Reinforcement Learning-Based WIP Control for Balancing Productivity and Lead Time in Manufacturing Systems

Silvestro Vespoli[1,*], Liberatina Carmela Santillo[1]

[1]Università degli Studi di Napoli "Federico II" – Piazzale Tecchio, 80 – 80125, Napoli

**Corresponding Author (silvestro.vespoli@unina.it)*

Abstract: In the era of Industry 4.0 and mass customization, manufacturing systems face the challenge of managing increased product variety and demand variability. Effective Work-In-Progress (WIP) control is crucial for maintaining optimal productivity and lead time in these dynamic environments. This paper proposes a novel adaptive WIP control approach for semi-heterarchical manufacturing systems using Deep Reinforcement Learning (DRL). The proposed approach leverages a Deep Q-Network (DQN) agent to learn optimal WIP control policies through interaction with a stochastic simulation environment. The DQN agent considers the current system state, processing time variability, and throughput targets to make real-time decisions and dynamically adjust WIP levels. The problem is formulated as a Markov Decision Process (MDP), and the DQN agent is trained using a custom simulation environment developed with the Simpy library. The experimental results validate the performance and adaptability of the proposed approach under different production scenarios and variability levels. The DQN-based WIP control approach demonstrates its ability to maintain the desired throughput while minimizing WIP levels, leading to improved overall performance of the manufacturing system. This research contributes to the advancement of intelligent manufacturing and provides a data-driven solution for adaptive WIP control in semi-heterarchical production systems.

Keywords: Work-In-Progress (WIP) control, Semi-heterarchical manufacturing systems, Deep reinforcement learning, Industry 4.0, Mass customization.

1. Introduction

In the era of Industry 4.0, manufacturing systems face the challenge of managing increased product customisation and demand variability, which introduce significant complexity and uncertainty in production processes (Zhong et al., 2017). Mass customisation has emerged as a key paradigm to address these challenges, enabling companies to provide personalised products and services while maintaining the efficiency and cost-effectiveness of mass production (Fogliatto et al., 2012; Salatiello et al., 2022).

Work-In-Progress (WIP) control plays a critical role in balancing productivity and lead time in manufacturing systems. Effective WIP control strategies aim to maintain optimal inventory levels to ensure smooth production flow and minimise the impact of variability (Hopp and Spearman, 2011). The CONstant Work In Progress (CONWIP) control policy has been widely adopted in manufacturing systems due to its ability to limit WIP levels and achieve a pull-based production control (Spearman et al., 1990). However, determining the appropriate WIP levels in the face of varying demand and processing times remains a significant challenge (Framinan et al., 2003).

Recent advancements in Industry 4.0 technologies, such as Cyber-Physical Systems (CPS) and the Internet of Things (IoT), have enabled the collection and analysis of real-time production data (Monostori, 2015). This data-driven approach has paved the way for the application of machine learning techniques, particularly Reinforcement Learning (RL), in production control (Wuest et al., 2016). RL has emerged as a promising approach for adaptive decision-making in complex and dynamic environments, allowing agents to learn optimal control policies through interaction with the environment (Sutton and Barto, 2020).

Several studies have explored the application of RL in manufacturing systems, showing promising results in various manufacturing control applications, including adaptive scheduling, resource allocation, and maintenance planning. For instance, Waschneck et al. (2018) applied Deep Q-Networks (DQN) for adaptive scheduling in a flexible manufacturing system, demonstrating improved performance compared to traditional heuristics. Marchesano et al. (2022) proposed a deep reinforcement learning approach for maintenance planning in a flow-shop scheduling problem, further demonstrating the versatility of RL in manufacturing optimization. Despite these advancements, the application of RL for adaptive WIP control in semi-heterarchical architectures remains largely unexplored. Moreover, existing studies often assume simplified production environments and fail to consider the impact of processing time variability on WIP control.

To address these gaps, this paper proposes a novel approach for adaptive WIP control in semi-heterarchical manufacturing architectures using RL. The proposed framework focuses on a flow-shop production system with a CONWIP control policy, where job processing times follow a variable gamma distribution. By employing a DQN agent as a model-free controller and utilising simulation-based training, optimal WIP control policies

can be derived without relying on explicit mathematical models.

The main contributions of this paper are as follows:

1. A novel RL-based framework for adaptive WIP control in semi-heterarchical manufacturing architectures, considering variable processing times.
2. A simulation environment based on the gamma distribution for modelling different realistic production variability.
3. A preliminary evaluation of the proposed approach, demonstrating its robustness and adaptability in handling variations in processing time variability and throughput targets.

The remainder of this paper is organised as follows. Section 2 reviews the relevant literature on WIP control, RL in manufacturing, and semi-heterarchical architectures. Section 3 presents the proposed methodology, including the problem formulation, the DQN agent architecture, and the simulation environment. Section 4 describes the experimental setup and results. Finally, Section 6 concludes the paper and outlines future research directions.

2. Background and Related Work

2.1 Work-In-Progress Control in Manufacturing Systems

WIP control is a critical aspect of production management that aims to maintain optimal inventory levels in manufacturing systems. Effective WIP control strategies help to balance productivity, lead time, and inventory costs by regulating the flow of material through the production process (Hopp and Spearman, 2011). Various WIP control policies have been proposed in the literature, including kanban, base stock, and CONWIP (Framinan et al., 2003).

Among these policies, CONWIP has gained significant attention due to its ability to limit WIP levels and achieve a pull-based production control (Spearman et al., 1990). In a CONWIP system, a fixed number of cards or tokens are used to control the release of jobs into the production line. When a job is completed, its associated card is returned to the beginning of the line, allowing a new job to be released (Hopp and Spearman, 2011). This mechanism ensures that the total WIP in the system remains constant, reducing the impact of variability and improving flow.

Hopp and Spearman (2011) proposed a set of performance measures for CONWIP systems under various operating conditions. For a balanced line with no variability, the best-case performance measures are given by:

$$TH_{max} = \begin{cases} \frac{w}{T_0} & if \quad w \le W_0 \\ r_b & otherwise \end{cases}$$

$$CT_{min} = \begin{cases} T_0 & if \quad w \le W_0 \\ \frac{w}{r_b} & otherwise \end{cases}$$

where w is the WIP level, T_0 is the raw processing time, W_0 is the critical WIP level, and r_b is the bottleneck rate. Here CT_{min} and TH_{max} represent the best-case (minimum) cycle time and the (maximum) throughput for a given WIP level (w), respectively.

For a balanced line with exponentially distributed processing times (i.e., the Practical Worst Case (PWC)), the performance measures are given by:

$$TH_{PWC} = \left(\frac{w}{W_0 + w - 1}\right) \cdot r_b$$

$$CT_{PWC} = T_0 + \left(\frac{w-1}{r_b}\right)$$

The introduction of the PWC serves as a benchmark for improvement targets, providing a realistic worst-case scenario against which actual system performance can be compared. Determining the optimal WIP level in a CONWIP system is a challenging task, particularly in the presence of variable demand and processing times (Framinan et al., 2003). Traditional approaches to WIP control often rely on analytical models and heuristics, which may not adequately capture the complexity and uncertainty of real-world manufacturing systems. As a result, there is a growing need for adaptive and data-driven methods that can dynamically adjust WIP levels based on the current state of the system.

2.2 Reinforcement Learning in Manufacturing

Reinforcement Learning (RL) is a branch of machine learning that focuses on learning optimal control policies through interaction with an environment (Sutton and Barto, 2020). In an RL framework, an agent learns to make decisions by receiving rewards or penalties based on the outcomes of its actions. The goal of the agent is to maximize the cumulative reward over time, which is achieved by learning a policy that maps states to actions (Wuest et al., 2016).

The RL problem can be formulated as a Markov Decision Process (MDP), defined by a tuple (S, A, P, R, γ), where S is the state space, A is the action space, P is the transition probability matrix, R is the reward function, and γ is the discount factor (Sutton and Barto, 2020). The agent's goal is to learn a policy $\pi: S \rightarrow A$ that maximizes the expected cumulative discounted reward:

$$J(\pi) = E_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$$

where r_t is the reward received at time step t.

One of the most popular RL algorithms is Q-learning (Watkins and Dayan, 1992), which learns an action-value function $Q(s,a)$ that represents the expected cumulative reward for taking action a in state s and following the optimal policy thereafter. The Q-function is updated iteratively using the following rule:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha\left[r_t + \gamma \max_a Q(s_{t+1},a) - Q(s_t,a_t)\right]$$

where α is the learning rate.

In recent years, there has been a growing body of research exploring the application of reinforcement learning techniques to adaptive WIP control. Waschneck et al. (2018) applied Deep Q-Networks (DQN) for adaptive scheduling in semiconductor production, demonstrating improved performance over traditional heuristics. DQN is an extension of Q-learning that uses a deep neural network to approximate the Q-function (Mnih et al., 2015). Dittrich and Fohlmeister (2020) proposed a cooperative multi-agent system using RL for production control in job shop environments. Overbeck et al. (2021) utilized Proximal Policy Optimization (PPO) agents for decision-making in an automated assembly system, showing improvements in decision quality and production output over time. In the context of matrix production systems, Gankin et al. (2021) implemented a DQN-based approach for modular production control, while May et al. (2021) explored an economic bidding approach using RL to increase utilisation efficiency.

These studies have shown promising results in terms of improved system responsiveness and efficiency. However, the application of RL to WIP control in semi-heterarchical architectures, particularly in environments with high variability, remains an area with significant potential for further exploration and improvement, as highlighted in our problem formulation.

2.3 Semi-Heterarchical Architectures in Manufacturing

Traditional manufacturing control architectures can be classified into two categories: centralized and decentralized (Trentesaux, 2009). Centralized architectures rely on a single decision-making entity that has complete control over the production system, while decentralized architectures distribute decision-making among multiple autonomous entities (Monostori, 2015).

Semi-heterarchical architectures have emerged as a promising approach that combines the benefits of both centralized and decentralized control (Grassi et al., 2020). In a semi-heterarchical architecture, decision-making is distributed among multiple levels, with higher levels providing global coordination and lower levels handling local execution (Bendul and Blunck, 2019). This hybrid approach allows for greater flexibility and responsiveness to changes in the production environment while maintaining a degree of central control.

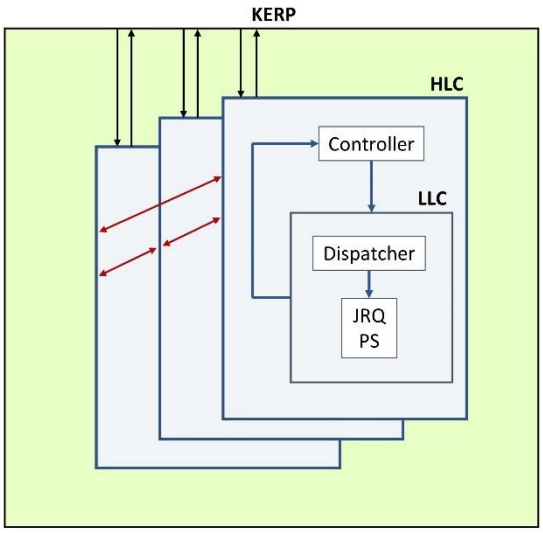

Figure 1: The semi-heterarchical architecture (inspired by (Grassi et al., 2020)

Several studies have investigated the application of semi-heterarchical architectures in manufacturing systems. Bendul and Blunck (2019) proposed a semi-heterarchical architecture for production planning and control in Industry 4.0 environments, highlighting its ability to handle complexity and uncertainty. Leitão et al. (2016) developed a semi-heterarchical architecture for self-organized manufacturing systems, demonstrating its potential for improved adaptability and robustness. Converso et al. (2015) developed a system dynamics model for bed management strategy in healthcare units, showcasing the applicability of hierarchical approaches to complex systems management beyond traditional manufacturing settingv

Grassi et al. (2020) proposed a novel semi-heterarchical architecture for manufacturing control (Figure 1), consisting of three main levels: (1) the Knowledge-based Enterprise Resource Planning (KERP) level, responsible for strategic decision-making and cloud-based coordination; (2) the High-Level Controller (HLC) level, which manages the overall performance of the production system; and (3) the Low-Level Controller (LLC) level, which handles the execution of production tasks at the shop floor level. In this architecture, the HLC plays a crucial role in managing the Work-In-Progress (WIP) levels of the production system, aiming to maintain optimal inventory levels and ensure smooth production flow.

The HLC in the semi-heterarchical architecture proposed by Grassi et al. (2020) is responsible for dynamically adjusting the WIP levels based on the current state of the production system and the performance targets set by the KERP level. This adaptive WIP control mechanism allows the system to respond effectively to changes in demand, product mix, and production variability, improving overall system performance and resilience. owever, current methods struggle in highly variable and uncertain environments. Using advanced techniques like reinforcement learning for WIP control could significantly

enhance responsiveness, efficiency, and overall performance in adaptive manufacturing control.

3 Proposed Methodology

3.1 Problem Formulation

Consider a semi-heterarchical manufacturing control architecture, as proposed by Grassi et al. (2020), where the High-Level Controller (HLC) is responsible for adaptively controlling the Work-In-Progress (WIP) levels in a flow-shop production system. The objective is to maintain optimal WIP levels that balance productivity and lead time in the face of varying demand and processing time variability.

The flow-shop production system consists of m machines arranged in series, where each job visits the machines in the same order. This setup aligns with recent research on Industry 4.0-enabled job-shop environments (Salatiello et al., 2022) and allows for the integration of digital twin concepts for enhanced system modeling and control (Rozhok et al., 2021). The processing times of jobs on each machine follow a gamma distribution, which allows for modeling a wide range of variability scenarios. The shape parameter α of the gamma distribution determines the level of variability, with $\alpha < 0.75$ representing high variability, $0.75 < \alpha < 1.25$, representing moderate variability (e.g., in mass customization scenarios) and $\alpha > 1.25$ representing low variability (e.g., in standardized production).

The HLC's decision-making problem can be formulated as a Markov Decision Process (MDP), defined by the tuple (S, A, P, R, γ), where:

- S is the state space, representing the current state of the production system. The state variables include the current WIP level (w), the critical WIP (W_0), equal to the number of machines in a balanced line, the normalized throughput target (TH_{target}), the error between the target and observed throughput (e_t), and the coefficient of variation (CV) of the job processing times. Mathematically, the state at time t can be represented as:

$$s_t = \left[w_t, W_0, TH_{target}, e_t, CV\right]$$

  The inclusion of the error term e_t in the state representation provides the agent with information about the direction and magnitude of the deviation from the target throughput, enabling it to make more informed decisions. To leverage the findings of Vespoli et al. (2023) on the generalized performance estimation approach for CONWIP flow-shop systems, the DQN agent's state space includes the normalized throughput rate (TH_{norm}), which is the actual throughput rate scaled by the mean processing time. By incorporating this normalized throughput rate, the agent can learn a WIP control policy that is independent of the specific mean processing times and can be applied to various production scenarios with different processing time distributions without the need of further training step.
- A is the action space, representing the available WIP control actions. The action $a \in A$ corresponds to the WIP level to be set in the production system.
- P is the transition probability matrix, specifying the probability of transitioning from one state to another under a given action.
- R is the reward function, quantifying the performance of the system based on the difference between the normalized observed throughput and the normalized target throughput. The reward function is designed as a Gaussian function to encourage the HLC to maintain the actual throughput close to the target:

$$R = \exp\left(-\frac{(TH_{target} - TH_{norm})^2}{2 * \sigma^2}\right)$$

  where $(TH_{target} - TH_{norm})^2$ is the quadratic absolute difference between the normalized target throughput and the normalized observed throughput, and σ is a hyperparameter controlling the width of the Gaussian function. This reward function provides a smooth gradient signal to guide the learning process.

- γ is the discount factor, balancing the importance of immediate and future rewards.

The objective is to find an optimal WIP control policy $\pi^*: S \rightarrow A$ that maximizes the expected cumulative discounted reward over an infinite horizon:

$$\pi^* = \arg\max_{\pi} E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid \pi\right]$$

where $s_t \in S$ and $a_t \in A$ are the state and action at time step t, respectively.

3.2 Deep Q-Network (DQN) for Adaptive WIP Control

To solve the formulated MDP and learn the optimal WIP control policy, we propose the use of a Deep Q-Network (DQN) as the HLC's decision-making agent. The DQN architecture consists of an input layer taking the state variables, followed by two fully connected hidden layers with ReLU activation functions, and an output layer providing Q-value estimates for each available action. The specific architecture used in this study includes:

- Input layer: 5 units (corresponding to the state variables);
- Hidden layers: 2 layers with 64 units each, ReLU activation;

- Output layer: 21 units (corresponding to the discretized action space), linear activation.

During training, the agent interacts with the simulation environment, observing states, selecting actions based on an ε-greedy exploration strategy, and receiving rewards. The experiences (state, action, reward, next state) are stored in a prioritized replay buffer, enabling efficient and stable learning. The DQN is updated using gradient descent to minimize the temporal-difference error between the predicted Q-values and the target Q-values obtained from the double Q-learning update rule.

The exploration rate is decayed over time to gradually shift from exploration to exploitation. To enhance the adaptability of the DQN agent, an adaptive sigma mechanism is introduced. The width of the Gaussian reward function, controlled by the hyperparameter σ, is dynamically adjusted during training based on the current training iteration. The value of σ starts at a higher value (e.g., 0.15) to encourage broader exploration in the early stages of training and gradually decreases to a lower value (e.g., 0.08) to focus on fine-tuning the policy in the later stages. This adaptive sigma mechanism allows the agent to effectively explore the state-action space and converge to a robust WIP control policy.

4 Experimental Setup and Results

4.1 Simulation Environment and Training Procedure

The flow-shop production system is simulated using a custom environment developed with the Simpy library in Python. The environment is designed to be highly configurable, allowing for the modeling of different flow-shop configurations and processing time variability. The experiments utilize the RLlib library, a scalable reinforcement learning framework built on top of Ray, which provides a wide range of reinforcement learning algorithms, including DQN, and offers distributed training capabilities for improved performance.

The training procedure for the DQN agent involves the following steps:

1. Initialize the DQN agent with random weights and the experience replay buffer D.
2. For each episode:
   - Reset the simulation environment to its initial state s_0.
   - For each time step t:
     - Observe the current state s_t.
     - Select an action a_t using the ε-greedy strategy based on the current Q-network.
     - Apply the action a_t to the simulation environment and observe the next state s_{t+1} and reward r_t.
     - Store the experience tuple (s_t, a_t, r_t, s_{t+1}) in the replay buffer D.
     - Sample a mini-batch of experiences from D and perform a training step:
       - Compute the target Q-values using the double Q-learning update rule.
       - Update the Q-network parameters using gradient descent to minimize the temporal-difference error.
       - Update the priorities of the sampled experiences based on the absolute temporal-difference error.
3. Repeat the training process for a specified number of episodes test on scenarios with different variability levels and throughput targets.

The hyperparameters of the DQN agent, such as the learning rate, discount factor, and exploration rate, are tuned using Ray Tune, a scalable hyperparameter tuning library, to ensure the best performance. During training, the agent's performance is monitored using TensorBoard, a visualization toolkit for machine learning.

4.2 Training Results

The training progress of the DQN agent is evaluated using several metrics, including the gradient norm, temporal-difference (TD) error, and episode statistics (minimum, mean, and maximum reward).

Figure 2 shows the gradient norm and the the TD error during training. The gradient norm measures the magnitude of the gradients used to update the Q-network parameters. A stable and decreasing gradient norm indicates that the agent is converging towards an optimal policy. The results demonstrate that the gradient norm stabilizes and decreases over time, suggesting successful

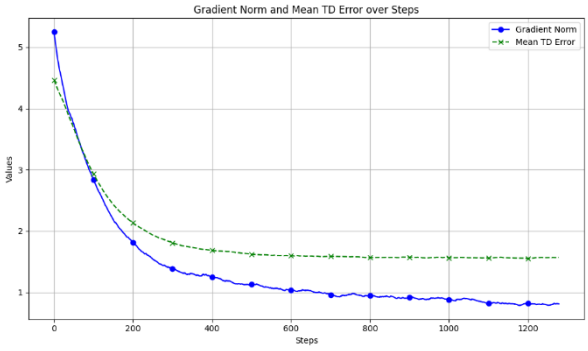

Figure 2 – Gradient Norm and Temporal Difference (TD) over steps during training

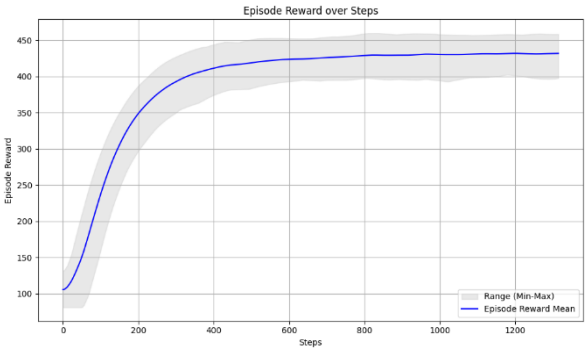

Figure 3 - Episode rewards over time for the DQN agent, displaying the range and mean values

learning of the WIP control policy. The TD error, instead, represents the difference between the predicted Q-values and the target Q-values obtained from the double Q-learning update rule. A decreasing TD error indicates that the agent's predictions are becoming more accurate and consistent with the observed rewards. The results show a steady decrease in the TD error, confirming the effectiveness of the learning process.

Figure 3 summarizes the episode statistics, illustrating the minimum, mean, and maximum rewards achieved by the DQN agent during training. The shaded area represents the range between the minimum and maximum rewards, while the solid line indicates the mean reward per episode. The episode reward reflects the cumulative reward obtained by the agent in each episode. The increasing trend in the mean episode reward, along with the narrowing range, suggests that the agent is learning to make better decisions and stabilize the WIP control policy over time.

4.3 Validation Results

To validate the performance of the trained DQN agent, several test scenarios are designed with varying throughput targets and processing time variability. The agent's ability to adapt to changing production conditions and maintain the desired throughput while minimizing WIP levels is evaluated.

Figure 4 illustrates the WIP control behavior of the DQN agent for different throughput targets. The results demonstrate that the agent effectively adjusts the WIP levels to maintain the actual throughput close to the target throughput. When the target throughput is increased, the agent responds by allowing higher WIP levels to meet the demand. Conversely, when the target throughput is decreased, the agent reduces the WIP levels to minimize inventory and maintain efficiency.

Figure 5 presents the WIP control performance of the DQN agent under different levels of processing time variability, represented by the coefficient of variation (CV). The results show that the agent adapts its WIP control policy based on the processing time variability. When the variability is high (lower CV), the agent maintains slightly higher WIP levels to buffer against the increased uncertainty and maintain the target throughput. Conversely, when the variability is low (higher CV), the agent reduces the WIP levels to minimize inventory while still achieving the desired throughput.

These validation results demonstrate the adaptability and robustness of the proposed DQN-based WIP control approach. The agent effectively learns to make dynamic decisions based on the current production conditions, considering the throughput targets and processing time variability. The ability to adapt the WIP levels in real-time enables the manufacturing system to maintain a balance between productivity and lead time, leading to improved overall performance.

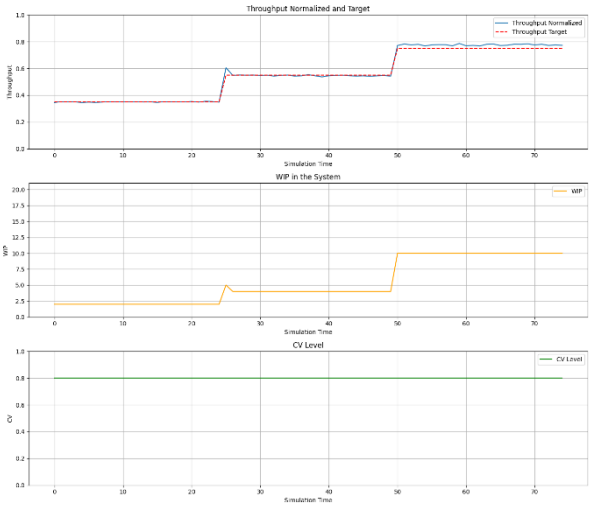

Figure 4 - The DQN agent's WIP control, with changing TH_{target} value

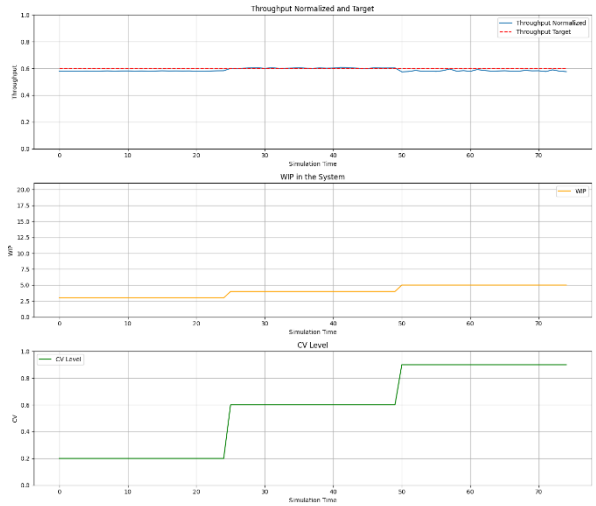

Figure 5 - The DQN agent's WIP control, with changing CV value

5 Conclusion

In this paper, we proposed a novel adaptive WIP control approach for semi-heterarchical manufacturing systems using deep reinforcement learning. The proposed approach addresses the challenges of managing production variability and meeting throughput targets in the context of Industry 4.0 and mass customization. By leveraging the capabilities of Deep Q-Networks (DQN) and integrating them with a stochastic simulation environment, the proposed approach enables real-time decision-making and dynamic adjustment of WIP levels based on the current production conditions.

The experimental results validate the performance and adaptability of the DQN-based WIP control approach. The training results show stable learning progress, with the agent converging towards an optimal WIP control policy. The validation results demonstrate the agent's ability to adapt to changing production conditions, such as varying throughput targets and processing time variability. By dynamically adjusting the WIP levels, the proposed

approach enables the manufacturing system to maintain a balance between productivity and lead time, leading to improved overall performance.

The robustness of the proposed approach is evidenced by its adaptability to various processing time variabilities and throughput targets. However, practical implementation may face challenges such as computational resource requirements and the need for high-quality historical data. Future research should focus on extending the approach to more complex manufacturing systems, incorporating additional production objectives, and developing strategies to reduce implementation barriers, making the system more accessible to a wider range of manufacturing environments.

References

Bendul, J. C. and Blunck, H. (2019), 'The design space of production planning and control for industry 4.0', *Computers in Industry* **105**, 260--272.

Converso, G., di Giacomo, S., Murino, T., Rea, T. (2015), 'A system dynamics model for bed management strategy in health care units', *Communications in Computer and Information Science* 532, 610--622.

Dittrich, M.-A. and Fohlmeister, S. (2020), 'Cooperative multi-agent system for production control using reinforcement learning', *CIRP Annals* **69**(1), 389--392.

Fogliatto, F. S., da Silveira, G. J. and Borenstein, D. (2012), 'The mass customization decade: An updated review of the literature', *International Journal of Production Economics* **138**(1), 14--25.

Framinan, J. M., González, P. L. and Ruiz-Usano, R. (2003), 'The CONWIP production control system: Review and research issues', *Production Planning & Control* **14**(3), 255--265.

Gankin, D., Mayer, S., Zinn, J., Vogel-Heuser, B. and Endisch, C. (2021), 'Modular production control with multi-agent deep Q-learning', In *26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Vasteras, Sweden, Sep., 1--8.

Grassi, A., Guizzi, G., Santillo, L. C. and Vespoli, S. (2020), 'A semi-heterarchical production control architecture for industry 4.0-based manufacturing systems', *Manufacturing Letters* **24**, 43--46.

Hopp, W. and Spearman, M.Mark L. Spearman, ed., (2011), *Factory Physics*, Waveland Pr Inc, Long Grove, Ill.

Leng, J., Yan, D., Liu, Q., Xu, K., Zhao, J. L., Shi, R., Wei, L., Zhang, D. and Chen, X. (2020), 'ManuChain: Combining Permissioned Blockchain With a Holistic Optimization Model as Bi-Level Intelligence for Smart Manufacturing', *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **50**(1), 182--192.

Marchesano, M.G., Staiano, L., Guizzi, G., Castellano, D., Popolo, V. (2022), 'Deep Reinforcement Learning Approach for Maintenance Planning in a Flow-Shop Scheduling Problem', *Frontiers in Artificial Intelligence and Applications* 355, 385--399.

May, M. C., Kiefer, L., Kuhnle, A., Stricker, N. and Lanza, G. (2021), 'Decentralized multi-agent production control through economic model bidding for matrix production systems', *Procedia CIRP* **96**, 3--8.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S. and Hassabis, D. (2015), 'Human-level control through deep reinforcement learning', *Nature* **518**(7540), 529--533.

Monostori, L. (2015), 'Cyber-physical production systems: roots from manufacturing science and technology', *at - Automatisierungstechnik* **63**(10), 766--776.

Overbeck, L., Hugues, A., May, M. C., Kuhnle, A. and Lanza, G. (2021), 'Reinforcement learning based production control of semi-automated manufacturing systems', *Procedia CIRP* **103**, 170--175.

Rozhok, A.P., Zykova, K.I., Suschev, S.P., Revetria, R. (2021), 'The use of digital twin in the industrial sector', *IOP Conference Series: Earth and Environmental Science* 815(1), 012032.

Salatiello, E., Guizzi, G., Marchesano, M.G., Santillo, L.C. (2022), 'Assessment of performance in Industry 4.0 enabled Job-Shop with a due-date based dispatching rule', *IFAC-PapersOnLine* 55(10), 2635--2640.

Spearman, M., Woodruff, D. and Hopp, W. (1990), 'CONWIP: a pull alternative to kanban', *International Journal of Production Research* **28**(5), 879--894.

Sutton, R. S.Andrew Barto, ed., (2020), *Reinforcement learning - An Introduction*, The MIT Press, Cambridge, Massachusetts.

Trentesaux, D. (2009), 'Distributed control of production systems', *Engineering Applications of Artificial Intelligence* **22**(7), 971--978.

Vespoli, S., Grassi, A., Guizzi, G. and Popolo, V. (2023), 'Generalised Performance Estimation in Novel Hybrid MPC Architectures: Modeling the CONWIP Flow-Shop System', *Applied Sciences* **13**(8), 4808.

Waschneck, B., Reichstaller, A., Belzner, L., Altenmüller, T., Bauernhansl, T., Knapp, A. and Kyek, A. (2018), 'Optimization of global production scheduling with deep reinforcement learning', *Procedia CIRP* **72**, 1264--1269.

Watkins, C. J. C. H. and Dayan, P. (1992), 'Q-learning', *Machine Learning* **8**, 279--292.

Wuest, T., Weimer, D., Irgens, C. and Thoben, K.-D. (2016), 'Machine learning in manufacturing: advantages, challenges, and applications', *Production & Manufacturing Research* **4**(1), 23--45.

Zhong, R. Y., Xu, X., Klotz, E. and Newman, S. T. (2017), 'Intelligent Manufacturing in the Context of Industry 4.0: A Review', *Engineering* **3**(5), 616--630.