is an MHR with agent decision and determines the dispatching process related to the entities in the buffer and in the post-processing station. Thus, the MSF operates with a single decision-making agent in an MHR. Hence, tower handler is an important component that controls the overall process and system efficiency.
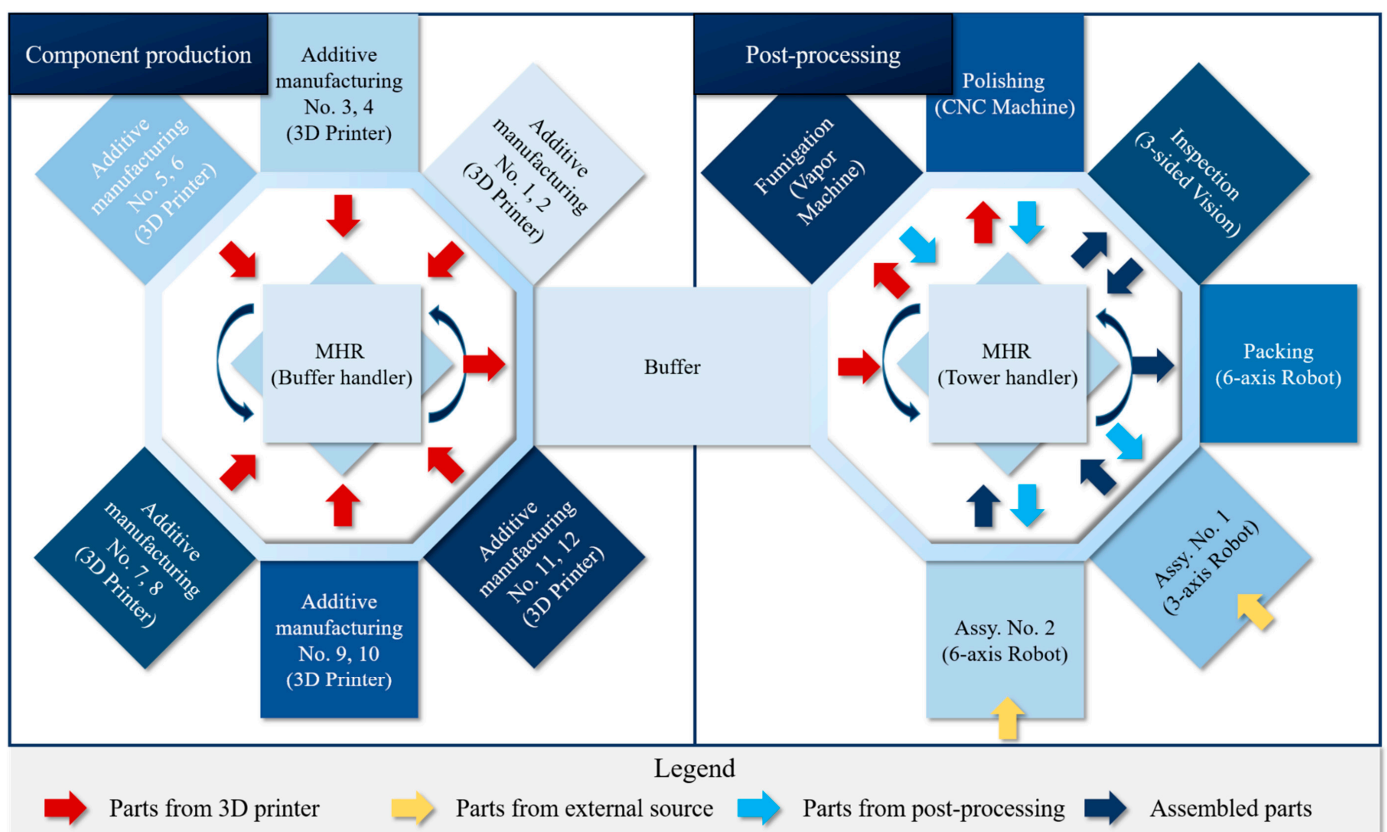

Figure 1. Configuration of a micro smart factory (MSF) [2,5,7,16].

The implemented CPPS and MSF are illustrated in Figure 2 [7]. On the left side of Figure 2, an MSF manufactured by Daejeon-si, Republic of Korea, is shown. On the right in Figure 2, the CPPS for resilient production control is illustrated. The five abovementioned technical functionalities are implemented for the production operation of the MSF based on the DT-based CPPS. The proposed method is also applied to this CPPS for enhancing the dispatching rule of the MSF. Thus, the proposed method addresses the limitation of current research studies on MSF.
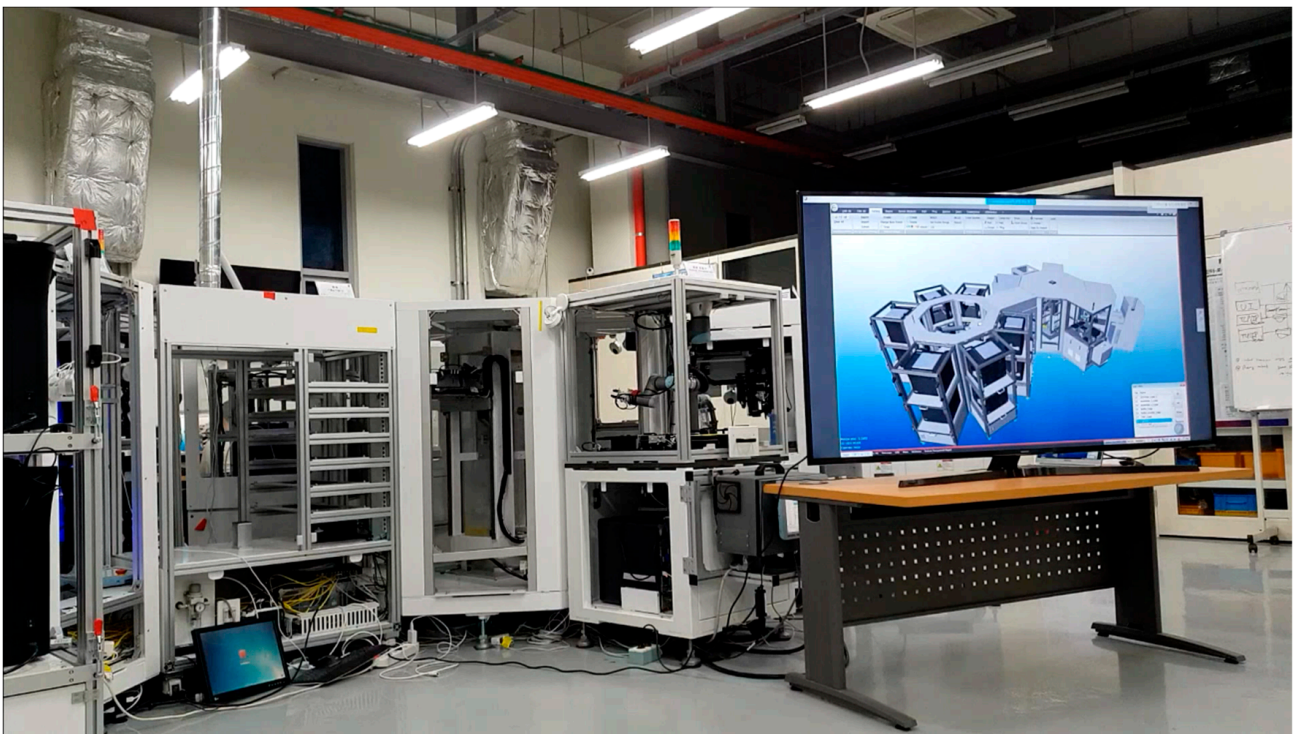
Figure 2. Implemented cyber-physical production system (CPPS) and MSF [7].

4. Method for Resilient Production Control in Modular Manufacturing System

4.1. Problem Definition

Although the MSF, which is an MMS for FaaS platform, is a concept designed to handle the cost hurdle in personalized production, its increased complexity creates performance hurdles. The performance hurdle in the type and instance phases of the work center-level value stream must be solved to achieve production efficiency. As described in the introduction, this includes dynamic selection of parameters, evaluating and improving the dispatching rule, and adjusting the reactive plan and schedule improvement efficiency. The detailed requirements for resilience in MSF are as follows:

- One of the main characteristics of MMS is the ability to restructure. Therefore, the MSF has also the ability to restructure to enhance the production efficiency. From the control perspective, the policy also changes when the configuration is restructured. The number and relationship of elements in the physical work center are also changed, and it is necessary to revise the functional units to enable production operation. Therefore, dynamic selection of parameters is necessary, but the traditional heuristics-based production control cannot respond to this dynamic selection.

- In personalized production, high product diversity affects the management of production operations. The MTO production environment leads to an increase in the complexity of decision-making and control. To overcome this performance hurdle, the dispatching rule for production control must be updated when the product for production is changed. As mentioned above, heuristics-based production control cannot revise this dynamic update.

- To achieve resilience in production control, the core functional requirements need to be satisfied. Action selection, KPI measurement, and adjustment are required for the proposed method. The proper estimation of parameters for selecting action needs to be provided in the operation planning phase, which is the type phase of the instance stage in the work center-level value stream. Dynamic adjustment of parameters for meeting the revised production plan and schedule in the operation execution phase, which is the instance phase of the instance stage in the work center-level value

stream. Furthermore, the KPI needs to be measured for evaluating the policy network alternative in both phases in the instance stage of the work center-level value stream.

- The dynamic estimation of parameters for the reaction to an abnormal situation needs to be synchronized with the current information in the physical work center. Without synchronizing the production operation, the estimated dispatching rule might cause a gap in the physical work center. The production volume, work in process (WIP), machine status, and changed situation are to be synchronized to decrease the gap.

To support the five service-composition-based technical functionalities, *production planning and scheduling*, *automated execution*, and *dynamic response* should be considered to design the method. The *production planning and scheduling* and *dynamic response* are established to plan and schedule to the required time point. In addition, the result of this method is applied to the *automated execution* and needs to consider the tool center points for extracting OLP codes.

4.2. Cyber-Physical Production System Architectural Framework for Resilient Production Control

The proposed method applies DT and RL to satisfy the abovementioned requirements. The DT can provide the evaluation result to the learning process of the policy network. The policy network is an RL-based network model that selects action *a* according to state *s* to maximize reward *r*. In addition, the RL policy network is denoted as $\pi^R(a|s)$ and is learned from the initial solution $\pi^C(a|s)$. Through the learning process, the RL policy network $\pi^R(a|s)$ is adjusted to maximize reward *r*, and the virtual event logs for this network are returned by DT. Moreover, the RL technique enables the estimation of parameters that are suitable for the product diversity in the production operation, the revised plan and schedule, and the current situation of the physical work center.

In the proposed method, the DT plays a role in providing the virtual event trace and KPI for learning the RL policy network $\pi^R(a|s)$. The virtual event trace is the pair of action *a* and state *s* during the DT simulation. RL uses state *s* as inputs and action *a* as an output for indicating the derived entity in the MMS. In addition, the reward *r* is also required for DT application to maximize the specific KPIs from the production control perspective. Moreover, the current information from the physical work center needs to be synchronized to minimize the gap between DT and the physical work center. If the current information, such as progressed production volume, WIP, and machine status, is not considered in the DT simulation, the simulation result might support the learning of RL policy network $\pi^R(a|s)$ with the inappropriate solution space.

To satisfy the abovementioned requirements, the DT application is designed, as shown in Figure 3. The architectural framework follows an AAS model with SOA principles. To enable the interoperability in the heterogeneous development environment, the entire system considers loosely coupled integration based on web services. Following the CPPS architectural framework, which was proposed by Park et al. [7], the advanced planning and scheduling (APS) application and device control application are included. Moreover, the P4R information model is applied for efficient information management and application of 'type and instance' concept based on the VREDI [9]. The following are the detailed descriptions of elements in this architectural framework:

- Component manager: This element is a centralized coordination component and takes the role of a service bus in the SOA principle. The component manager is a subject of vertical integration and horizontal coordination and controls the entire service composition and engineering applications.
- DT application: This is a core element in this architectural framework. The operation module creates, synchronizes, and utilizes DT with DT engine. This application provides simulation-related technical functionalities and visualization according to the request from the service composition.
- Policy generation module: This element learns and deploys the RL policy network using the virtual event logs from the DT application. The RL policy network is learned to maximize reward *r* and is deployed in the format of a systematic logic library (SLL).

- APS application: This application returns the production plan and schedule alternative that needs validation and objective values. The APS algorithm is necessary to establish alternative and simulation-based optimization, metaheuristics, and heuristics can be an option for the core functional engine.
- Device control application: This element extracts the path, kinematics, and estimation related to the robotics configuration. Based on the locations of the MHRs, the required extraction is operated to use forward and backward functions in the simulation.
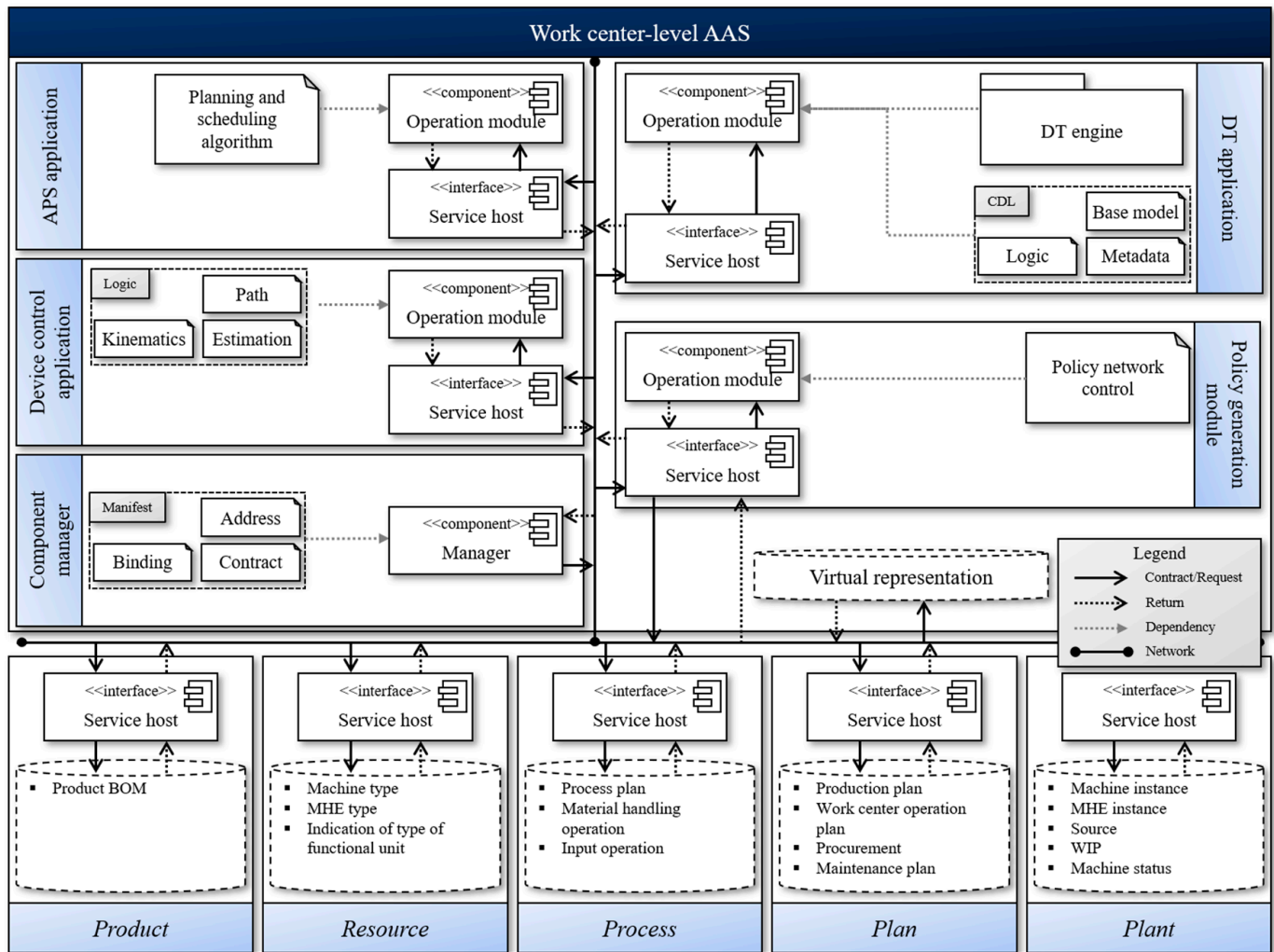

Figure 3. CPPS architectural framework for resilient production control in MSF.

4.3. Policy Network for Production Control in Micro Smart Factory

The policy network is the result of the proposed method. As described above, the RL policy network $\pi^R(a|s)$ is learned based on the virtual event trace, which is a pair of states s and action a. The initial virtual event trace is reported by the DT that reflects the current policy function $\pi^C(a|s)$. In this study, the RL technique for learning is selected for the dueling network technique, which was proposed by Wang et al. [50]. The dueling network technique is an advanced Q-learning technique and has the advantage that the policy network and value network are in the same network. Additionally, the Q-learning-based techniques can be controlled in discrete time and coordinated with discrete event simulation [51–53]. Moreover, the dueling networks separately learn $V(s)$, which is determined only by the state, and the advantage $A(s, a)$, which is determined according to actions, to derive $Q(s, a)$. This approach has the advantage of being able to divide the information

of the Q-function into the portion determined only by the state, and that is determined according to actions. Furthermore, in contrast to a deep Q-network (DQN), it learns the combined weights that lead to $V(s)$ at every step regardless of action. It also requires fewer episodes to complete learning compared to a DQN, which results in better performance as the number of action types increases [50,52,54,55].

With the dueling network exhibiting the abovementioned advantages that make it suitable for application to this method, the Q-function of the RL policy network is presented in Equation (1). In addition, the RL policy network $\pi^R(a_t|s_t)$ selects the action type with the highest Q-function among the actions in step t when the decision of the tower handler in MSF is required. This policy network is designed as a single agent, and it is not necessary to consider coordination between multi-agents.

$$Q(s, a_t) = A(s, a_t) + V(s) \tag{1}$$

$$\pi^R(a_t|s_t) = MAX(Q(s_t, a_t)) \ (\forall i, \forall j, \forall t) \tag{2}$$

As described in Equation (3), the action a_t of each neuron indicates the priority $p_{m,t}$ for what-next, which is for the selection of the part in buffer. Additionally, the configuration of MSF is enabled to restructure, and the number of selectable resource types can be changed. Therefore, because the capacity of all resource types is equal to 1 and the time of material handling operation is not significant, the number of resource instances can be projected to the machine capacity of each resource type u_k. Until the entire resource instances are occupied or all feasible actions are finished, the material handling operation from space m is performed according to the priority $p_{m,t}$.

$$\left\{ o^r_{k,t} \leftarrow o^r_{k,t} + x_{k,t} \middle| \left(o^r_{k,t} \leq u_k \right) \vee (y_{k,m,t} > 0) \right\} \begin{cases} x_{k,t} = 1 \ p_{m,t} = MAX_k(p_{m,t}) \\ x_{k,t} = 0 \ p_{m,,t} \neq MAX_k(p_{m,t}) \end{cases} \tag{3}$$
$$(a_t \ni p_{m,t}, \forall k, \ \forall m)$$

To meet the requirements of the MSF, the state is selected by considering production and delivery. State s includes the remaining production volume $v^r_{m,t}$, remaining due date $d_{i,t}$, the number of WIPs in each resource type $o^r_{k,t}$, the number of WIPs in buffer o^b_t, machine availability $y_{k,t}$ that includes machine failure, processing time $t^p_{i,j,k}$, and setup time $t^s_{i,j,k}$. As illustrated in Equation (4), the information indexed by part i and process j is pre-processed to information with indexing space m. Thus, the state s is projected to two dimensions for the efficient representation.

$$s_t \ni v^r_{m,t}, d_{m,t}, o^r_{k,t}, o^b_t, y_{k,t}, \ t^p_{k,m}, t^s_{k,m} \ (\forall k, \forall m) \tag{4}$$

The reward function is designed to minimize the makespan $C_{max,n}$ and standard deviation of cycle time $\sigma(c_{i,n})$ for enabling the affordable delivery, and to minimize the number of deadlock case k_n for preventing a deadlock. Minimizing the standard deviation of cycle time $\sigma(c_{i,n})$ enables the inspection and packaging process with a constant workload. As shown in Equation (5), the variable r^t_n for deriving the reward variable r_n is calculated based on the three KPIs with normalization. All r^t_n of each episode is recalculated when the episode is finished.

$$\begin{aligned} r^t_n &= [\{C_{max,n} - MIN_n(C_{max,n})\} / \{MAX_n(C_{max,n}) - MIN_n(C_{max,n})\} \\ &+ \{\sigma(c_{i,n}) - MIN_n(\sigma(c_{i,n}))\} / [I \times \{MAX_n(\sigma(c_{i,n})) - MIN_n(\sigma(c_{i,n}))\}] \\ &+ \{k_n - MIN_n(k_n)\} / \{MAX_n(k_n) - MIN_n(k_n)\}] \end{aligned} \tag{5}$$

$$r_n = 1 - r^t_n / MAX_n(r^t_n) \cdot (\forall n) \tag{6}$$

The ending rule for terminating the learning process is designed to confirm the appropriation of learning. The episodes for learning this policy network need to be repeated until the ending value e_n meets the ending limit e^l.

$$\left\{ e_n \leftarrow x_n(e_n + 1) \middle| e_n \leq e^l \right\} \left\{ \begin{array}{ll} x_n = 1 & r_n/MAX_n(r_n) \geq e^w \\ x_n = 0 & r_n/MAX_n(r_n) < e^w \end{array} \right. \tag{7}$$
$$(a_t \ni p_{m,t}, \forall k, \forall m)$$

4.4. Service Composition Procedures to Enable Policy Network

The service composition is a procedure of contacting and receiving the results of heterogeneous components in CPPS. As all components in this CPPS inherit an AAS model with the SOA principle, all cases of interaction between the components receive and return information objects. To support this service composition for learning policy networks between heterogeneous components in CPPS, the virtual event is logged, and results from the DT application are provided to the policy network construction module. Otherwise, the learned policy network after the episode ends, which has to reflect in DT applications.

Figure 4 illustrates the service composition for resilient production control in MSF. This service composition is referenced from the horizontal coordination method for RL-based production control in a re-entrant job shop, which was proposed by Park et al. [19]. Additionally, this service composition procedure is implemented when the production plan and schedule are determined in CPPS. Based on the virtual representation object, the DT application creates the DT with the current policy function $\pi^C(a|s)$ to reflect systematic behavior in MSF. After operation procedures of the DT application, the reported states s, action a, and reward r are delivered to the policy network construction module. Based on the virtual event logs, the module initiates and learns RL policy network $\pi^R(a|s)$, and sends it to the DT application.

The SLL is the point for contacting from the policy network construction module of the DT application. Because the SLL is used to create the procedure in the operation module, the generated RL policy network $\pi^R(a|s)$ is reflected when the DT is created in the DT engine. The virtual event logs, which include information for describing action a, state s, and reward r, are delivered as an information object to the policy network construction module. After the ending rule is satisfied, the *automated execution* technical functionality is requested to derive the OLP codes for controlling MHRs.

This implementation and ending of service composition procedures are identical in the type and instance phases of the instance stage of the work center-level value stream. In the type phase, the *production planning and scheduling*, and *automated execution* technical functionalities are the start and end points of this service composition. In the instance phase, the *dynamic response* technical functionality requires this service composition after the *production planning and scheduling* is determined, and the *automated execution* technical functionality is executed after this service composition is finished.

The learning process of the RL policy network $\pi^R(a|s)$ is the activity for action selection in the type phase and adjustment in the instance phase. This service composition takes the role of action selection with the established production plan and schedule in the type phase. In contrast, this service composition also takes the role of adjustment with *dynamic response* in the instance phase. In addition, the simulation for evaluating and supporting the RL policy network $\pi^R(a|s)$, which is executed in the DT application, supports the action selection and adjustment. Moreover, the aforementioned evaluation is the core activity for KPI measurement.
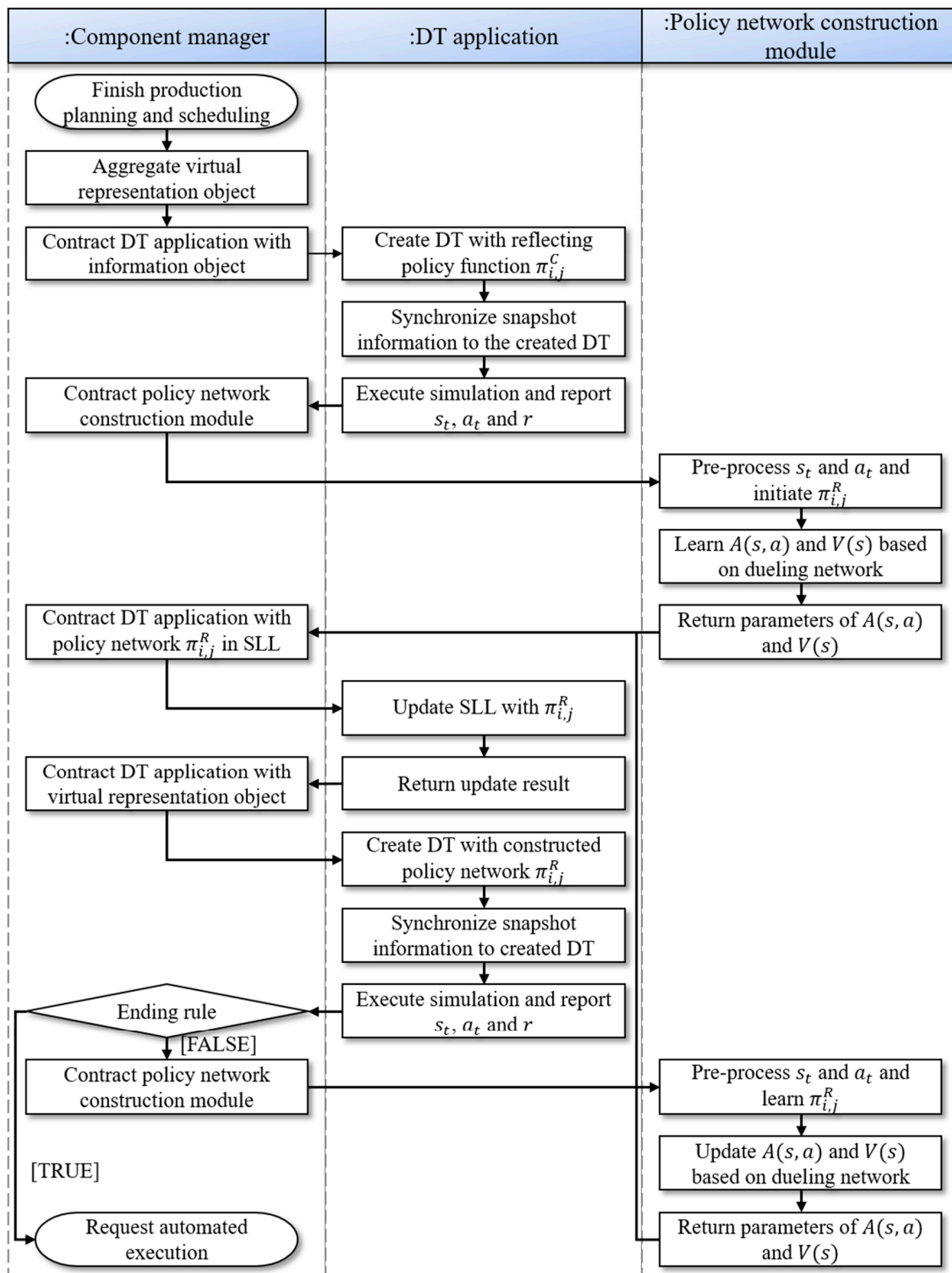
Figure 4. Service composition procedures for enabling policy network (Revised from Park et al. [19]).

5. Industrial Case Study

5.1. Design of Experiments

As shown in Figure 2, the target work center of this experiment was selected as Daejeon-si, Republic of Korea. To supplement the shortage of dispatching in the CPPS, the proposed method is applied to the MSF. To validate the DT and RL-based resilient production control method in MSF, an experiment needs to be designed. The objective values are the makespan $C_{max,n}$, lead time $l_{i,n}$, and the number of deadlock cases k_n. These objective values need to be minimized by the proposed method. As described above, the

makespan $C_{max,n}$ and lead time $l_{i,n}$ are selected to enable affordable delivery of personalized products. The number of deadlock cases k_n is chosen to achieve efficient production control.

The DT and RL-based resilient production control method is proposed to overcome the limitation of MSF, which is an MMS for personalized production. In addition, the proposed method is included in the technical functionalities of CPPS. Therefore, the proposed method needs to be validated from two perspectives. The proposed method needs to improve the efficiency when the configuration of the MSF is changed. This restructuring is the characteristic of MMS and the solution for the cost hurdle.

In the experiment, it is also necessary to demonstrate resilience perspective. The proposed method is realized with the technical functionalities *production planning and scheduling*, and *dynamic response*. For a clear comparison, the results of these technical functionalities are fixed to each case. Additionally, the experiment is divided into two scenarios according to the work center-level value stream. In contrast, the experiment for the reactive production plan and schedule is prepared to validate the proposed method in the instance phase of the instance stage in the work center-level value stream.

To implement the proposed method from the perspective of the restructuring, the cases in which each machine type is added to the MSF are defined, and the performance indicators are compared. To demonstrate resilience in the proposed method, an experiment for a given production plan and schedule is conducted in the type phase of the instance stage in the work center-level value stream. In the instance phase of the instance stage in the work center-level value stream, it is assumed that an event requiring the reaction occurs 48 h after beginning the production operation. When an event occurs, the reactive plan and schedule are executed to solve the event.

5.2. Benchmark Sample and Implementation Information

Table 1 describes the product information for the experiments from two perspectives. The DT and RL-based resilient production control method, proposed in this paper, uses benchmark samples in the experiment. Additionally, these samples are also used in *production planning and scheduling*, and *dynamic response* technical functionalities. The parts that have 'A0' in Part ID are the base modules of assembly. The process plan must be executed to produce the products.

Table 1. Benchmark samples for experiment (time unit: hour).

Product ID	Target Volume	Due Date	Part ID	Process Plan					
P0	30	40	P0A0 P0A1	Building Building	Polishing Polishing	Fumigation Fumigation	Assy. 1 Assy. 1	Inspection	Packaging
P1	40	30	P1A0 P1A1	Building Building	Fumigation Fumigation	Assy. 2 Assy. 2	Fumigation	Inspection	Packaging
P2	25	20	P2A0 P2A1	Building Building	Assy. 1 Assy. 1	Inspection	Packaging		
P3	40	40	P3A0 P3A1 P3A2	Building Building Building	Fumigation Polishing Polishing	Assy. 1 Fumigation Assy. 2	Assy. 2 Assy. 1	Inspection	Packaging
P4	30	40	P4A0	Building	Polishing	Fumigation	Inspection	Packaging	
P5	40	50	P5A0 P5A1	Building Building	Assy. 2 Polishing	Inspection Assy. 2	Packaging		
P6	30	30	P6A0	Building	Polishing	Fumigation	Inspection	Packaging	
P7	40	50	P7A0 P7A1	Building Building	Assy. 1 Fumigation	Inspection Assy. 1	Packaging		
P8	40	40	P8A0	Building	Polishing	Fumigation	Inspection	Packaging	
P9	40	50	P9A0 P9A1	Building Building	Polishing Fumigation	Assy. 1 Assy. 1	Inspection	Packaging	

Table 2 represents the implementation information for an industrial case study. All components coordinate with each other based on the windows communication foundation (WCF) framework. This framework enables a simple object access protocol (SOAP) that satisfies the SOA principle. The extensible markup language (XML) format is applied to the SOAP messages, and the VREDI object for the creation and synchronization of DT. In addition, the DT application uses Plant Simulation as its DT engine to support discrete event simulation for extracting virtual event logs. The SLL for reflecting the RL policy network is formatted in XML. The dueling network technique in PyTorch library in Python is applied in the policy network construction module.

Table 2. Implementation information for industrial case study.

Component	Item	Content
Component manager	Development environment	Visual studio 2019
	Programming language	C#
	Programming model	WCF
	Programming framework	.NET framework 4.7.1
Service hosts	Development environment	Visual studio 2019
	Programming language	C#
	Programming model	WCF
	Programming framework	.NET framework 4.7.1
DT application	Development environment	Visual studio 2019
	Programming language	C#
	Programming framework	.NET framework 4.7.1
	Virtual representation	VREDI
	SLL	XML
	Core functional engine	Plant Simulation 15
Policy network construction module	Development environment	Visual studio 2019
	Programming language	Python 3.7
	Core functional engine	Dueling network (PyTorch)

The control group for comparison is the case with the heuristic rule in the tower handler of the MSF. This heuristic rule is the current rule for production operations in MSF. As described in Equation (8), the workload $w_{k,m,\,t}$ is used as the priority value $p_{k,m,t}$. The large workload $w_{k,m,\,t}$ is prior to being produced to enable efficient production operation. It has a concept similar to the longest processing time (LPT), which is the state-of-the-art heuristics rule.

$$p_{k,m,t} \leftarrow w_{k,m,\,t} = t_{k,m}^p \left(v_{k,m,t}^p + v_{k,m,t}^r \right) / m_k \tag{8}$$

5.3. Experimental Result

The experiments were performed based on the DT application and policy network construction module. The first experiment results for the restructuring of the MMS perspective are summarized in Table 3. Each resource type is added to the empty space in the MSF, and the performance indicators are compared between the proposed method and existing heuristics rule, which is described in Equation (8). The makespan C_{max} is decreased in all cases when the number of machine instances is added. In contrast, the standard deviation of cycle time $\sigma(c_i)$ and the number of deadlock case k are decreased in some cases. Comparing the proposed method with existing heuristics, there is an improvement of 2.585% in makespan $C_{max,n}$, 6.456% in standard deviation of cycle time $\sigma(c_i)$, and 13.953% in the number of deadlock case k in the proposed method. This experiment shows that the proposed method can provide an efficient and robust solution in the case of adding the resource instance.

Table 3. Result of the experiment for the restructure of MMS perspective (unit: hour).

Type	Proposed Method			Existing Heuristics		
	C_{max}	$\sigma(c_i)$	k	C_{max}	$\sigma(c_i)$	k
Current	37.152	0.198	6	38.075	0.211	7
Polishing	34.521	0.187	8	35.877	0.200	9
Fumigation	33.152	0.176	8	33.962	0.188	13
Assy. No. 1	35.645	0.210	9	36.423	0.226	8
Assy. No. 2	36.329	0.215	6	37.154	0.229	6
Average	35.360	0.985	7.4	36.298	0.211	8.6

The results of the second experiment for supporting resilient production control in the type phase of the instance stage of the work center-level value stream are summarized in Table 4. Each case has the same production plan and schedule for comparison according to the benchmark samples. As summarized in Table 4, the makespan C_{max} and the standard deviation of cycle time $\sigma(c_i)$ are decreased in all cases when the production plan and schedule are executed. However, the number of deadlock cases k of has improved in four cases. The proposed method shows an improvement of 3.015% in makespan C_{max}, 8.325% in the standard deviation of cycle time $\sigma(c_i)$, and 9.677% in the number of deadlock cases k. Thus, the proposed method has shown improvement when the *production planning and scheduling* technical functionality is determined, and this resilient production control method is executed in the CPPS.

Table 4. Result of the experiment for resilience in type phase (unit: hour).

Case No.	Proposed Method			Existing Heuristics		
	C_{max}	$\sigma(c_i)$	k	C_{max}	$\sigma(c_i)$	k
1	37.152	0.198	6	38.075	0.211	7
2	36.613	0.200	6	37.810	0.215	6
3	36.492	0.187	5	37.809	0.193	6
4	37.139	0.196	7	38.428	0.218	5
5	37.172	0.199	4	38.184	0.231	7
Average	36.914	0.196	5.6	38.061	0.214	6.2

The last experiment results for supporting resilient production control in the instance phase of the instance stage of the work center-level value stream are described in Table 5. Half of the makespan C_{max} of each case was determined to the time point of the event that decreased production capacity in fumigation. The fumigation module is a bottleneck process for the production operation with the bottleneck process. This event was assumed to be solved in three hours. Moreover, the case numbers are matched to the case number in Table 4. All three performance indicators of the proposed method are better than those of the existing heuristics rule. The proposed method is improved by 4.617% of the makespan C_{max}, 17.468% of the standard deviation of cycle time $\sigma(c_i)$, and 23.529% of the number of deadlock case k. These results show the highest improvement because the proposed method with the synchronization of dynamic situation provides an efficient solution.

5.4. Discussion

The three experiments illustrate the improved performance of the proposed method over the existing heuristics rule described in Equation (7), which is similar in concept to the LPT rule—the state-of-the-art heuristics rule for dispatching. Thus, the experiment can be projected as an experiment between the proposed method and the state-of-the-art heuristics rule that was modified for appropriate application in MSF. In addition, the three experiments verify and validate the three aspects discussed below. The verification is performed based on Plant Simulation, which is the selected DT engine in this study.

Additionally, the three validation aspects are considered from the perspectives of when the configuration of MSF is restructured; when the type phase of work center-level value stream requires resilience for preventing the degradation of performance indicators; and when the instance phase of work center-level value stream also requires the resilience.

Table 5. Result of the experiment for resilience in instance phase (unit: hour).

Case No.	Proposed Method			Existing Heuristics		
	C_{max}	$\sigma(c_i)$	k	C_{max}	$\sigma(c_i)$	k
1	42.534	0.455	10	44.743	0.561	8
2	42.326	0.436	8	43.956	0.549	12
3	41.892	0.426	5	44.113	0.488	10
4	43.074	0.448	9	43.278	0.524	9
5	41.846	0.431	7	45.829	0.539	12
Average	42.334	0.439	7.8	44.384	0.532	10.2

In most cases of each experiment and as shown in Table 6, the makespan C_{max} shows an evident improvement because all cases show an improvement in this indicator. To enable the affordable delivery of personalized products to end-customers, the improvement of the lead time perspective supports this aspect. In addition, the proposed method also shows a relatively constant cycle time to balance the workload of inspection and packaging processes. The last processes that have the appointed capacity can enhance the process and systematic efficiency by balancing the workload. Moreover, the robustness of the proposed method is demonstrated when the resource instance is added as a characteristic of MMS, and the *dynamic response* is performed to prevent performance hurdles because of the events.

Table 6. Result of application of the proposed method (unit: %).

Experiment	Improvement Rate		
	C_{max}	$\sigma(c_i)$	k
Restructure of MMS	2.585	6.456	13.953
Resilience in type phase	3.015	8.325	9.677
Resilience in instance phase	4.617	17.468	23.529

6. Conclusions

To improve the CPPS for enhancing the process and systematic efficiency of MSF, the DT and RL-based resilient production control methods are proposed in this paper. This method enables learning of the RL policy network that replaces the dispatching rule in the post-processing station of MSF. To design an efficient method, the technical requirements are defined. Because of the restructuring characteristic of MMS, the robustness needs to be considered. Additionally, the MTO production environment of personalized production increases the complexity of MSF. Moreover, the technical functionalities of CPPS in MSF must be considered in the design to achieve resilience. Furthermore, dynamic information, such as progress production volume, WIP, machine status, and changed situation, needs to be synchronized in the DT.

With the technical functionalities in CPPS, this method is implemented based on the coordination between the DT application and policy network construction module. The DT application creates, synchronizes, and utilizes the DT for providing DT simulation as its technical functionality. The DT simulation provides the virtual event logs for supporting the learning process of the RL policy network. In contrast, the proposed policy network construction module learns the RL policy network using the dueling network technique. Based on the action, state, and reward in the virtual event logs, the RL policy network is learned and applied. The creation procedure of DT application reflects the RL policy

network repeatedly, and the utilization procedure of the DT application evaluates the RL policy network.

The proposed method has several aspects of originality, contribution, and findings. This method is an early case of coordination between DT and RL. Using the advanced characteristics of DT, the RL-based production control, which uses the traditional DES, can enhance its robustness and efficiency. The advanced characteristics are vertical integration and horizontal coordination and exhibit the advantage of better representing the environment from a learning perspective. In addition, this study is also an early case of applying priority concepts to decide what-next/where-next with DT and RL. Moreover, the event definition with the CPPS architectural framework can be one of the contributions of the proposed method. The abovementioned aspects were verified and validated in the three experiments. Furthermore, the proposed framework and concept can be extended to an efficient solution in various manufacturing domains because the priority rule concept is frequently applied.

As a further study, the event definition in the concept of end-to-end integration needs to be enhanced. This enhancement needs to consider the business and manufacturing process perspectives in the entire supply chain of personalized production. Because personalized production has an MTO production environment and an agent supply chain system, the decision complexity is increased.

Author Contributions: Conceptualization, K.T.P., Y.H.S. and S.W.K.; Data curation, Y.H.S.; Investigation, S.W.K.; Writing —original draft, K.T.P.; Writing— review & editing, S.D.N. All authors have read and agreed to the published version of the manuscript.

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

Abbreviations

Indices

i	Index denoting a part requiring the production process ($i = 1 \ldots I$)
j	Index denoting a process operation in the process plan ($j = 1 \ldots J$)
k	Index denoting a resource type ($k = 1 \ldots K$)
m	Index denoting a space for a part in buffer ($m = 1 \ldots M$)
n	Index denoting an episode ($n = 1 \ldots N$)
t	Index denoting a step in a matrix of an episode trace ($t = 1 \ldots T$).

Hyper-parameters

δ	Discount factor of policy network

Variables

$a_{n,t}$	Action in step t of episode n
$C_{max,\, n}$	Makespan value of episode n
$c_{i,n}$	Cycle time of part i of episode n
$d_{i,n,t}$	Remained due date of part i in step t of episode n
$d_{m,n,t}$	Remained due date of part in space m in step t of episode
e_n	Ending value of episode n
e^l	Selected ending limit
e^w	Selected ending weight
k_n	Number of deadlock cases of episode n
m_k	Number of resource instances instantiated by resource type k
$o^b_{n,t}$	Number of works in process (WIPs) in the buffer in step t of episode n
$o^r_{k,n,t}$	Number of WIPs in resource type k in step t of episode n
$p_{m,n,t}$	Priority value from space m in step t of episode n

$r_{i,j,n}$	Reward variable of process operation j of part i of episode n
$s_{n,t}$	State in step t of episode n
$t^p_{k,m}$	Processing time of process operation of part in space m in machine type k
$t^s_{k,m}$	Setup time of process operation of part in space m in machine type k
u_k	Capacity of resource type k
$v^p_{k,m,n,t}$	Proceeded production volume of part in space m to resource type k in step t of episode n
$v^r_{i,j,n,t}$	Remained production volume of process operation j of part i in step t of episode n
$v^r_{k,m,n,t}$	Remained production volume of part in space m to resource type k in step t of episode n
$w_{k,m,n,\,t}$	Workload of part in space m to resource type k in step t of episode n
$x_{k,n,t}$	Binary variable for indicating the material handling operation to resource type k in step t of episode n
x_n	Binary variable for calculating the ending value of episode n
$y_{k,n,t}$	Availability of resource type k in step t of episode n
$y_{k,m,n,t}$	Feasibility from space m to resource type k in step t of episode n
Functions	
$A(s,\,a)$	Advantage functions of states s and action a
$Q(s,\,a)$	Q-function of states s and action a
$V(s)$	Value function of state s
$\pi^C(a|s)$	Current policy function in a physical asset.
$\pi^R(a|s)$	RL policy network

References

1. Wiktorsson, M.; Noh, S.D.; Bellgran, M.; Hanson, L. Smart Factories: South Korean and Swedish examples on manufacturing settings. *Procedia Manuf.* **2018**, *25*, 471–478. [CrossRef]
2. Park, K.T.; Nam, Y.W.; Lee, H.S.; Im, S.J.; Noh, S.D.; Son, J.Y.; Kim, H. Design and implementation of a digital twin application for a connected micro smart factory. *Int. J. Comput. Integr. Manuf.* **2019**, *32*, 596–614. [CrossRef]
3. Yao, X.; Lin, Y. Emerging manufacturing paradigm shifts for the incoming industrial revolution. *Int. J. Adv. Manuf. Technol.* **2016**, *85*, 1665–1676. [CrossRef]
4. Mai, J.; Zhang, L.; Tao, F.; Ren, L. Customized production based on distributed 3D printing services in cloud manufacturing. *Int. J. Adv. Manuf. Technol.* **2016**, *84*, 71–83. [CrossRef]
5. Son, J.; Kang, H.C.; Bae, H.C.; Lee, E.S.; Han, H.Y.; Kim, H. IoT-based open manufacturing service platform for mass personalization. *J. Korean Inst. Commun. Sci.* **2015**, *33*, 42–47.
6. Kumar, A. From mass customization to mass personalization: A strategic transformation. *Int. J. Flex. Manuf. Syst.* **2007**, *19*, 533. [CrossRef]
7. Park, K.T.; Lee, J.; Kim, H.-J.; Noh, S.D. Digital-twin-based cyber physical production system architectural framework for personalized production. *Int. J. Adv. Manuf. Technol.* **2020**, *106*, 1787–1810. [CrossRef]
8. Du, X.; Jiao, J.; Mitchell, M.T. Understanding customer satisfaction in product customization. *Int. J. Adv. Manuf. Technol.* **2006**, *31*, 396–406. [CrossRef]
9. Park, K.T.; Yang, J.; Noh, S.D. VREDI: Virtual representation for a digital twin application in a work-center-level asset administration shell. *J. Intell. Manuf.* **2020**, *32*, 501–544. [CrossRef]
10. Park, K.T.; Son, Y.H.; Noh, S.D. The architectural framework of a cyber physical logistics system for digital-twin-based supply chain control. *Int. J. Prod. Res.* **2020**, 1–22. [CrossRef]
11. Park, K.T.; Lee, D.; Noh, S.D. Operation procedures of a work-center-level digital twin for sustainable and smart manufacturing. *Int. J. Precis. Eng. Manuf. Technol.* **2020**, *7*, 791–814. [CrossRef]
12. Ivanov, D. *Structural Dynamics and Resilience in Supply Chain Risk Management*; Springer International Publishing: Cham, Switzerland, 2018. [CrossRef]
13. Ivanov, D.; Dolgui, A.; Sokolov, B. The impact of digital technology and industry 4.0 on the ripple effect and supply chain risk analytics. *Int. J. Prod. Res.* **2019**, *57*, 829–846. [CrossRef]
14. Tsukune, H.; Tsukamoto, M.; Matsushita, T.; Tomita, F.; Okada, K.; Ogasawara, T.; Takase, K.; Yuba, T. Modular manufacturing. *J. Intell. Manuf.* **1993**, *4*, 163–181. [CrossRef]
15. Dolgui, A.; Ivanov, D.; Rozhkov, M. Does the ripple effect influence the bullwhip effect? An integrated analysis of structural and operational dynamics in the supply chain. *Int. J. Prod. Res.* **2020**, *58*, 1285–1301. [CrossRef]
16. Kang, H.S.; Noh, S.D.; Son, J.Y.; Kim, H.; Park, J.H.; Lee, J.Y. The FaaS system using additive manufacturing for personalized production. *Rapid Prototyp. J.* **2018**, *24*, 1486–1499. [CrossRef]
17. Ďurica, L.; Gregor, M.; Vavrík, V.; Marschall, M.; Grznár, P.; Mozol, Š. A route planner using a delegate multi-agent system for a modular manufacturing line: Proof of concept. *Appl. Sci.* **2019**, *9*, 4515. [CrossRef]

18. Kaid, H.; Al-Ahmari, A.; Li, Z.; Davidrajuh, R. Automatic supervisory controller for deadlock control in reconfigurable manufacturing systems with dynamic changes. *Appl. Sci.* **2020**, *10*, 5270. [CrossRef]

19. Park, K.T.; Jeon, S.-W.; Noh, S.D. Digital twin application with horizontal coordination for reinforcement-learning-based production control in a re-entrant job shop. *Int. J. Prod. Res.* **2021**. [CrossRef]

20. Wu, J.; Wei, Z.; Li, W.; Wang, Y.; Li, Y.; Sauer, D. Battery thermal- and health-constrained energy management for hybrid electric bus based on Soft Actor-Critic DRL algorithm. *IEEE Trans. Ind. Inform.* **2020**. [CrossRef]

21. Wu, J.; Wei, Z.; Liu, K.; Quan, Z.; Li, Y. Battery-involved energy management for hybrid electric bus based on expert-assistance deep deterministic policy gradient algorithm. *IEEE Trans. Veh. Technol.* **2020**, *66*, 12786–12796. [CrossRef]

22. Lin, C.-C.; Deng, D.-J.; Chih, Y.-L.; Chiu, H.-T. Smart manufacturing scheduling with edge computing using Multiclass Deep Q Network. *IEEE Trans. Ind. Inform.* **2019**, *15*, 4276–4284. [CrossRef]

23. Mourtzis, D. Simulation in the design and operation of manufacturing systems: State of the art and new trends. *Int. J. Prod. Res.* **2020**, *58*, 1927–1949. [CrossRef]

24. Mourtzis, D.; Vlachou, E. A cloud-based cyber-physical system for adaptive shop-floor scheduling and condition-based maintenance. *J. Manuf. Syst.* **2018**, *47*, 179–198. [CrossRef]

25. Lee, J.; Bagheri, B.; Kao, H.-A. A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manuf. Lett.* **2015**, *3*, 18–23. [CrossRef]

26. Monostori, L.; Kádár, B.; Bauernhansl, T.; Kondoh, S.; Kumara, S.; Reinhart, G.; Sauer, O.; Schuh, G.; Sihn, W.; Ueda, K. Cyber-physical systems in manufacturing. *CIRP Ann.* **2016**, *65*, 621–641. [CrossRef]

27. Sztipanovits, J.; Ying, S. *Foundations for Innovation: Strategic R&D Opportunities for the 21th Century Cyber-Physical Systems*; National Institute of Standards and Technology (NIST): Gaithersburg, MD, USA, 2013; p. 32.

28. Park, K.T.; Kang, Y.T.; Yang, S.G.; Bin Zhao, W.; Im, S.J.; Kim, D.H.; Choi, S.Y.; Noh, S.D.; Kang, Y.-S. Cyber physical energy system for saving energy of the dyeing process with industrial Internet of Things and manufacturing big data. *Int. J. Precis. Eng. Manuf. Technol.* **2020**, *7*, 1–20. [CrossRef]

29. Ribeiro, L.; Björkman, M. Transitioning from standard automation solutions to cyber-physical production systems: An assessment of critical conceptual and technical challenges. *IEEE Syst. J.* **2017**, *12*, 3816–3827. [CrossRef]

30. Ribeiro, L. Cyber-physical production systems' design challenges. In Proceedings of the 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), Edinburgh, UK, 19–21 June 2017; pp. 1189–1194. [CrossRef]

31. Lee, J.; Davari, H.; Singh, J.; Pandhare, V. Industrial artificial intelligence for Industry 4.0-based manufacturing systems. *Manuf. Lett.* **2018**, *18*, 20–23. [CrossRef]

32. Lee, J.; Ardakani, H.D.; Yang, S.; Bagheri, B. Industrial big data analytics and cyber-physical systems for future maintenance & service innovation. *Procedia CIRP* **2015**, *38*, 3–7. [CrossRef]

33. Otto, J.; Vogel-Heuser, B.; Niggemann, O. Automatic parameter estimation for reusable software components of modular and reconfigurable cyber-physical production systems in the domain of discrete manufacturing. *IEEE Trans. Ind. Inform.* **2018**, *14*, 275–282. [CrossRef]

34. Crawley, E.; de Weck, O.; Eppinger, S.; Magee, C.; Moses, J.; Seering, W.; Schindall, J.; Wallace, D.; Whitney, D. The influence of architecture in engineering system. *Monograph* **2004**, *3*.

35. Chiriac, N.; Hölttä-Otto, K.; Lysy, D.; Suh, E.S. Level of modularity and different levels of system granularity. *J. Mech. Des.* **2011**, *133*, 101007. [CrossRef]

36. Grieves, M. *Digital Twin: Manufacturing Excellence through Virtual Factory Replication*; Dassault Systèmes: Vélizy-Villacoublay, France, 2014.

37. Cheng, Y.; Zhang, Y.; Ji, P.; Xu, W.; Zhou, Z.; Tao, F. Cyber-physical integration for moving digital factories forward towards smart manufacturing: A survey. *Int. J. Adv. Manuf. Technol.* **2018**, *97*, 1209–1221. [CrossRef]

38. Qi, Q.; Tao, F. Digital twin and big data towards Smart Manufacturing and Industry 4.0: 360 degree comparison. *IEEE Access* **2018**, *6*, 3585–3593. [CrossRef]

39. Tao, F.; Zhang, H.; Liu, A.; Nee, A.Y.C. Digital twin in industry: State-of-the-art. *IEEE Trans. Ind. Inform.* **2018**, *15*, 2405–2415. [CrossRef]

40. Liu, Q.; Zhang, H.; Leng, J.; Chen, X. Digital twin-driven rapid individualised designing of automated flow-shop manufacturing system. *Int. J. Prod. Res.* **2019**, *57*, 3903–3919. [CrossRef]

41. Ding, K.; Chan, F.T.S.; Zhang, X.; Zhou, G.; Zhang, F. Defining a digital twin-based cyber-physical production system for autonomous manufacturing in smart shop floors. *Int. J. Prod. Res.* **2019**, *57*, 6315–6334. [CrossRef]

42. Lu, Y.; Xu, X.; Wang, L. Smart manufacturing process and system automation—A critical review of the standards and envisioned scenarios. *J. Manuf. Syst.* **2020**, *56*, 312–325. [CrossRef]

43. Dorst, W. (Ed.) *Umsetzungsstrategie Industrie 4.0: Ergebnisbericht der Plattform Industrie 4.0*; Bitkom Research GmbH: Berlin, Germany, 2015.

44. Adolphs, P.; Auer, S.; Bedenbender, H.; Billmann, M.; Hankel, M.; Heidel, R.; Hoffmeister, M.; Huhle, H.; Jochem, M.; Kiele-Dunsche, M.; et al. *Structure of the Administration Shell*; Federal Ministry for Economic Affairs and Energy: Berlin, Germany, 2016.

45. Hankel, M.; Rexroth, B. *Reference Architectural Model Industrie 4.0 (RAMI 4.0)*; Federal Ministry for Economic Affairs and Energy: Berlin, Germany, 2015.

46. Suri, K.; Cadavid, J.; Alferez, M.; Dhouib, S.; Tucci-Piergiovanni, S. Modeling Business Motivation and Underlying Processes for RAMI 4.0-Aligned Cyber-physical Production Systems. In Proceedings of the 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Limassol, Cyprus, 12–15 September 2017; pp. 1–6. [CrossRef]
47. Kagermann, H.; Wahlster, W.; Helbig, J. *Securing the Future of German Manufacturing Industry: Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0*; Acatech: Munich, Germany, 2013.
48. ZVEI. *Examples of the Asset Administration Shell for Industrie 4.0 Components—Basic Part*; German Electrical and Electronic Manufacturers' Association: Frankfurt, Germany, 2017.
49. Do, N. Developing a BOM management system for personal manufacturing. *Korean J. Comput. Des. Eng.* **2017**, *22*, 352–362. [CrossRef]
50. Wang, Z.; Schaul, T.; Hessel, M.; van Hasselt, H.; Lanctot, M.; de Freitas, N. Dueling network architectures for deep reinforcement learning. *arXiv* **2015**, arXiv:1511.06581.
51. Park, I.-B.; Huh, J.; Kim, J.; Park, J. A reinforcement learning approach to robust scheduling of semiconductor manufacturing facilities. *IEEE Trans. Autom. Sci. Eng.* **2019**, *17*, 1420–1431. [CrossRef]
52. Gabel, T.; Riedmiller, M. Adaptive reactive job-shop scheduling with reinforcement learning agents. *Int. J. Inf. Technol. Intell. Comput.* **2008**, *24*, 1–60.
53. Gosavi, A. Reinforcement learning: A tutorial survey and recent advances. *INFORMS J. Comput.* **2009**, *21*, 178–192. [CrossRef]
54. Van Hasselt, H.; Guez, A.; Silver, D. Deep Reinforcement Learning with Double Q-learning. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
55. Nair, A.; Srinivasan, P.; Blackwell, S.; Alcicek, C.; Fearon, R.; de Maria, A.; Panneershelvam, V.; Suleyman, M.; Beattie, C.; Petersen, S.; et al. Massively parallel methods for deep reinforcement learning. *arXiv* **2015**, arXiv:1507.04296.