

On the Path to Biointelligent Production: Developing a Control System for Biointelligent Logistics

Zheng Sun,
Jifeng Zhou,
Aya Ounissi,
Robert Schulz,

*Institute of Mechanical Handling and Logistics (IFT)
Faculty of Engineering Design, Production Engineering and Automotive Engineering
University of Stuttgart*

Increasing demand volatility, resource constraints, and failures of production resources increase the need for production logistics systems that can adapt without requiring a central scheduler to determine every operation assignment. However, the operational transfer of biological control mechanisms into executable production-logistics models remains insufficiently specified. This paper therefore develops, implements, and evaluates a bio-inspired control framework for decentralised and self-organising production logistics. Following the inspiration pathway of biological transformation, the biological mechanisms of self-organisation, stigmergic coordination, and checkpoint-based regulation are abstracted and transferred to production logistics control mechanisms. Production units are represented as autonomous agents with heterogeneous capabilities, operational parameters and emission rates, while orders and transport resources are modelled as dynamic simulation objects within a shared event-driven environment. Temporary task-specific structures are formed through state-dependent assignments between orders, their next required operations, and available and capable production units. Failure and repair events support the interruption and subsequent reassignment, thereby allowing the production logistics network to reconfigure in response to resource unavailability. The resulting agent-based model library and simulation module provide an executable representation of decentralised production logistics control and are applied to cooperative multi-agent reinforcement learning considering time- and emission-related objectives. In the investigated scenario, the learned policy outperforms the considered rule-based dispatching strategies in terms of team reward and achieves lower mean makespan and total CO₂ emissions across the investigated failure settings.

[Keywords: Biointelligent production logistics, Bio-inspired control, Self-organisation, Multi-agent systems]

1 Introduction

Production and logistics systems are increasingly exposed to volatile demand, resource constraints, and disruptions [1–3]. In production logistics, these changes affect not only the availability of manufacturing resources, but also the feasibility of operation assignments, material-flow requirements, transport demand, and the validity of existing schedules. At the same time, industrial value creation is expected to improve productivity and delivery performance while reducing resource consumption and environmental impacts [3, 4]. These partly conflicting requirements increase the complexity of production-logistics control and challenge conventional system architectures. Consequently, machine failures, resource shortages, transport bottlenecks, and changing order priorities may require production plans and resource assignments to be revised during operation [5, 6].

Control approaches based on predefined process chains and centralised scheduling can achieve high efficiency under stable and predictable operating conditions [7, 8]. However, frequent changes in order requirements, resource availability, and transport conditions may require frequent system-wide information updates and rescheduling. More flexible control structures are therefore required to adapt operation or resource assignments to current order states, resource capabilities, availability and logistics constraints. Recent studies in logistics have also demonstrated the potential of deep reinforcement learning for state-dependent decision-making in complex material-flow systems, includ-

ing retrieval, relocation, and sequencing decisions in multi-deep automated storage and retrieval systems [9, 10].

Decentralised and self-organising control structures provide a potential approach for responding to changing production conditions and improving system resilience [7, 11]. Rather than defining a ll o perational r elationships i n advance, production units can be represented as autonomous decision-making agents that perceive information, evaluate currently feasible order assignments, and coordinate their behaviour based on their capability and the current factory state. Temporary task-specific production structures are represented by state-dependent assignments between orders, their next required operations, and capable production resources, taking into account resource capabilities, current availability, and logistics constraints. These temporary assignments end after operation completion and can be reconfigured when resource states or order requirements change. This enables the production-logistics network to adapt its operational configuration without requiring a complete centralised reorganisation.

The biological transformation of industrial value creation provides a promising conceptual basis for such systems [12, 13]. In particular, biological mechanisms such as self-organisation, environment-mediated coordination, and condition-dependent process validation can be abstracted and transferred to production logistics. These principles enable autonomous system elements for local decision-making, temporary resource-order assignments, and event-driven responses without requiring a central scheduler to determine every operation assignment.

Although biological transformation and biointelligent production have been extensively discussed at the conceptual level [12, 14], the operational transfer of biological control mechanisms to executable production logistics systems remains insufficiently specified. In particular, it is necessary to define how biological mechanisms can be mapped onto orders, production sources, transport resources, system states, and local decision-making, and how the resulting interactions can be represented in an event-driven simulation environment. Moreover, production assignments, transport processes, resource failures, and environmental performance need to be represented within a consistent control and simulation framework.

Against this background, this paper investigates how biological control principles can be transferred to decentralised production logistics. The objective of this paper is to develop and implement a bio-inspired control framework that enables production units to coordinate locally, form temporary operational structures, and adapt to changing system conditions. The framework represents production units as autonomous agents and supports state-dependent operation assignments under changing order states, resource availability, transport conditions, and production-unit fail-

ures. Accordingly, the study addresses how selected biological control mechanisms can be systematically translated into agent behaviours, feasibility rules, and event-driven interactions for production logistics. The main contributions of this paper are threefold.

- First, selected biological control mechanisms are abstracted and systematically mapped onto production logistics entities, states, and interactions.
- Second, a decentralised control architecture and an agent-based model library are developed to represent production units, orders, transport resources, and their dynamic relationships.
- Third, the framework is implemented in an event-driven simulation module that supports asynchronous decision-making, checkpoint-based action validation, explicit transport constraints, and production-unit failures. Its application to cooperative multi-agent reinforcement learning is evaluated against rule-based dispatching strategies under different failure intensities using time- and emission-related performance indicators.

The paper is structured as follows. Chapter 2 introduces the fundamentals of biological transformation and biointelligent production logistics. Chapter 3 presents the methodological transfer of biological principles, the conceptual design, the agent-based model library, and the simulation module. Chapter 4 applies and evaluates the developed control framework in a reinforcement-learning-based production logistics scenario. Finally, Chapter 5 summarizes the findings and outlines directions for further research.

2 Fundamentals of Biointelligent Production Logistics Systems

Biointelligent production logistics addresses the need for production systems that can adapt their structure and behaviour to changing orders, resource availability, and disturbances.

It combines biological knowledge, technical systems, and information technologies in order to transfer biological principles of living systems to production and logistics, such as autonomy, self-organisation, adaptability, decentralization, and resource-efficient behaviour. The biological transformation of manufacturing therefore extends the predominantly digital perspective of Industry 4.0 by treating biological principles not only as objects of observation, but also as models for the organisation and control of industrial systems [12, 13].

This perspective is particularly relevant for production logistics. Machines, transport systems, resources, and production orders are subject to continuous temporal and spatial changes. Adaptive production logistics systems there-

fore require mechanisms that enable resources to respond to changing system states and coordinate their actions according to current demand. Instead of relying exclusively on centralised planning, individual system elements should be able to perceive their environment, exchange information, evaluate local alternatives, and coordinate their actions. On this basis, temporary and task-specific networks can emerge according to current demand.

The biological transformation of industrial value creation describes the systematic transfer of biological knowledge, principles, and functions to industrial products, processes, and production systems. Its objective is to increase the adaptability, resilience, and resource efficiency of industrial value creation by combining biological, technical, and information-processing approaches [12, 13].

Miehe et al. distinguish three basic development modes of biological transformation: inspiration, integration, and interaction [12]. Inspiration refers to the abstraction of biological principles and their transfer to technical systems. Typical examples include self-organisation, adaptation, decentralization, swarm behaviour, and feedback mechanisms. Integration describes the direct incorporation of biological components or processes into technical applications, for example through microorganisms, enzymes, or bio-based materials. Interaction represents the close cooperation of biological and technical systems within a common value-creation process.

The research presented in this paper is based on the mode of inspiration. It analyses and abstracts biological mechanisms of decentralised coordination and process regulation and transfers them to the control of production logistics systems. In particular, the study focuses on self-organisation and dynamic cluster formation as mechanisms for increasing the adaptability and efficiency of production logistics. An external stimulus, such as the arrival of a customer order or the failure of a production unit, can trigger the temporary combination of suitable production units into a task-specific network. After task completion, the network is dissolved and the units become available for new tasks [15].

This cellular and decentralised understanding differs from conventional production systems with centralised control structures. Instead of defining all process sequences in advance, the system behaviour emerges from the local interaction of autonomous production units. Each unit provides specific capabilities, responds to coordination-relevant information in the shared environment, and participates in temporary production logistics clusters according to current requirements.

The conceptual model in fig.1 illustrates this principle by showing how individual cells form a structured production network from an initially unstructured basic system.

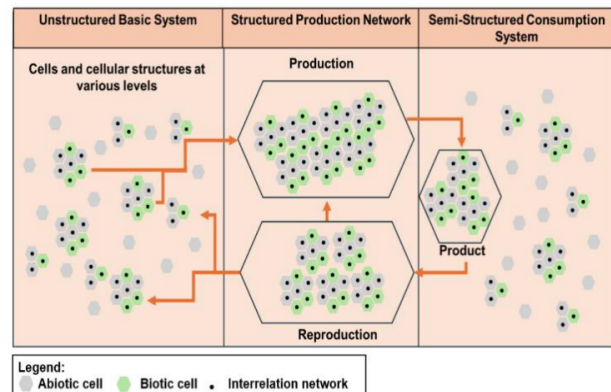


Figure 1: Conceptual Model of Biointelligent Production According to [15]

Following production, the units can return to the basic system or participate in further value-creation and reproduction processes. The figure serves as a conceptual reference rather than a direct representation of the simulation architecture developed in this paper.

Based on this understanding, the biological transformation provides the conceptual foundation for the bio-inspired control framework developed in this paper.

3 Development of a control framework for Biointelligent Production Logistics

This chapter presents the development of a bio-inspired control framework for production logistics. Building on the fundamentals of biological transformation and decentralised control structures introduced in the previous chapter, biological principles of self-organisation and temporary cluster formation are transferred to industrial production logistics systems. The development covers the conceptual definition of the control framework, the representation of production units as autonomous agents, and the implementation of model components in a software module. The following sections first describe the methodological approach and the sequence of the individual development steps.

3.1 Methodological Approach

The development of the bio-inspired control framework follows a structured approach based on bio-inspiration that links the identification of biological control principles with their implementation in an agent-based simulation environment. The methodological procedure consists of five inter-related steps.

First, a structured literature analysis is conducted to identify biological organisational and control mechanisms

with potential relevance to support decentralised coordination and self-organisation. Three principles are selected based on their relevance to decentralised decision-making, indirect coordination, and local process validation for transfer to production logistics systems. The selected principles are subsequently abstracted from their original biological context in order to derive generally applicable functional mechanisms.

Second, a systematic transfer logic is applied to map these mechanisms onto production logistics systems. For this purpose, biological entities, interactions and environmental stimuli are translated into corresponding elements of production logistics, such as production units, transport resources, orders, process states, and system disturbances.

Third, the conceptual structure of the control framework is developed. Relevant production logistics units are represented as autonomous agents. Each agent is characterised by static resource parameters, such as its capabilities, processing rates, emission rates and failure parameters, as well as dynamic local information, such as availability of resources, operating status, and the currently assigned order. Based on these elements, decentralised decision-making and temporary cluster formation can be described.

Fourth, the resulting agent types and behavioural mechanisms are formalised in an agent-based model library. The library provides reusable model components for the representation of different production units and enables the configuration of self-organising production and logistics networks.

Finally, the model library is integrated into a software module that supports the simulation of bio-inspired control structures. The software module provides the technical basis for implementing different system configurations and for conducting subsequent simulation-based investigations of the developed control principles.

The application and evaluation of the resulting control framework are addressed in Chapter 4.

3.2 Transfer of Biological Principles to Production and Logistics Systems

The transfer of biological principles to production logistics systems requires the abstraction of biological mechanisms from their original context and their reformulation as technical control principles. In this study, the transfer focuses on three mechanisms that are particularly suitable: self-organisation, stigmergy, and checkpoint-based action validation. These mechanisms address complementary aspects of a bio-intelligent control framework, namely the autonomous formation of virtual operational structures, indirect coordination between distributed production units, and the local prevention of invalid process execution.

Self-organisation describes the autonomous formation of complex structures from comparatively simple units through local interactions [16]. In biological systems, global patterns and coordinated behaviour can emerge without a central decision-making authority. Transferred to production logistics, this principle can be represented by a multi-agent system in which production-logistics entities act autonomously according to locally available information, defined capabilities, and individual decision rules. The agents cooperate dynamically to fulfil a manufacturing task, while the resulting production structure emerges from their interactions rather than from a predefined centrally controlled schedule. Self-organisation therefore provides the fundamental organisational principle of the proposed control framework.

Stigmergy complements self-organisation by providing a mechanism for decentralised communication and coordination. In biological collectives, stigmergic communication is based on environment-mediated signals [17]. These signals influence the behaviour of other individuals without requiring direct communication between them. In the present system, stigmergy is interpreted as environment-mediated coordination. The shared factory state stores the consequences of previous actions, such as machine occupation, order progress, transport queues, and resource availability. These state changes influence the subsequent decisions of other production units. Agents observe these variables and adapt their decisions autonomously. For example, a manufacturing agent may expose its current availability through the shared environment, while other agents evaluate this information together with process requirements, delivery-related and relevant system-state information. Coordination is therefore achieved indirectly through a shared system environment. This reduces the need for centralised information processing and enables agents to react locally to changes in system conditions.

The third transferred principle is derived from DNA replication checkpoints. Biological cells use checkpoint mechanisms to monitor critical stages of DNA replication and prevent process continuation when essential conditions are not fulfilled [18, 19]. The key transferable idea is the integration of local validation points into complex manufacturing processes. In the proposed control framework, checkpoints are implemented through action masks that restrict the set of orders available to a production unit. An order is included in the feasible action set only if the order has arrived, has not been completed, is not currently being processed and the next operation is included in the capability set of the active production unit. In this way, infeasible processing decisions are excluded locally before an action is executed, without requiring continuous centralised supervision.

Within the control framework, the three principles are combined into a coherent control logic. Self-organisation

defines how autonomous agents form temporary production logistics structures. Stigmergy specifies how the agents obtain and exchange coordination-relevant information through the shared environment. Checkpoints constrain the agents' decision-making by ensuring that only valid processes can be selected and then executed. The combination of these principles enables decentralised coordination while maintaining process feasibility and operational control. It thus provides the conceptual basis for the subsequent design of the agent types, their capabilities, and their interaction rules.

3.3 Conceptual Design of the control framework

The conceptual design of the control framework builds on the methodological transfer of self-organisation, stigmergy, and checkpoint-based process validation to production logistics. The resulting system is conceived as a decentralised multi-agent architecture in which production logistics resources are represented as autonomous agents. In the present implementation, the production units constitute the primary agent type. Each production unit is characterised by a technical capability profile, resource-specific performance parameters, and a dynamic operational state. These attributes define the individual decision space of the agent and form the basis for differentiated behaviour within the production system.

The control framework does not rely on a central scheduler that assigns all operations in a predefined sequence. Instead, production decisions emerge from the interaction between autonomous production-unit agents and a shared event-driven factory environment. This environment stores the information including the current states of orders, production units, transport resources, process progress, failures, repairs, and emissions. It therefore acts as the medium through which agents coordinate indirectly. In accordance with the principle of stigmergy, agents do not need to communicate through direct message exchange. By observing state changes in the environment caused by previous actions and events, agents adapt their decisions accordingly.

A central requirement of the proposed control concept is an asynchronous decision logic. In real production logistics systems, resources do not become available in synchronised decision rounds. Production units finish operations at different times, may wait for transport, or may temporarily become unavailable due to failures. Therefore, the control framework is designed so that a production unit only participates in decision-making when it is actually able to execute a feasible next operation. This supports a closer representation of real operational dynamics.

At the implementation level, dynamic cluster formation is operationalized through temporary resource-order assignments. Across successive process steps, these assignments form a changing production logistics network. A production

unit does not belong permanently to a fixed process chain. Instead, it forms a temporary relation with an order and its currently required process step. The assignment terminates when the operation is completed or interrupted by a failure. In the latter case, the unfinished operation becomes eligible for reassignment after the corresponding state updates have been processed. This corresponds to the biological principle of self-organisation, since the production structure is not predefined centrally but arises dynamically from local capabilities, resource availability, and order requirements.

To maintain process feasibility, checkpoint-based validation restricts each production unit to currently feasible order assignments. If no order satisfies the checkpoint conditions, no decision is triggered and the production unit remains idle until a subsequent event changes the system state and creates a feasible assignment.

Transport is considered an integral part of the production logistics control problem. If the next operation of an order has to be carried out by a different production unit, a transport process becomes necessary. Transport availability affects operation start time, queueing delays, completion times, and transport-related emissions, and must therefore be considered when dispatching decisions. A production unit's decision is not only evaluated by its local processing ability, but also by its effect on material flow, waiting time, completion time, and CO₂ emissions. This integration enables the control framework to represent production and logistics as one coupled decision problem.

The architecture supports cooperative multi-agent reinforcement learning. Approaches based on reinforcement learning have already demonstrated their potential for solving complex, state-dependent logistics decision problems [9, 10]. In the implemented system, each production unit selects an order from its locally feasible action set, while the combined effects of local decisions are assessed through system-level performance indicators.

3.4 Development of the Agent-Based Model Library

The agent-based model library was developed to formalise the relevant production logistics entities as reusable simulation objects. It distinguishes between autonomous decision-making agents and environment-managed simulation objects. In the current implementation, production units act as decision-making agents, whereas orders and transport resources are represented as environment-managed simulation objects. Each production unit represents a cellular manufacturing resource that can participate in decentralised decision-making and can be trained as a reinforcement learning agent. The model library therefore provides the structural basis for representing production resources, their capabilities, their operational states, and their interaction with orders and transport processes.

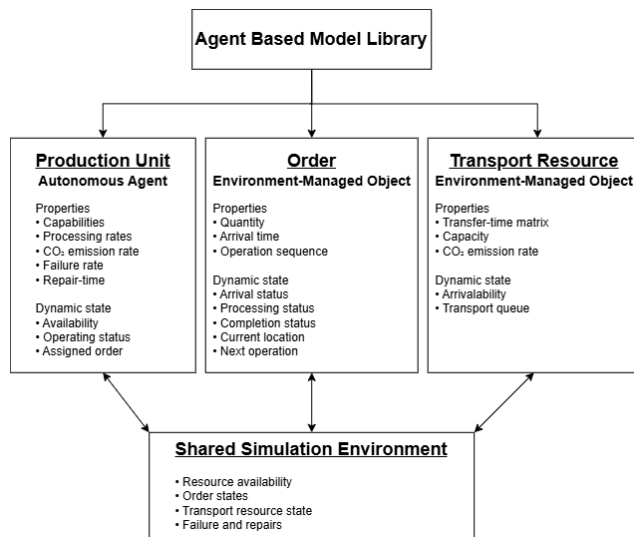


Figure 2: Overview of the Agent-Based Model Library

Figure 2 provides an overview of the agent-based model library and the relationships between its main components. Production units constitute the autonomous decision-making agents, while orders and transport resources are represented as environment-managed simulation objects. Their properties and parameters define the feasible interactions within the production logistics system, while dynamic state information is exchanged through the shared simulation environment.

The central agent type in the current model library is the production unit. A production unit is specified by a unique identifier, a set of process capabilities, operation-specific processing rates, a CO₂ emission rate, a failure rate, and a repair-time interval. These parameters describe both the functional and ecological properties of the production resource. During simulation, a production unit may be available, occupied by processing or transport-related waiting, or unavailable due to failure and repair. This state model enables the simulation to represent not only productive operation, but also temporary unavailability, logistical dependencies, and failure-related interruptions.

In addition to production units, the library includes a transport module that represents the logistics resource responsible for material movement between production units. The transport module is characterised by a transfer-time matrix representing travel times between different production units, transport capacity, and a transport-related CO₂ emission rate. Although the current implementation treats the transport module primarily as a shared logistics resource rather than as an independently learning agent, it is modelled explicitly because transport availability directly affects production decisions, material flow, waiting times, and

emissions. This allows the system to represent production and logistics as an integrated control problem.

Orders are modelled as process carriers that move through the production logistics system. Each order is defined by static attributes, including quantity, arrival time, and a required operation sequence. Its dynamic state records whether it has arrived, whether it is currently in progress, whether it has been completed, its current material location, and its next required operation. The remaining route can be derived from the operation sequence and the current process position. The model library thus enables production units to evaluate whether they are capable of processing a specific order. Orders therefore function as dynamic coordination objects within the shared environment.

The library also specifies the capability logic and provides state attributes required to construct the checkpoint-based action mask for decentralised decision-making, as described in Chapter 3.2. It ensures that invalid actions are filtered locally before execution and that the model representation and the decision interface use the same feasibility conditions.

Resource parameters are included not only to describe technical performance, but also to support sustainability-oriented control. Processing rates determine operation durations, while CO₂ emission rates allow the simulation to calculate the environmental impact of both production and transport activities. Failure rates and repair-time ranges introduce robustness-related behaviour into the model. As a result, the model library can represent heterogeneous production units whose decisions differ according to speed, emissions, reliability, and current availability.

The implementation of the model library follows a modular structure. Domain objects such as production units, orders and transport units are defined separately from the simulation logic. This separation supports the reuse of the domain model in different control interfaces, including rule-based dispatching and reinforcement-learning-based dispatching. The library therefore forms an intermediate modelling layer between the conceptual control design and the executable software module.

Overall, the developed agent-based model library provides a functional representation of the relevant production logistics units. It formalised the production units, orders, transport resources, capabilities, state variables, and sustainability-related parameters required by the control framework. This establishes a reusable and extensible modelling layer between the conceptual architecture and the executable simulation module.

3.5 Development of the Simulation Module

The simulation module implements the bio-inspired control logic in an executable event-driven simulation environment for self-organising production logistics networks. It operationalises the bio-inspired control principles derived in the previous chapter by combining an event-driven factory simulator with a multi-agent decision interface. The module therefore provides the technical basis for investigating how autonomous production units can form temporary operational structures, react to disturbances, and reorganise production flows under changing system conditions.

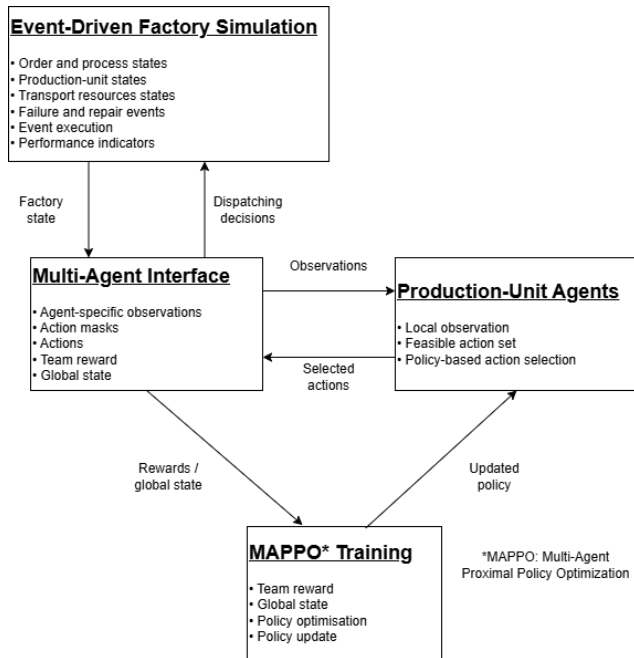


Figure 3: Architecture of the Simulation and Multi-Agent Reinforcement-Learning Interface

Figure 3 illustrates the interaction between the event-driven simulation, the production-unit agents, and the reinforcement-learning interface. The simulation maintains and updates the factory state, while decision-ready production units receive agent-specific observations through the multi-agent interface. Selected actions are returned to the simulation for execution, and the resulting state transitions and rewards provide the basis for subsequent decisions and reinforcement learning.

At the core of the software module is a discrete-event simulation engine. This engine maintains the state of the production system, including production units, orders, transport resources, processing progress, failures, repairs, transport queues, and environmental performance indicators. The simulation time is advanced by events such as order arrivals, production completions, transport completions, machine failures, and repair completions. Events up-

date the factory state and may activate new production-unit decisions. This event-based structure is essential for representing self-organising production logistics networks, because decisions are triggered by changes in system conditions rather than by fixed time intervals.

t= 0.00	ORDER_ARRIVAL	Order-9 arrived quantity=21 operations=2 route=inspection -> assembly
t= 0.00	ORDER_ARRIVAL	Order-10 arrived quantity=23 operations=2 route=inspection -> mix
t= 0.00	DISPATCH	PU-1 selected Order-8 operation=1 "assembly" processing_time=4.83 estimated_CO2=16.92
t= 0.00	DISPATCH	PU-2 selected Order-5 operation=1 "assembly" processing_time=7.25 estimated_CO2=18.88
t= 0.00	DISPATCH	PU-3 selected Order-9 operation=1 "inspection" processing_time=5.25 estimated_CO2=15.12
t= 4.83	PROCESS_END	PU-1 completed Order-8 operation=1 "assembly" remaining=drilling -> cutting
t= 4.83	DISPATCH	PU-1 selected Order-2 operation=1 "cutting" processing_time=5.68 estimated_CO2=19.60
t= 5.25	PROCESS_END	PU-3 completed Order-9 operation=1 "inspection" remaining=assembly
t= 5.25	DISPATCH	PU-3 selected Order-7 operation=1 "inspection" processing_time=3.75 estimated_CO2=9.38
t= 7.25	PROCESS_END	PU-2 completed Order-5 operation=1 "assembly" remaining=drilling
t= 7.25	DISPATCH	PU-2 selected Order-9 operation=2 "assembly" processing_time=5.25 estimated_CO2=7.88
t= 7.25	TRANSPORT_START	TR-1 transferring Order-9 quantity=21 PU-3 -> PU-2 transfer_time=1.50 TR_CO2=1.50
t= 8.75	TRANSPORT_END	TR-1 delivered Order-9 PU-3 -> PU-2 PU-2 started operation "assembly"
t= 9.00	PROCESS_END	PU-3 completed Order-7 operation=1 "inspection" remaining=assembly
t= 9.00	DISPATCH	PU-3 selected Order-3 operation=1 "inspection" processing_time=2.50 estimated_CO2=6.25
t= 11.50	PROCESS_END	PU-3 completed Order-3 operation=1 "inspection" remaining=drilling -> mix
t= 11.50	DISPATCH	PU-3 selected Order-10 operation=1 "inspection" processing_time=5.75 estimated_CO2=14.38
t= 12.95	PU_FAILURE	PU-3 failed on Order-10 operation="inspection" processed_time=1.45 progress=rolled_back order=returned_to_pool material_eta=PU-3 repair_time=12.00
t= 13.43	TRANSPORT_END	TR-1 delivered Order-7 PU-3 -> PU-1 PU-1 started operation "assembly"
t= 14.00	PROCESS_END	PU-2 completed Order-9 operation=2 "assembly" remaining=none
t= 14.00	ORDER_COMPLETE	Order-9 fully completed
t= 14.00	DISPATCH	PU-2 selected Order-3 operation=2 "assembly" processing_time=2.50 estimated_CO2=3.75
t= 14.00	TRANSPORT_START	TR-1 transferring Order-3 quantity=23 PU-3 -> PU-2 transfer_time=1.50 TR_CO2=1.50
t= 17.93	PROCESS_END	PU-1 completed Order-8 operation=2 "cutting" remaining=drilling
t= 17.93	DISPATCH	PU-1 selected Order-4 operation=1 "mix" processing_time=6.50 estimated_CO2=22.75
t= 18.00	PROCESS_END	PU-2 completed Order-3 operation=2 "assembly" remaining=none
t= 18.00	ORDER_COMPLETE	Order-3 fully completed
t= 18.00	DISPATCH	PU-2 selected Order-10 operation=1 "inspection" processing_time=7.67 estimated_CO2=11.50
t= 18.00	TRANSPORT_START	TR-1 transferring Order-10 quantity=23 PU-3 -> PU-2 transfer_time=1.50 TR_CO2=1.50
t= 19.50	TRANSPORT_END	TR-1 delivered Order-10 PU-3 -> PU-2 PU-2 started operation "inspection"

Figure 4: Excerpt from the simulation event log illustrating the implementation of dispatching, processing, transport execution, and failure-induced reconfiguration.

Figure 4 illustrates an excerpt from the event log generated during simulation and provides an example of how the event-driven control mechanisms are implemented. Each log entry represents a discrete simulation event, while t denotes the corresponding simulation time. The simulation clock therefore advances according to events rather than fixed time steps. Events include dispatching decisions, processing and transport completion, order completion, and production-unit failures.

A DISPATCH event records the local decision of a decision-ready production-unit agent. For example, PU-2 (Production Unit) selected Order-3 indicates the temporary assignment of the order and its next required operation to PU-2. The operation can only be selected if it is validated by the checkpoint mechanism based on the current system state and the production unit's capabilities. The reported processing_time, such as processing_time=2.50 in Fig. 4, represents the required duration for execution of the selected operation and is calculated as

$$t_{proc} = \frac{q_{order}}{r_{PU,op}}, \quad (1)$$

where q_{order} denotes the order quantity and $r_{PU,op}$ the operation-specific processing rate of the production unit. The corresponding estimated_CO2 represents the expected production-related emissions associated with the assignment and is calculated from the processing duration and the production-unit-specific emission rate. If the production

unit differs from the current material location, a transport request is generated before processing can start, as illustrated by the `TRANSPORT_START` and `TRANSPORT_END` events. Transport-related emissions `TR_CO2` depend on the transfer duration and the emission rate of the transport resource.

The failure event in Fig. 4 illustrates the implemented re-configuration mechanism. At $t = 12.95$, PU-3 fails while processing the inspection operation of Order-10 after 1.45 minutes of elapsed processing time. The elapsed processing time and associated emissions are recorded, while the unfinished operation is rolled back and the order is returned to the dispatchable order pool. Its material location remains at PU-3, and PU-3 becomes unavailable for the specified repair period. The updated order and material state subsequently affect the feasible action sets of the remaining production-unit agents, allowing other capable production units to take over the interrupted operation and triggering transport if relocation is required. The event sequence therefore provides an implementation-level example of self-organisation through temporary order-resource assignments, environment-mediated coordination, checkpoint-based validation, and failure-induced reconfiguration within the shared simulation environment.

The cluster formation logic is implemented through the temporary assignment of orders to capable production units. When a production unit becomes available, it autonomously evaluates which orders are feasible under the current system state. In the simulation, temporary cluster formation is operationalized through the assignment of an order and its next operation to a capable production unit, together with a transport request where relocation is required. These clusters are not permanent structures. They are created dynamically during dispatching and dissolve after completion or failure. In this way, the software module translates the conceptual principle of self-organisation into a concrete decision and execution mechanism.

The decision logic follows a multi-agent architecture, where each production unit is represented as an independent agent within an asynchronous decision interface. An agent is triggered for decision-making only when it is idle and at least one operation is available according to the checkpoint-based action mask. This asynchronous mechanism avoids artificial synchronised decision rounds and eliminates the need for unavailable production units to perform no-operation actions. If several production units become decision-ready at the same simulation time, random arbitration determines the sequence in which their decisions are processed. It does not replace the local dispatching decision itself, but avoids a permanent priority caused by a fixed agent order. Reproducibility is ensured by using controlled random seeds. As a result, decentralised decision-making is achieved while maintaining a controlled and analytically tractable simulation process. In the reinforcement-learning-

based application, the available production unit selects one order from its currently feasible action set.

Failure recovery and reconfiguration are implemented through interruption handling, repair events, and reassignment of unfinished operations, as illustrated in Fig. 4. If a production unit fails during processing, the elapsed processing time and associated emissions are recorded, while the current operation remains incomplete and must be restarted by a capable production unit. The corresponding order states are updated, and the unfinished operation becomes eligible for reassignment once the resulting state changes have been processed. At the same time, the failed production unit is removed from the available resource set and a repair event is scheduled. Other capable production units may then take over the interrupted operation if permitted by the current system state. After repair, the failed production unit becomes available again and can participate in subsequent dispatching decisions. This mechanism enables automatic re-routing after resource failure and supports dynamic reorganisation of the production logistics network.

Transport processes are integrated as explicit logistics constraints. If an order has to move between production units, transport requests are processed by a shared transport resource according to a first-in-first-out discipline. This represents a simplifying dispatching assumption and can be replaced by alternative transport-control rules in future extensions. The transport module therefore influences the timing and thus the quality of production decisions. Dispatching decisions can create transport demand, transport waiting times, and additional CO₂ emissions. By modelling transport as a shared constrained resource, the simulation module represents production and logistics as a coupled network rather than as separate sequential processes.

External stimuli are incorporated through configurable and event-based changes in the simulation environment. Order arrivals introduce new production demand, machine failures reduce available capacity, repair completions restore resources, and transport completions update the material location of orders. These stimuli modify the shared environment observed by the production unit agents and can trigger new decision opportunities. The production logistics network therefore reorganises dynamically in response to newly arriving orders, resource states, and logistics constraints.

The simulation module provides the interfaces required for cooperative multi-agent reinforcement learning. Each decision-ready production unit is represented as an individual agent. The multi-agent interface provides observations and action masks, receives the selected order from the production unit as the agent action, and transfers the resulting decision to the event-driven simulation for execution. It additionally exposes rewards and global state information required for MAPPO (Multi-Agent Proximal Policy

Optimization)-based learning. The specific learning framework and training configuration are described in Chapter 4.

The reward mechanism evaluates the effects of decisions on system-level performance. The implemented interface supports a configurable team reward combining time- and emission-related penalties. The specific reward configuration used for the evaluation is described in Chapter 4.1. In addition, the module records performance indicators such as makespan, completed orders, production-related emissions, transport-related emissions, transport waiting times, and failure counts. These indicators provide the basis for evaluating the proposed bio-inspired control framework of production logistics networks under different scenarios.

Overall, the developed simulation module implements the bio-inspired control framework as a simulation environment for decentralised, adaptive production logistics. It realizes cluster formation through temporary resource-order assignments, checkpoint-based decision validation, failure recovery through failure handling and automatic re-routing, and event-driven responses to changing system conditions. It also provides the technical interfaces required for the reinforcement-learning-based evaluation presented in Chapter 4.

The implementation was verified with respect to the main process and control mechanisms of the simulation model. The checks included order-state progression, capability-based action masking, event-driven processing and transport execution, as well as failure, repair, and re-assignment behaviour. Functional behaviour of the integrated framework is subsequently examined in the application scenario presented in Chapter 4, where decentralised decision-making, event-driven reconfiguration, and system-level performance are evaluated under different operating conditions.

4 Application and Evaluation of the Developed control framework

This chapter applies the developed control framework in a simulation-based production logistics scenario and evaluates reinforcement-learning-based dispatching within the implemented bio-inspired control architecture. Since the architecture, model components, and event-driven decision logic have already been described in Chapter 3, the following analysis focuses on the experimental configuration and resulting system performance.

The evaluation examines the performance of the developed bio-inspired control framework when combined with reinforcement-learning-based dispatching and compares the learned policy with rule-based dispatching strategies under different production-unit failure intensities. Team reward, makespan, total CO₂ emissions are used as primary per-

formance indicators. The following section describe the experimental configuration, present the evaluation results, and discuss the observed system behaviour and limitations.

4.1 Application Scenario and Experimental Configuration

The application scenario represents a small-scale flexible production logistics system with heterogeneous production units and explicit transport constraints. It is used as a controlled simulation environment for examining whether reinforcement learning can improve the system performance within the developed bio-inspired control framework.

The production system consists of three production units and one shared transport unit. The production units differ with respect to their operation capabilities, processing rates, CO₂ emission rates, and failure parameters. The considered operation types are cutting, drilling, assembly, inspection, and mixing. Each production unit can process only a subset of these operations, so that an order may have several feasible alternatives for some process steps, while other steps can only be executed by specific units. This creates a non-trivial dispatching problem in which local decisions influence later routing options, transport demand, and resource occupation.

The order set consists of ten production orders. Each order is generated with a quantity between 10 and 30 units to be processed and requires a sequence of two to four operations. All orders arrive at the beginning of the simulation episode. The order routes are therefore known after arrival, but the actual execution sequence emerges from the decisions of the production-unit agents and from the event-driven state changes of the factory environment.

Transport unit is represented by a single shared transport resource. If the material from an order has to move from one production unit to another, a transport request is generated and processed according to a first-in-first-out rule. The transport time depends on the distance between source and target production unit. Transport therefore affects both the temporal performance of the system and the resulting CO₂ emissions. This ensures that dispatching decisions are evaluated not only by local processing time, but also by their consequences for material flow and transport-related waiting times and emissions.

To evaluate the adaptability of the control framework, three failure-intensity settings are considered. Machine failures are modelled based on machine operating time. For each production unit, the time to failure follows an exponential distribution with rate parameter λ , while the repair time is uniformly distributed between 5 and 15 minutes across all failure settings. In the no-failure setting, production-unit failures are disabled. In the high-failure setting, the failure rates of production-unit-1, production-unit-2, and production-unit-3 are set to 0.01, 0.015, and 0.01 min⁻¹, respectively. The low-failure setting uses 25% of these rates.

These settings allow the learned policy to be evaluated under different levels of disruption while keeping the underlying production-logistics structure unchanged.

The training objective combines time-related and environmental performance in a shared team reward. Makespan, measured in minutes, and total CO₂ emissions, measured in kilograms, are normalized and weighted equally, with both quantities negatively related to the resulting reward. Consequently, shorter completion times and lower emissions lead to a higher reward. Since both objectives are represented as penalties, the resulting reward values are negative, whereby values closer to zero indicate better overall performance. In addition to the aggregated team reward, makespan and CO₂ emissions are evaluated separately to assess the individual objective components.

The learned policy is compared with several reference policies as baselines. The initial and final policies are used to assess the effect of training, while the best policy is selected according to mean team reward for the subsequent independent evaluation. In addition, conventional rule-based dispatching strategies are included as reference policies, namely random dispatching, shortest processing time (SPT), an immediate CO₂-oriented heuristic, and a most-work-remaining rule (MWKR). At each decision point, the environment has already determined the production unit which is ready to process an order. The rule-based policies therefore do not perform global resource allocation across all production units, instead they select one order from the set of orders that can be legally assigned to the currently active production unit. Random dispatching selects an order at random from the feasible set, providing an uninformed reference. SPT prioritizes the shortest processing time of the current assignment, CO₂ minimizes the immediate processing and transport emissions of the current assignment, and MWKR prioritizes orders with the largest estimated remaining workload. All baselines operate under the same event-driven environment and legal-action constraints as the learned policy and use only information available at the current decision time. Together, these strategies provide the benchmark for assessing whether the learned policy achieves improved system performance beyond simple local dispatching rules.

4.2 Evaluation Results of the Learned Control Policy

Based on the application scenario described above, the evaluation examines whether the control framework developed in this work can be transformed into an effective control policy through reinforcement learning. In particular, the analysis considers whether the production-unit agents learn to select suitable order-operation assignments from their locally feasible action sets and whether the resulting system-level behaviour remains effective when production-unit failures occur and thus change the system state. Since process feasibility, asynchronous decision-making, transport execu-

tion, and failure recovery are already enforced by the simulation environment, the evaluation focuses on the quality of the decisions made within this control structure.

Training leads to a consistent improvement of the policy in all three investigated failure settings. Compared with the initial policies, the best learned policies achieve higher mean team rewards across the three training seeds. The best learned policies are obtained at different training updates across the three training seeds, indicating that the final policy does not necessarily provide the highest validation performance.

Figure 5 illustrates the aggregated behaviour for the high-failure setting, which represents the highest failure intensity considered in this study. The mean reward improves substantially during the first part of training and subsequently approaches a relatively stable level. The learned policy exceeds the fixed most-work-remaining rule (MWKR) reference during training and remains predominantly above it in the later updates. At the same time, the variation across the three training seeds decreases compared with the early training phase.

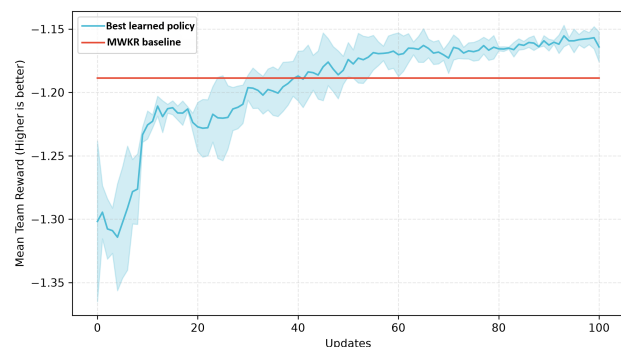


Figure 5: Aggregated validation reward of the learned policy under the high-failure setting across three training seeds. The shaded area represents the mean ± 1.96 standard errors across training seeds, and the horizontal line indicates the team reward achieved by the fixed MWKR baseline.

The policy validation procedure is therefore used to select the best policy for the subsequent independent evaluation. The independent evaluation confirms that the improvement observed during training translates into better operational performance of the production logistics system. Figure 6 compares the best learned policy with the four rule-based dispatching strategies in terms of mean team reward, makespan, and total CO₂ emissions under the three investigated failure intensities. For the learned policy, the error bars represent the variation across the three independently training seeds. The rule-based strategies do not constitute independent training runs and are therefore reported separately without interpreting their repeated evaluations as

training-seed variability. Among the rule-based strategies, MWKR consistently provides the strongest reference performance, whereas random dispatching, shortest processing time (SPT), and the immediate CO₂-oriented heuristic result in lower mean rewards. The following comparison therefore concentrates on MWKR as the most relevant rule-based baseline.

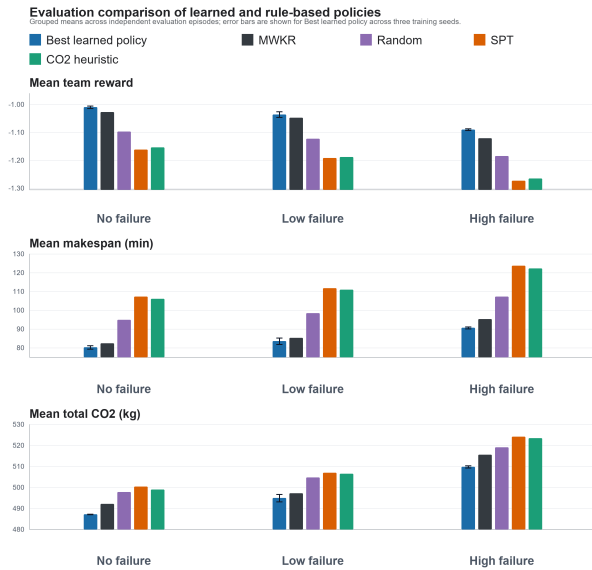


Figure 6: Comparison of mean team reward, makespan, and total CO₂ emissions for the best learned policy and the rule-based dispatching policies under the three production-unit failure settings.

To examine whether the reward improvement results from trade-off between time and emissions, Fig. 7 compares the policies directly in the makespan–CO₂ performance space. Each point represents the mean performance of one policy under one failure setting.

The position of the policies in Fig. 7 confirms that the learned policy is not merely exploiting the weighting between the two reward components. Within each of the investigated failure settings, the best learned policy achieves both a lower mean makespan and lower mean total CO₂ emissions than the rule-based baselines. In the considered mean-performance space, the learned policy therefore occupies the preferable region of the evaluated policy set. This result is particularly relevant because production and transport decisions are coupled in the developed simulation environment: an assignment can simultaneously affect processing duration, subsequent transport demand, resource occupation, and the corresponding emissions.

Although checkpoint-based action validation restricts all policies to currently executable order assignments, the choice among these feasible alternatives has a substantial ef-

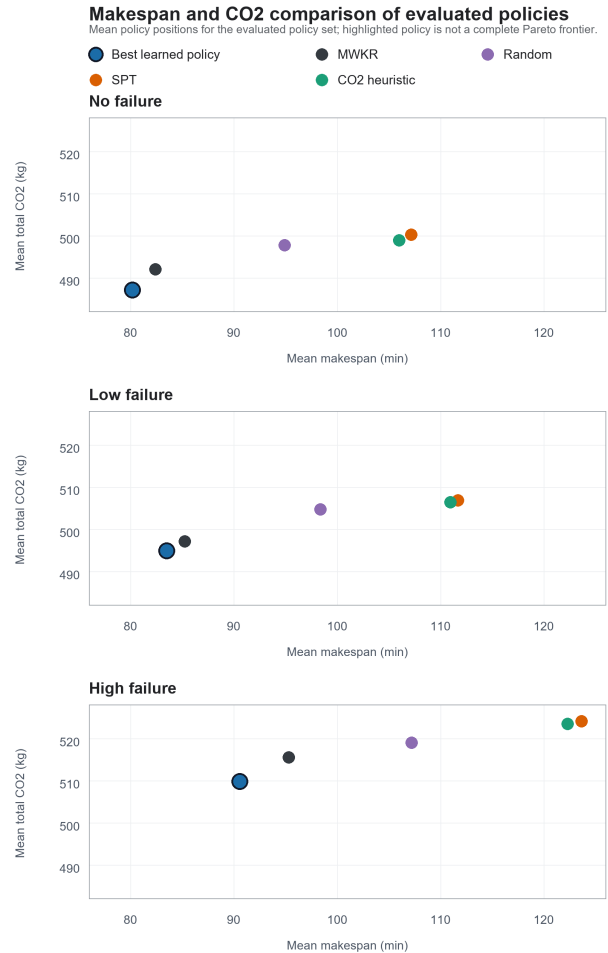


Figure 7: Mean makespan and total CO₂ emissions of the evaluated policies under the three production-unit failure settings. The best learned policy is highlighted.

fect on the resulting production-logistics performance. This distinction is important for the proposed control concept. The described checkpoint mechanism ensures that agents do not execute invalid process steps, but it does not prescribe which feasible order should be selected. The observed performance improvement therefore results from the learned prioritisation among valid temporary order-resource assignments rather than from feasibility enforcement itself. In this sense, the reinforcement-learning policy complements the bio-inspired control framework: it defines how local decisions can be made and reconfigured, while the learned policy determines which of the currently feasible alternatives should be preferred.

Overall, the learned policy achieves better system-level performance than the investigated rule-based baselines across all three failure settings.

4.3 Discussion of System Behaviour and Limitations

The observed behaviour is consistent with the transferred biological principles described in Chapter 3. Temporary order-resource assignments provide the basis for self-organisation, while environment-mediated state information supports indirect coordination and checkpoint-based validation maintains process feasibility. Reinforcement learning complements these structural mechanisms by determining which feasible assignment should be preferred.

The results under increasing failure intensity further indicate that the decentralised architecture can retain its functionality when the availability of production resources changes. In the simulation module, reconfiguration is realised by interrupting failed operations and making the unfinished operation available for reassignment to other capable production units after the corresponding states update. The persistence of the learned policy's performance advantage under the investigated failure settings suggests that state-dependent dispatching can make use of this reconfiguration mechanism. This is consistent with the underlying bio-inspired concept, in which temporary task-specific structures are formed according to the current demand and system state instead of being fixed in advance.

At the same time, the evaluation is subject to several limitations. First, the investigated application scenario is deliberately small and controlled, consisting of three production units, one shared transport resource, and ten orders. The obtained performance differences therefore cannot be directly generalised to larger production systems with more resources, longer process chains, multiple transport resources, storage systems, or more heterogeneous order structures. In addition, the transport resource is represented as an environment-managed shared resource instead of an independently deciding agent. These assumptions support a controlled evaluation of the production-unit policies, but they restrict the range of dynamic interactions represented in the current model. Finally, the evaluation is simulation-based and does not yet demonstrate transfer to a physical production-logistics system. The developed framework establishes an executable representation of the bio-inspired control principles and enables their quantitative investigation, but practical implementation would additionally require interfaces to machines and transport systems, reliable acquisition of resource states, and validation under real operational disturbances.

Despite these limitations, the results provide an initial indication that bio-inspired control mechanisms can be combined with reinforcement-learning-based dispatching within a coherent production-logistics control architecture. The framework separates process feasibility and event execution from policy optimisation and therefore provides a suitable basis for further investigation. The current evaluation uses rule-based dispatching strategies as reference poli-

cies and does not include a direct comparison with a fully centralised optimisation-based production control system.

5 Summary and Outlook

This paper developed and evaluated a bio-inspired control framework for decentralised and self-organising production logistics. Selected biological principles were transferred to production-logistics control and implemented in an agent-based, event-driven simulation environment. The resulting framework integrates decentralised decision-making, local feasibility validation, transport processes, and failure-induced reconfiguration, and provides an interface for reinforcement-learning-based dispatching.

The simulation-based evaluation shows that the framework supports reinforcement-learning-based decentralised dispatching under different production-unit failure intensities. In the investigated scenario, the learned policy outperforms the considered rule-based dispatching strategies in terms of team reward and achieves lower mean makespan and total CO₂ emissions than the strongest rule-based baseline.

These findings indicate that bio-inspired structural control mechanisms can be combined with reinforcement-learning-based dispatching to support adaptive production logistics. The developed framework therefore provides an executable basis for further investigating self-organising production-logistics control.

Future work should extend the evaluation to larger and more heterogeneous production-logistics systems with dynamic order arrivals, additional production and transport resources, and more complex disturbance scenarios. Alternative transport-control strategies and different reward formulations should also be investigated to analyse their influence on the resulting system behaviour. Furthermore, the robustness and generalisation of the learned policies should be evaluated across varying system configurations. Ultimately, the developed approach should be transferred from the current simulation environment to real-world production-logistics systems to assess its applicability under industrial operating conditions.

6 Funding Acknowledgement

This publication was prepared as part of the project "Auf dem Weg zur biointelligenten Produktion: Entwicklung eines Steuerungssystems für biointelligente Logistik" (BUP63 - LOGbiotik). The authors would like to thank the Ministry of Science, Research and Arts of the Federal State of Baden-Württemberg for the financial support of the projects within the InnovationCampus Future Mobility.

REFERENCES

- [1] S. Yang and Y. Fu, "Interconnectedness among supply chain disruptions, energy crisis, and oil market volatility on economic resilience," *Energy Economics*, vol. 143, p. 108290, Mar. 2025.
- [2] D. Ivanov and A. Dolgui, "The shortage economy and its implications for supply chain and operations management," *International Journal of Production Research*, vol. 60, no. 24, pp. 7141–7154, Dec. 2022.
- [3] M. M. Queiroz, S. Fosso Wamba, R. D. Raut, and I. O. Pappas, "Does Resilience Matter for Supply Chain Performance in Disruptive Crises with Scarce Resources?" *British Journal of Management*, vol. 35, no. 2, pp. 974–991, 2024.
- [4] N. M. P. Bocken, S. W. Short, P. Rana, and S. Evans, "A literature and practice review to develop sustainable business model archetypes," *Journal of Cleaner Production*, vol. 65, pp. 42–56, Feb. 2014.
- [5] D. Gyulai, A. Pfeiffer, and L. Monostori, "Robust production planning and control for multi-stage systems with flexible final assembly lines," *International Journal of Production Research*, vol. 55, no. 13, pp. 3657–3673, Jul. 2017.
- [6] H. Aytug, M. A. Lawley, K. McKay, S. Mohan, and R. Uzsoy, "Executing production schedules in the face of uncertainties: A review and some future directions," *European Journal of Operational Research*, vol. 161, no. 1, pp. 86–110, Feb. 2005.
- [7] P. Leitão and F. Restivo, "ADACOR: A holonic architecture for agile and adaptive manufacturing control," *Computers in Industry*, vol. 57, no. 2, pp. 121–130, Feb. 2006.
- [8] A. Luft, N. Luft, and K. Arntz, "A Basic Description Logic for Service-Oriented Architecture in Factory Planning and Operational Control in the Age of Industry 4.0," *Applied Sciences*, vol. 13, no. 13, p. 7610, Jan. 2023.
- [9] F. Li, J. Zhou, Y. Tian, R. Noortwyck, A. Ounissi, B. Hong, K. Wen, Q. Kang, and R. Schulz, "Deep reinforcement learning with instance-invariant baseline regularization for joint retrieval and relocation scheduling in multi-deep warehouses," *Expert Systems with Applications*, vol. 328, p. 132893, Oct. 2026.
- [10] F. Li, Y. Tian, R. Noortwyck, J. Zhou, L. Kuang, and R. Schulz, "Topology-aware and highly generalizable deep reinforcement learning for efficient retrieval in multi-deep storage systems," *Journal of Intelligent Manufacturing*, vol. 37, no. 6, pp. 2503–2536, Jun. 2026.
- [11] J. Leng, X. Zhu, Z. Huang, K. Xu, Z. Liu, Q. Liu, and X. Chen, "ManuChain II: Blockchain Smart Contract System as the Digital Twin of Decentralized Autonomous Manufacturing Toward Resilience in Industry 5.0," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 8, pp. 4715–4728, Aug. 2023.
- [12] R. Mieke, T. Bauernhansl, O. Schwarz, A. Traube, A. Lorenzoni, L. Waltersmann, J. Full, J. Horbelt, and A. Sauer, "The biological transformation of the manufacturing industry – envisioning biointelligent value adding," *Procedia CIRP*, vol. 72, pp. 739–743, 2018.
- [13] S. Harst, M. Früchtel, and R. Neugebauer, "Biological transformation of manufacturing – from a vision to industrial transfer – interim evaluation from the perspective of applied research," *Procedia CIRP*, vol. 107, pp. 925–930, Jan. 2022.
- [14] R. Mieke, Y. Baumgarten, and T. Bauernhansl, "Towards a Common Understanding of the Biointelligence Concept," *Procedia CIRP*, vol. 120, pp. 1416–1421, 2023.
- [15] R. Mieke, L. Buckreus, S. Kiemel, A. Sauer, and T. Bauernhansl, "A Conceptual Framework for Biointelligent Production—Calling for Systemic Life Cycle Thinking in Cellular Units," *Clean Technologies*, vol. 3, no. 4, pp. 844–857, Dec. 2021.
- [16] T. Karimi, "Molecular Mechanism of Self-Organization in Biological Systems," in *Molecular Mechanisms of Autonomy in Biological Systems: Relativity of Code, Energy and Mass*, T. Karimi, Ed. Cham: Springer International Publishing, 2018, pp. 51–70.
- [17] A. Dipple, K. Raymond, and M. Docherty, "General theory of stigmergy: Modelling stigma semantics," *Cognitive Systems Research*, vol. 31–32, pp. 61–92, Dec. 2014.
- [18] M. N. Boddy and P. Russell, "DNA replication checkpoint," *Current Biology*, vol. 11, no. 23, pp. R953–R956, Nov. 2001.
- [19] M. P. Bansal, "Cell Cycle, DNA Damage Repair Systems, and Impact of Redox Regulation in Cancer," in *Redox Regulation and Therapeutic Approaches in Cancer*, M. P. Bansal, Ed. Singapore: Springer Nature, 2023, pp. 269–309.

Zheng Sun, M.Sc., Research Associate at the Institute of Mechanical Handling and Logistics, University of Stuttgart.
Phone: +49 711 685 83795,
E-Mail: zheng.sun@ift.uni-stuttgart.de
ORCID: 0009-0000-8998-6774

Jifeng Zhou, B.Sc., Student Assistant at the Institute of Mechanical Handling and Logistics, University of Stuttgart.
E-Mail: jifeng.zhou@ift.uni-stuttgart.de
ORCID: 0009-0008-3910-6300

Aya Ounissi, M.Sc., Head of the Logistics Processes Department at the Institute of Mechanical Handling and Lo-

gistics, University of Stuttgart.
Phone: +49 711 685 83935,
E-Mail: aya.ounissi@ift.uni-stuttgart.de
ORCID: 0009-0009-5739-4489

Univ.-Prof. Dr.-Ing. Robert Schulz,
Phone: +49 711 685 83771,
E-Mail: robert.schulz@ift.uni-stuttgart.de
ORCID: 0000-0003-0091-1632

Address: Institute of Mechanical Handling and Logistics,
University of Stuttgart, Holzgartenstraße 15B, 70174
Stuttgart, Germany.