Reinforcement Learning based Multi-connectivity Resource Allocation in Factory Automation Systems

Mohammad Farzanullah*, Hung V. Vu† and Tho Le-Ngoc*
* Department of Electrical & Computer Engineering, McGill University, Montréal, QC, Canada
† Huawei Technologies Canada, Ottawa, ON, Canada

Abstract—We propose joint user association, channel assignment and power allocation for mobile robot Ultra-Reliable and Low Latency Communications (URLLC) based on multi-connectivity and reinforcement learning. The mobile robots require control messages from the central guidance system at regular intervals. We use a two-phase communication scheme where robots can form multiple clusters. The robots in a cluster are close to each other and can have reliable Device-to-Device (D2D) communications. In Phase I, the APs transmit the combined payload of a cluster to the cluster leader within a latency constraint. The cluster leader broadcasts this message to its members in Phase II. We develop a distributed Multi-Agent Reinforcement Learning (MARL) algorithm for joint user association and resource allocation (RA) for Phase I. The cluster leaders use their local Channel State Information (CSI) to decide the APs for connection along with the sub-band and power level. The cluster leaders utilize multi-connectivity to connect to multiple APs to increase their reliability. The objective is to maximize the successful payload delivery probability for all robots. Illustrative simulation results indicate that the proposed scheme can approach the performance of the centralized algorithm and offer a substantial gain in reliability as compared to single-connectivity (when cluster leaders are able to connect to 1 AP).

Index Terms—Multi-connectivity, Factory automation, Resource allocation, Reinforcement Learning

I. INTRODUCTION

5G is envisioned to support a variety of key services which includes Enhanced Mobile Broadband (eMBB), massive Machine-Type Communications (mMTC), and Ultra-Reliable and Low Latency Communications (URLLC). URLLC will be relevant to many emerging domains such as wireless control and automation in factory environments, Cellular Vehicle-to-Everything (C-V2X) communications, and tactile internet [1]. Resource allocation (RA) design for URLLC is challenging due to the high-reliability requirement (99.999%) coupled with low latency (1 ms) requirements [2].

Industrial automation is one of the key sectors where 5G aims to provide connectivity. In the industrial setting, wired connections are currently used. However, the factory of the future envisions to have a dynamic environment with multiple mobile robots due to which the wired solutions will not be viable. Motivated by such limitations, communications in the industrial setting is envisioned to migrate from wired to wireless communications. To guarantee a safe and efficient operation, emerging wireless technologies must support URLLC services. The factory of the future will consist of multiple mobile robots requiring URLLC [3]. A mobile robot is programmable and is able to perform various activities in an industry e.g., an Automated Guided Vehicle (AGV) is an autonomous vehicle to move materials within a floor. The mobile robots would require control messages from a central guidance system to avoid collisions and manage jobs. Since the robots require URLLC, an intelligent RA design is necessary.

Achieving URLLC in industrial automation systems is challenging due to fading. To address this challenge, diversity has been proposed as an attractive solution to combat the fading effect in a wireless environment. The probability that multiple paths experience a deep fade simultaneously is lower, decreasing the probability of error and increasing the reliability of the system. In the existing literature, recent works [2], [4] have demonstrated that multi-connectivity is a promising solution to enable URLLC. Multi-connectivity uses space and/or frequency diversity by allowing the receiver to connect to multiple transmitters (e.g., base-stations (BSs) or access-points (APs)) on either the same or different frequencies. Further, extremely high data rates (correspondingly low latency) can be obtained via exploiting multi-connectivity in URLLC networks [5], [6] develops RA strategy in multi-user multi-cell systems with shared spectrum resources. Based on a matching theory approach, the proposed RA algorithm jointly performs link selection and sub-band scheduling in which multiple resources (sub-bands) are mapped to one user.

In the context of factory automation, the multi-connectivity problem is considered, together with dynamic channel allocation for interference management in a so-called campus network where the wireless communications in factory is impaired by the external interference [7]. Reinforcement learning (RL) is adopted to resolve the joint user association and channel assignment problem. The authors in [8] use RL in a factory setting for time-slot selection for packet transmission. However, in [6]–[8], the authors do not take into consideration power allocation, which is certainly beneficial to manage the interference.

Additionally, there is a strong line of work [9], [10] that utilizes two-hop transmission for message relaying. This stems from an observation that, due to the fading, not all robots can establish a reliable direct connection to an AP. To overcome this drawback, these works propose collaborative schemes that allow to extend the network coverage. In a factory setting, there are many devices, making it difficult to achieve URLLC with conventional broadcasting techniques. Inspired by these works, we develop a two-phase transmission strategy for a mobile robot scenario in a factory automation system. Under this strategy, the robots form multiple clusters, each consisting of one leader and multiple members. In the mobile robot scenario, the robots work in close proximity and can form reliable D2D connections. The D2D connections will have stronger channels as compared to AP-to-robot connections.
The leader would receive control messages for its cluster from the APs in Phase I and will relay it to its members in Phase II.

Moreover, the RA techniques using a central controller are not practical in a mobile URLLC setting due to stringent latency and reliability. The mobility of the robots causes fast channel variations along with the increase in latency to transmit CSI to the central controller. Furthermore, traditional optimization algorithms have been used for RA in factory automation systems [9]. However, these methods require a large number of iterations to reach a satisfying solution and are sensitive to change in system parameters. Recent works have adopted distributed RL approach for RA in multiple domains, e.g., in cellular communications [11], and C-V2X [12].

Reinforcement Learning (RL) is a sub-field of Machine Learning where an agent is able to learn in an interactive environment by trial and error. The intelligent agents takes an action in an environment and receives a reward. In RL, the aim of the agent is to maximize the cumulative reward $G_t$ it receives in the long run. Q-learning is a model-free RL algorithm that learns the value of an action $a$ in a state $s$. It iteratively updates the action-value function for each state-action pair $(s, a)$ until they converge to the optimal action-value function $Q^*(s, a)$. The Deep Q-Learning [13] uses a Deep Neural Network (DNN) as a function approximator to solve for the optimal policy. The parameters of the DNN learn to approximate the Q-function in a Deep Q-Network (DQN).

This paper develops a two-phase scheme for control messages delivery from the APs to the robots in a factory automation system. The robots form multiple clusters, each consisting of one leader and multiple members. During Phase I, the APs combine the messages for the whole cluster and deliver them to the cluster leader. While in Phase II, the leader broadcasts this message to its members. We develop a MARL based distributed algorithm for Phase I, where the cluster leader acts as an agent and performs joint user association, channel assignment, and power allocation. Furthermore, we utilize dual-connectivity where each leader is able to connect to a maximum of two APs at the same time in order to enhance reliability. The agents only use their local CSI and learn to optimally decide the APs to connect to along with the sub-band and power level. Our optimization objective is reliability, which is formally defined as the average probability of successful payload delivery within a required latency. Simulation results indicate that the proposed scheme outperforms the MARL single-connectivity [14] and random allocation, and achieves result close to that of centralized algorithm, even outperforming it in some cases.

II. SYSTEM MODEL AND PROBLEM FORMULATION

Fig. 1 shows the system model, where we consider the downlink communications in a single-floor factory setting. The floor consists of multiple robots, each of which needs to receive a control message from the central controller. There are $K$ Access Points (APs) in the factory that are required to deliver control messages to all the robots. Further, we assume that there are $M$ sub-bands to communicate on. We assume that the robots are in close proximity due to which they form $N$ clusters for reliable communication. A cluster $n$ consists of one leader and $O$ members. We denote the set of members in cluster $n$ as $V_n$, $n = 1, \ldots, N$, i.e., there are $N$ clusters with $O$ members in each cluster. Moreover, each robot can be part of single cluster, i.e., $V_n \cap V_i = \emptyset$ for $n \neq i$. We assume that the members can have a maximum distance of $d$ meters from their cluster leader. Each robot requires a payload of $B_o$ bytes within a latency of $T$ seconds. We develop a two-phase communication approach where the APs deliver the control messages to the cluster leader in the first phase and the cluster leader relays these messages to the cluster members in the second phase via point-to-multipoint D2D connections. In Phase I, the payload for the cluster is combined and the APs transmit this payload to the cluster leader. Since the payload is for the whole cluster, it consists of $B = B_o \times (O+1)$ bytes. In the second phase, the cluster leaders broadcast the received messages to its members. The set of APs, clusters and sub-bands available is denoted by $\mathcal{K}$, $\mathcal{N}$ and $\mathcal{M}$ respectively. We assume that both the APs and the robots are equipped with a single antenna. Furthermore, we assume $M = N/2$ which necessitates spectrum sharing.

In Phase I, the communication is from the APs to the cluster leader. We consider multi-connectivity in Phase I, i.e., the cluster leader is able to connect to multiple APs simultaneously. The APs are connected to the central controller, which can coordinate the synchronous transmission of control messages to the cluster leader. The cluster leader needs to decide the APs it needs to connect to along with the sub-band and power level. In our work, we assume that in the case of multi-connectivity, the cluster leader can connect to a maximum of two APs, and the assigned radio-bands are located on the same sub-bands. Suppose that a cluster leader $n$ is served by the AP in sub-band $s$, then the received signal-to-interference-plus-noise ratio (SINR) at leader $n$ in sub-band $m$ is given by:

$$
\text{SINR}_{n}^{(c)}[m] = \frac{\sum_{k \in \mathcal{K}_n} P_{kn}^{(c)} L_{kn}^{(c)} g_{kn}^{(c)}[m]}{\sum_{i \in \mathcal{K}} \sum_{j \neq n} P_{ij}^{(c)} L_{ij}^{(c)} g_{ij}^{(c)}[m] + \sigma^2}
$$

(1)
where \( P_{xy}^{(c)} \) refers to the power from transmitter \( x \) to cluster leader \( y \). \( \sigma \) refers to the noise power at the receiver. Moreover, \( g_{ab}^{(c)} \) refers to the small-scale fading in the AP-to-leader link \( ab \) and \( L_{ab}^{(c)} \) refers to the large-scale fading in the link \( ab \). We consider that the small-scale fading power follows the Rayleigh distribution. The large-scale fading consists of both the path loss and shadowing. \( P_{ij}^{(c)}[m] \) indicates the indicator function which is set to 1 if link \( ij \) reuses the sub-band \( m \), and 0 otherwise.

The achievable throughput of leader \( n \) in sub-band \( m \) is given by:

\[
R_n^{(c)} = W \log_2(1 + \text{SINR}_n^{(c)}[m])
\]

where \( W \) is the bandwidth of each spectrum sub-band.

Phase II consists of the communication from the cluster leader to members. The cluster leader broadcasts the payload received to all of the members. We assume that each member can decode its message from the combined payload. A member \( o \) will be affected by the interference from the leaders that transmit using the same sub-band. The SINR at the receiver of member \( o \in O \) is given by:

\[
\text{SINR}_{no}^{(d)} = \frac{P_{no}^{(d)}}{\sum_{i \neq n} P_{io}^{(d)} + g_{io}}[m] + \sigma^2
\]

where \( P_{no}^{(d)} \) denotes the power from transmitter \( n \) (i) to receiver \( o \) in sub-band \( m \).

The throughput of member \( o \) of cluster \( n \) is given by:

\[
R_o^{(d)} = W \log_2(1 + \text{SINR}_{no}^{(d)})
\]

### A. Problem Formulation

In this work, the objective is to reliably transmit the control messages from the APs to the robots. Each robot is required to receive a payload of size \( B \) bytes within a latency of \( T \) seconds. Phase I consists of \( T_1 \) seconds during which all the cluster leaders need to receive the control messages. Mathematically, it can be stated as \( P \left( \sum_{t = 1}^{T_1} \Delta_T R_n^{(c)}(t) \geq B \right) \). where \( \Delta_T \) is the time duration of one time slot \( t \). Hence, this is an optimization problem where for each cluster leader \( n \), we need to find the APs to connect to along with the sub-band and power level.

For Phase II, the formulation can be mathematically stated as \( P \left( \sum_{t = T_1}^{T_2} \Delta_T R_n^{(d)}(t) \geq B \right) \). In this Phase, we assume that the leaders transmit to the members on different resource blocks and this will be discussed in Section III. The members are close to the leaders due to which reliable transmission can be achieved.

### III. RESOURCE ALLOCATION USING DEEP REINFORCEMENT LEARNING

#### A. Multi-Agent Resource Allocation Algorithm

In this section, we formulate the multi-agent RL problem. The objective is to maximize the reliable transmission of control messages from the AP to the robots. Each of the cluster leader acts as an agent. At each time-step, an agent observes the environment, takes an action, and receives a reward. The agent learns to optimally allocate the resources by this trial and error method. We use RL for Phase I only.

The multi-agent RL problem consists of two phases that are training and testing. The training is centralized and a common reward is used to ensure collaboration between the agents. This common reward is available to all the agents during the training phase. Based on the reward, the agents train their distinct DQN to fulfill the common objective. If each of the robots receives a separate reward, it would become a competitive game where each of them will try to maximize their own throughput, resulting in increased interference. Meanwhile, during the testing phase, the agents use the trained DQN to select the action. The state space, action space, and the reward design are outlined below:

1) **State Space:** The state space of the agent consists of the measurements from the last time-step. Let \( Z_n(t) \) denote the state of agent \( n \) at time-step \( t \). The state space consists of the following measurements: the direct channels from all the APs \( k \in K \) to the cluster leader \( n \), i.e., \( \{L_{kn}^{(c)}[c], g_{kn}[c]\} \), the aggregate interference and noise powers at the agent \( n \), i.e., \( \{\sum_{i \in K} \sum_{j \neq n} P_{ij}^{(c)}[m]F_{ij}^{(c)}L_{ij}^{(c)}[c]g_{ij}^{(c)}[m] + \sigma^2\} \), the remaining payload and time limitation after the current time-step, the training iteration number \( e \), and the probability of random action selection \( e \) of agent \( n \).

2) **Action Space:** Each agent \( n \) has an identical action space. The action space consists of the APs selected along with the sub-band and power level. We limit the connection of the cluster leader to a maximum of 2 APs at time-step \( t \). Moreover, to reduce the complexity, we assumed that in case the cluster leader decides to connect to 2 APs, it would be assigned the same sub-band. This reduces the complexity of the DQN, though at the costs of higher interference. The power level is divided into discrete levels in the range \([0, P_d]\) where \( P_d \) is the maximum allocated power.

3) **Reward Design:** A common reward is used between the agents to ensure collaboration between them. We design the reward function correlating to the final objective, i.e., maximizing the successful payload delivery probability to the robots. The reward is designed as the sum of the rate of all of the cluster leaders. The rate at a cluster leader \( n \) at time-step \( t \) is thus defined as:

\[
U_n^{(c)}(t) = \begin{cases} R_n^{(c)}(t), & \text{if } B_n(t) \geq 0, \\ 0, & \text{otherwise}. \end{cases}
\]

where \( U \) needs to be empirically adjusted. Experiments show that the value of \( U \) should be greater than the achievable rate at leader \( n \), and lesser than two times of achievable rate. Meanwhile, \( B_n \) is the remaining payload for cluster leader \( n \) at time-step \( t \). The common reward at time-step \( t \) is defined as:

\[
r_t = \sum_{n=1}^{N} U_n^{(c)}(t)
\]

### B. Training Algorithm and Testing Strategy

There is a time limitation of \( T = T_1 + T_2 \) seconds to complete the transmission to all of the robots. The DQN is
designed as an episodic setting, where each episode is of time duration $T$. The position of the robots and the large-scale fading is updated at the start of each episode, while the small-scale fading is updated at each time-step. At each time-step, the variation in small-scale fading changes the dynamics of the system, causing the agents to adjust their actions. The training phase consists of only Phase I. Deep Q-Learning is used to train the agents and the algorithm is outlined in Algorithm 1.

**Algorithm 1 Training Algorithm**

1. Initiate the environment
2. Initiate the parameters $\theta$ for DNN of all cluster leaders
3. for each episode do
4. for each time-step $t$ do
5. Update the device locations and large-scale fading
6. Update small-scale fading
7. for each cluster leader $n$ do
8. Observe the state $s_t$
9. end for
10. The leaders take action according to the $\epsilon$-greedy policy and obtain a common reward $r_t$
11. for each cluster leader $n$ do
12. Observe the next state $s_{t+1}$
13. Store $<s_t,a_t,r_t,s_{t+1}>$ in the replay memory
14. end for
15. end for
16. for each cluster leader $n$ do
17. Sample a mini-batch $X$ from the replay memory
18. Train the deep Q-network for cluster leader $n$ using $X$
19. end for
20. end for

The testing phase consists of both Phase I and Phase II. In Phase I, the agents observe the state space $Z_{e}(t)$ and use the trained DNN to output the optimal action at each time-step. After Phase I, a fixed scheme is used for Phase II where each cluster leader communicates to its members in different time-frequency slots, i.e., on separate sub-band and time-step combination, which eliminates the need for interference management. The robots can reliably communicate using this scheme since the distance between the cluster leader and members is less.

**IV. ILLUSTRATIVE RESULTS**

We consider downlink communication for a factory automation system that consists of multiple robots. The robots need to periodically receive the control messages from the central guidance system. The robots can perform multiple operations in a factory and can have high mobility. Further, we assume robots form clusters, and each cluster works together to complete a specific task.

We consider a factory floor setting of dimension $40 \times 40$ m$^2$. There are 4 APs in the factory floor setting with coordinates $(10,10)$, $(10,30)$, $(30,10)$ and $(30,30)$ assuming the south-east corner to be the $(0,0)$ coordinate. Further, there are $N$ clusters on the floor with $O$ members in each. Further, we assume that the members are at a maximum distance of $d$ meters from the leader. The robots can move with a velocity of 1 m/s and each cluster moves either to the north, south, east or west with an equal probability. We consider both large-scale and small-scale fading in our simulations. The large-scale and small-scale fading is updated every 1 ms and 0.1667 ms respectively. For large-scale fading, we consider both path loss and shadowing. For path loss we consider the A1 indoor scenario of the WINNER II channel model [15]. We assume log-normal shadowing with a standard deviation of 3 dB. Moreover, we assume Rayleigh fading for small-scale fading. The simulation parameters are highlighted in Table I.

We used Python to develop the simulation, and implemented the DQN using the Tensorflow framework. There are 3 hidden layers in the DNN for DQN, with 83, 41, and 20 nodes respectively. RMSProp was used as the optimizer with a learning rate of 0.001 while rectified linear unit (ReLU) was used as the activation function. The training phase consists of 6000 episodes, where each episode consists of 6 time-steps. Meanwhile, the testing phase consists of 1000 episodes. For URLLC applications, we consider a sub-frame of 1 ms which is further divided into 6 sub-slots [16]. We use the $\epsilon$-greedy strategy during the training phase, where the value of $\epsilon$ was linearly reduced from 1 to 0.02 for 80% of the training and was fixed at 0.02 after that. The discount factor for DQN is set at 0.9. The value of $U$ defined in Eq. 5 was set as 40. In the training phase, we fixed the value of robot payload size $B_o$ as 100 bytes, while we varied the payload size between 20 bytes and 100 bytes for the testing phase [17].

We develop five benchmarks for comparison with our proposed strategy:

- **Random Allocation**: Each cluster leader connects to the nearest AP and randomly selects the sub-band and power level.
- **Centralized Algorithm**: Each cluster leader connects to the nearest AP and exhaustively searches for the sub-band and power level that maximizes the sum-rate.
- **Single-connectivity MARL**: We run the RL algorithm with the restriction that each of the cluster leaders can connect to just one AP.
- **Greedy single-connectivity**: Each cluster leader using the single best link available to transmit at maximum power.
- **Greedy multi-connectivity**: Each cluster leader using the two best links available to transmit at maximum power.

Fig. 2 shows the average successful payload delivery probability versus the payload size. The proposed algorithm was trained for $N = 4$ and the payload size $B_o$ was varied during the testing phase. Our proposed algorithm achieves a success-
ful payload delivery probability of 1 for payload size up to 60 bytes, and more than 0.999 for up to 100 bytes, outperforming both the centralized algorithm and the RL algorithm with single-connectivity. Moreover, it is to be noted that for the greedy approach, which is a naive strategy, single-connectivity performs better than multi-connectivity. This is due to higher interference when all of the cluster leaders attempt to use multiple links to complete their transmission. However, for RL, there is a substantial improvement in performance by utilizing multi-connectivity. This shows that multi-connectivity can yield a greater benefit only with an intelligent RA design.

Fig. 3 shows the effect of the number of clusters $N$ on the performance for the payload $B_0$ of 40 bytes and 60 bytes. For all cases, the average robot payload delivery probability is higher than 0.995, and degraded when $N > 6$, mainly due to the increasing interference among the cluster leaders. Furthermore, the training for MARL becomes more challenging as we increase the number of agents.

Fig. 2. Average robot payload delivery probability versus number of clusters $N=4$

Fig. 3. Average robot payload delivery probability versus number of clusters $N$

V. CONCLUSION

The paper utilized multi-connectivity based user association and resource allocation for a mobile robot scenario in a factory automation system. A two-phase scheme was considered for message relaying, where in Phase I the APs deliver control messages to the cluster leader and in Phase II the cluster leader broadcasts the messages to its members. For Phase I, we developed a distributed MARL algorithm for joint user association, channel assignment, and power allocation. For user association, each cluster leader was able to connect to multiple APs to enhance transmission reliability. The proposed algorithm achieved results close to that of the centralized algorithm. Moreover, it was shown that utilizing multi-connectivity offers substantial gain in reliability. In the future, cluster formation and cluster leader selection can be optimized, or multiple antennas at AP can be considered for beam-forming.

ACKNOWLEDGMENT

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada and in part by Huawei Technologies Canada.

REFERENCES

[1] Z. Li, M. A. Uusitalo, H. Shariatmadari, and B. Singh, “5G URLLC: Design challenges and system concepts,” in 2018 15th International Symposium on Wireless Communication Systems (ISWCS). IEEE, 2018, pp. 1–6.

[2] M. Bennis, M. Debbah, and H. V. Poor, “Ultrareliable and low-latency wireless communication: Tail, risk, and scale,” Proceedings of the IEEE, vol. 106, no. 10, pp. 1834–1853, Oct. 2018.

[3] “3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Study on Communication for Automation in Vertical Domains (release 16),” 3GPP. Tech. Rep., Jul. 2020.

[4] P. Popovski, Č. Stefanović, J. J. Nielsen, E. De Carvalho, M. Angelichinoski, K. F. Trillingsgaard, and A.-S. Bana, “Wireless Access in Ultra-Reliable Low-Latency Communication (URLLC),” IEEE Trans. Commun., vol. 67, no. 8, pp. 5783–5801, Aug. 2019.

[5] P. Mishra, S. Kar, V. Bollapragada, and K.-C. Wang, “Multi-connectivity using nr-dc for high throughput and ultra-reliable low latency communication in 5g networks,” in 2021 IEEE 4th 5G World Forum (5GWF). IEEE, 2021, pp. 36–40.

[6] T. Höfler, P. Schulz, E. A. Jorswieck, M. Simsek, and G. P. Fettweis, “Stable Matching for Wireless URLLC in Multi-Cellular, Multi-User Systems,” IEEE Trans. Commun., vol. 68, no. 8, pp. 5228–5241, Aug. 2020.

[7] B. Khodapanah, T. Höfler, B. Yuncu, A. N. Barreto, M. Simsek, and G. Fettweis, “Coexistence Management for URLLC in Campus Networks via Deep Reinforcement Learning,” in IEEE Wireless Commun. Networking Conf. (WCNC), 2020, pp. 1–6.

[8] J. Burgueño, R. Adeogun, R. L. Bruun, C. S. M. García, L. de la Rondera, and R. Barco, “Distributed deep reinforcement learning resource allocation scheme for industry 4.0 device-to-device scenarios,” in 2021 VTC2021-Fall. IEEE, 2021, pp. 1–7.

[9] L. Liu and W. Yu, “A D2D-based Protocol for Ultra-Reliable Wireless Communications for Industrial Automation,” IEEE Trans. Wireless Commun., vol. 17, no. 8, pp. 5045–5058, Aug. 2018.

[10] R. S. Khosravirad, H. Viswanathan, and W. Yue, “Exploiting Diversity for Ultra-Reliable and Low-Latency Wireless Control,” IEEE Trans. Wireless Commun, Sep. 2020.

[11] Y. S. Nasir and D. Guo, “Deep Reinforcement Learning for Joint Spectrum and Power Allocation in Cellular Networks,” arXiv preprint arXiv:2012.10682, 2020.

[12] L. Liang, H. Ye, and G. Y. Li, “Spectrum Sharing in Vehicular Networks Based on Multi-Agent Reinforcement Learning,” IEEE J. Sel. Areas Commun., vol. 37, no. 10, pp. 2282–2292, Oct. 2019.

[13] C. J. C. H. Watkins, “Learning from delayed rewards,” Ph.D. dissertation, King’s College, Cambridge United Kingdom, May 1989.

[14] M. Farzanullah, H. V. Vu, and T. Le-Ngoc, “Deep reinforcement learning for joint user association and resource allocation in factory automation,” in IEEE Wireless Commun. Networking Conf. (WCNC). IEEE, 2022, pp. 2059–2064.

[15] P. Kyosti, “WINNER II channel models,” IST. Tech. Rep. IST-4-027756 WINNER II D1. 1.2 VI. 2, 2007.

[16] T. Fehrenbach, R. Datta, B. Göktepe, T. Wirth, and C. Hellige, “URLLC services in 5G low latency enhancements for LTE,” in 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), 2018, pp. 1–6.

[17] “5G for Connected Industries and Automation (White Paper-Second Edition),” 5G-ACIA, Feb. 2019.