On Optimal Power Control for URLLC over a Non-stationary Wireless Channel using Contextual Reinforcement Learning

Mohit K. Sharma, *Member, IEEE*, Sumei Sun, *Fellow, IEEE*,
Ernest Kurniawan, *Member, IEEE*, and Peng Hui Tan, *Member, IEEE*

***Abstract*—In this work we investigate the design of energy-optimal policies for ultra-reliable low-latency communications (URLLC) over a non-stationary wireless channel, using a contextual reinforcement learning (RL) framework. We consider a point-to-point communication system over a *piece-wise stationary* wireless channel where the Doppler frequency of the channel switches between two distinct values, depending on the underlying state of the channel. To benchmark the performance, first we consider an *oracle agent* which has a perfect but causal information about the switching instants, and consists of two deep RL (DRL) agents each of which is tasked with optimal decision making in a unique partially stationary environment. Comparing the performance of the oracle agent with the conventional DRL reveals that the performance gain obtained using oracle agent depends on the dynamics of the non-stationary channel. In particular, for a non-stationary channel with *faster switching rate* the oracle agent results in approximately $15-20\%$ less energy consumption. In contrast, for a channel with *slower switching rate* the performance of the oracle agent is similar to the conventional DRL agent. Next, for a more realistic scenario when the information about the switching instants for the Doppler frequency of the underlying channel is not available, we model the non-stationary channel as a regime switching process modulated by a Markov process, and adapt the oracle agent by aiding a state tracking algorithm proposed for the regime switching process. Our simulation results show that the proposed algorithm yields a better performance compared to the conventional DRL agent.**

***Index Terms*—Energy minimization, non-stationary wireless channel, reinforcement learning, URLLC.**

I. Introduction

Ultra reliable low latency communications (URLLC) [1] supported by 5G and beyond 5G wireless networks will enable several niche applications such as, autonomous connected vehicles [2], industrial automation [3], tactile internet [4], by providing a latency around 1 msec. and packet loss rate in the range $10^{-5}-10^{-9}$. Recent research efforts in this direction have focused upon various resource allocation issues such as, designing a joint scheduler for enhanced mobile broadband (eMBB) and URLLC traffic [5], optimizing energy-latency trade-off [6], joint optimization of uplink and downlink bandwidth configuration for achieving URLLC [7]. Due to their effectiveness in optimizing non-convex and non-deterministic objective functions, the data-driven optimization approaches have also been pursued [8]. In particular, for its model-free learning approach, the reinforcement learning (RL) based schemes have been leveraged for online radio pattern selection [9], dynamic channel allocation [10], co-channel interference management [11], power minimization [12], etc. Most of the above work consider a stationary environment, and adapting these designs directly to a more realistic *non-stationary* operating environment [14], e.g., a non-stationary wireless channel, can be strictly sub-optimal [15].

Indeed, many intended applications of URLLC networks, such as unmanned aerial vehicles (UAV) based communications network [16], require the data to be transmitted over a non-stationary wireless channel [17]. In particular, temporal and spatial correlations in air-to-ground channels for a UAV [18] can change rapidly due to variations in Doppler frequency and scattering environment [19], respectively. Therefore, in the quest to design efficient URLLC network over a non-stationary wireless channel, we investigate the design of optimal power control schemes.

In this work, we consider a point-to-point communication system where randomly arriving input data packets need to be transmitted to a receiver, over a *non-stationary* wireless channel with an *unknown, time-varying Doppler frequency*. A packet is required to be successfully delivered to the receiver within a fixed time-interval from its time of arrival, and a packet which violates its delivery deadline is *dropped*. The incoming data packets are queued at the transmitter, and any given data packet remains in the queue until either it is successfully delivered or dropped. In any given slot, the first packet in the queue is transmitted at a power level determined by the policy adopted by the transmitter. The goal here is to design a scheduler which learns a power control policy that *jointly minimizes the packet drop probability (PDP) as well as long-term average power consumption*. The scheduler attempts to achieve this by adapting the transmit power, based on the information available about the queue length and the state of the wireless channel. The packet arrival process is assumed to follow a stationary but *unknown* distribution.

To learn an optimal policy in an *online* fashion, we undertake a reinforcement learning (RL) based approach. As in [14], the non-stationary wireless channel is approximated by a time-varying sequence of partially stationary channels characterized by their Doppler frequencies, and a unique RL agent is tasked with learning an optimal policy for a given partially stationary channel. Upon detection of a given Doppler frequency, its corresponding RL agent is activated for online learning and decision making. Since an RL agent is known to learn an optimal policy for a given stationary environment, a major

issue here is to quickly and reasonably accurately estimate the Doppler frequency of the underlying channel. An inaccurate or delayed estimation may potentially jeopardize the learning process of all the individual RL agents, and result in a sub-optimal performance. In light of this, our main contributions are the following:

1) Using the framework of contextual RL (CRL), we develop an online, model-free approach for learning a policy to minimize the power consumption for URLLC over a non-stationary channel, when the switching instants for the Doppler frequency are accurately known.
2) For the scenario when the switching instants are unknown, we model the non-stationary channel as a regime switching process modulated by a Markov process and develop an algorithm for detecting changes in the Doppler frequency. Further, by adapting this algorithm with CRL, we develop a scheme for learning an efficient power control policy for this scenario.
3) Through simulations, we benchmark the performance of the proposed algorithms against the conventional deep RL (DRL). Our results show the efficiency of the proposed approaches. For instance, the agent with accurate knowledge about the switching instants achieves approximately $15-20\%$ lower power consumption compared to DRL.

This work is the first to approach the design of a URLLC system over a non-stationary channel. Our results reveal that the performance gain achieved by using a CRL agent in a non-stationary channel (compared to conventional DRL) depends on the underlying dynamics of the channel. In general, our results provide new insights into the design of RL based learning mechanisms for a non-stationary environment.

The organization of the remaining part of the paper is as follows. In Sec. II we present the system model and the problem formulation, in Sec. III we describe the proposed algorithms for learning in non-stationary environment, in Sec. IV we provide simulation results, and conclude in Sec. V.

II. System Model and Problem Formulation

As shown in Fig. 1, we consider a point-to-point communication system where a transmitter needs to successfully deliver fixed-sized packets to a receiver. The time-slot index is denoted by n, where $n \in \mathbb{Z}^+$. At the start of the n^{th} slot, a packet containing B information bits arrives at the transmitter, after packet inter-arrival duration, i_n, assumed to be drawn from a *stationary and unknown* distribution denoted by $f_I(i)$. Each incoming data packet is stored in a queue at the transmitter, to be served in a first-in first-out fashion, and is required to be successfully delivered to the receiver within T slots (fixed) from its time of arrival. For example, a packet arriving in the m^{th} slot needs to be delivered by the $(m+T)^{\text{th}}$ slot, otherwise the packet is *dropped*.

Packets are transmitted over a non-stationary wireless channel which is assumed to vary in a *piece-wise stationary* fashion. In particular, we assume that the Doppler frequency of the wireless channel is time-varying, and takes a value from a finite set of Doppler frequencies denoted by $\mathcal{F}_D = \{f_{D_1}, \cdots, f_{D_L}\}$, where f_{D_i} denotes the i^{th} possible value of Doppler frequency. In the n^{th} slot, the Doppler frequency of the wireless channel, denoted by $f_D(n)$, changes to a new value taken from the set $\mathcal{F}_D$ according to some unknown transition probability, and stays constant for a random but finite number of slots, depending on the underlying dynamics of the channel. For simplicity, we assume that the switch in Doppler frequency occurs only at the start of the transmission of a new packet[1]. In general, we assume that the Doppler frequency switching instants are not known to the transmitter. However, the scheduler can estimate the state of the Doppler frequency based on the information about complex fading channel gain which is assumed to be accurately known to the transmitter.

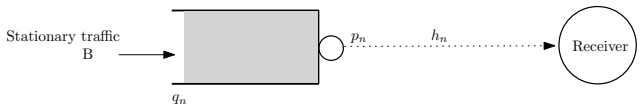

Fig. 1: System model: the data is transmitted using HARQ-IR over a non-stationary channel with unknown distribution. The queue length, the complex fading channel gain of the wireless channel, and the transmit power in the n^{th} slot are denoted by q_n, h_n, and p_n, respectively.

Furthermore, for a given Doppler frequency, the wireless channel is a narrowband block fading channel which remains constant within a slot. The complex fading channel gain in the n^{th} slot is denoted by h_n and distributed according to an unknown stationary distribution $\mathcal{G}$. We note that the non-stationarity only affects the frequency of channel variation, however the average channel gain remains constant across all the partially stationary environments[2].

The head-of-the-line packet of the queue is transmitted according to hybrid automatic repeat request (HARQ) with incremental redundancy (HARQ-IR) protocol [20], [21], until either the packet is successfully delivered or it is dropped. In particular, B information bits are encoded using a mother code of length N, and divided into K sub-codewords of equal length $n_i = \frac{N}{K}$ for $i = 1, \ldots, K$, where K equals the number of maximum transmissions allowed for a given packet. We assume that each sub-codeword requires $\frac{N}{K}$ channel uses for its transmission. We note that even with the stringent latency requirements for URLLC, it is desirable as well as possible to have at least a single re-transmission [6], [22], [23].

The sub-codewords are transmitted progressively, depending on the feedback obtained from the receiver. In particular, the ℓ^{th} sub-codeword is transmitted iff the delivery deadline for the packet has not expired, and the receiver is unable to decode the information using the previous $\ell - 1$ sub-codewords. On the other hand, if a packet is successfully decoded by the receiver then it sends an acknowledgment (ACK) signal, and the packet is removed from the queue at the transmitter. In the next slot, the transmission of the next packet in the queue is started.

Mathematically, the evolution of the total number of packets

[1]This assumption have a negligible impact on performance of our design when applied to a general scenario without this assumption, as in practice the Doppler frequency transitions does not occur very rapidly.

[2]We note that extension to a scenario where the average channel gain also changes along with variations in the Doppler frequency is straightforward.

in the queue which remain to be served can be expressed as follows:

$$q_{n+1} = \max\{q_n + \mathbb{1}_{\{d_n \neq 0\}} - \mathbb{1}_{\{\text{ACK}\}} - \mathbb{1}_D(n), 0\}, \quad (1)$$

where q_n denotes the total number of packets in the queue at the start of the n^{th} slot, and $\mathbb{1}_{\{d_n \neq 0\}}$, $\mathbb{1}_{\{\text{ACK}\}}$, and $\mathbb{1}_D(n)$ are indicator variables. The variable $\mathbb{1}_{\{d_n \neq 0\}}$ is equal to one if and only if there is packet arrival in the n^{th} slot, otherwise it takes value equal to zero, while the variable $\mathbb{1}_{\{\text{ACK}\}}$ equals one only when an ACK is sent by the receiver and remains zero otherwise. Further, $\mathbb{1}_D(n)$ takes value one if the deadline for the head-of-the-line (HOL) packet has expired in the n^{th} slot; otherwise it takes value zero. Note that, (1) is written for the scenario when the queue lengths can be infinite. However, (1) and subsequent development in this paper can be easily modified to account for a finite length queue.

In order to compute the PDP we consider two scenarios termed as slow fading and fast fading channel where, depending on the time-varying Doppler frequency, all the sub-codewords of a data packet being transmitted using HARQ-IR encounter the same and different channel realizations, respectively. Using the normal approximation, the outage probability for the ℓ^{th} transmission of HARQ-IR over a *slow* fading channel can be approximated as [6, Eq. 1]

$$\epsilon_\ell = Q\left(\frac{\sum_{i=1}^{\ell} n_i \ln(1+P_i) - B\ln 2}{\sqrt{\sum_{i=1}^{\ell} \frac{n_i P_i (P_i+2)}{(P_i+1)^2}}}\right), \quad (2)$$

where $Q(\cdot)$ is the complementary Gaussian cumulative distribution and P_i denotes the power used for the transmission of the i^{th} sub-codeword. On the other hand, the outage probability over a *fast* fading channel can be written as [21, Eq. 9]

$$\epsilon_\ell = Q\left(\frac{\sum_{i=1}^{\ell} n_i \log_2(1+P_i h_i) + \frac{\log(\ell n_i)}{n_i} - B}{\sqrt{\sum_{i=1}^{\ell} \frac{n_i P_i (P_i+2)}{(P_i+1)^2}}}\right), \quad (3)$$

here h_i denotes the complex fading channel gain during the i^{th} transmission.

The aim of this work is to design a mechanism to learn a policy which jointly minimizes the PDP and transmit power. In a given slot, a policy, $\mathbf{\Pi}$, needs to determine the optimal transmit power P_ℓ such that the PDP is minimized, taking into account the information available about the channel state and the queue length. The above objective can be written as

$$\min_{\mathbf{\Pi}} \limsup_{T\to\infty} \frac{1}{T} \sum_{n=1}^{T} \left(\mathbb{1}_D(n) + \gamma P_n\right), \quad (4)$$

where $\gamma \in [0,1]$ is a constant. In the n^{th} slot, the *system state* comprises of the queue length, q_n, and the state of the wireless channels, h_n. We note that although the channel state, h_n, is known, the information about the underlying state of the non-stationary channel, i.e., Doppler frequency, is not known. Also, the distributions of the packet inter-arrival times and the wireless channels are not available. Therefore, to learn an optimal online policy we develop a DRL-based approach for learning in a non-stationary environment. In the next section we describe our scheme.

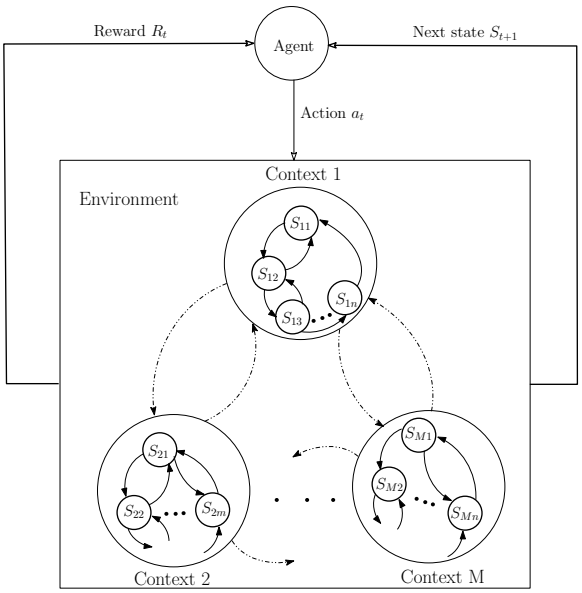

Fig. 2: Framework for learning in a non-stationary environment.

III. Reinforcement Learning in a Non-stationary Environment

In this section, we first describe our framework for learning in a non-stationary environment, and present the design of an oracle agent for a scenario when the Doppler frequency switching instants are accurately known to the scheduler. We then model the non-stationary channel as a regime switching process, and present a stochastic approximation based algorithm for detecting changes in the Doppler frequency. Using this algorithm we adapt the oracle agent to develop a scheme for learning in a general scenario when the switching instants for the Doppler frequency of the channel are not known.

To develop an RL based scheme for learning in a non-stationary environment, we approximate it as a sequence of *partially stationary environments* termed as *contexts*, as shown in Fig. 2. For a given context, an optimal policy for decision making in that context can be successfully learned by using a DRL agent dedicated for learning in that context only. In RL literature, this is known as *contextual RL* (CRL). In the CRL, for a given context its corresponding RL agent is activated for learning and training in the context. As soon as the context is switched the currently active RL agent is changed to the RL agent corresponding to new context. Therefore, given the efficiency of RL in learning in a stationary environment, designing an efficient CRL agent boils down to developing an algorithm for accurately tracking the underlying context. We note that the inaccuracy in detection of contexts affect the performance of an individual RL agent, and in turn of CRL agent, in two ways, i.e., this leads to an RL agent being active in a context different from its own, which not only leads to sub-optimal decision making but also have a negative impact on learning, due to training using inappropriate data.

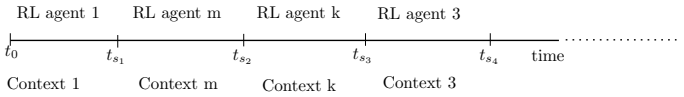

Fig. 3: Oracle CRL agent: RL agent 1 is activated in Context 1, RL agent m is activated in Context m, and so on.

In a scenario when the Doppler frequency, or context, switching instants are known to the scheduler with perfect accuracy, we call this type of scheduler as *oracle agent*. The oracle agent is simply a collection of RL agents with each agent corresponding to a distinct context, with total number of RL agents equal to the number of contexts in the underlying non-stationary environment. As shown in Fig. 3, whenever a context is switched the corresponding RL agent for that context is immediately activated. Since there is no tracking error introduced, any given RL agent trains and makes decision in its corresponding context (which is stationary). As such, following the optimality of RL in stationary environments, it can be easily shown that CRL agent with perfect knowledge achieves optimal performance over a non-stationary channel. In the following we develop an algorithm for detecting the Doppler frequency change for a non-stationary channel.

A. Context Tracking Using Stochastic Approximation Algorithm for Regime Switching Processes

In this section, we model the problem of tracking the Doppler frequency of the wireless channel as a regime switching process modulated by a discrete-time Markov chain (DTMC) [24]. In particular, we model the Doppler frequency transitions as a DTMC with state space equal to $\mathcal{F}_D$, and the sequence of channel gains, h_n, is modeled as another faster varying Markov chain, conditioned on the slower varying DTMC of Doppler frequencies. Adapting the results in [24], we develop a constant-step adaptive algorithm for tracking the invariant distribution of the faster Markov chain. In turn, the invariant distribution of the channel gain is used for tracking the Doppler frequency of the underlying channel. In the following we briefly describe our algorithm.

Let $\mathcal{D}_f$ be a discrete-time Markov chain with a state space $\mathcal{F}_D$ and transition probability matrix $\boldsymbol{P}^{\epsilon} = \boldsymbol{I} + \epsilon \boldsymbol{Q}$. Here, $\epsilon > 0$ is a small parameter, $\boldsymbol{I}$ is an $L \times L$ identity matrix, and $\boldsymbol{Q} = (q_{ij}) \in \mathbb{R}^{L \times L}$ is a generator of continuous-time Markov chain, i.e., $q_{ij} > 0$ for $i \neq j$, and $\sum_{j=1}^{L} q_{ij} = 0$ for each $i = 1, \ldots, L$. Furthermore, $\boldsymbol{Q}$ is irreducible. Also, let $\mathcal{H}$ be an S-state conditional Markov chain, conditioned on the Doppler frequency Markov chain, with state space $\mathcal{S} \triangleq \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_S\}$ where, for $i = 1, \ldots, S$, $\boldsymbol{e}_i$ denotes the i^{th} standard unit vector with i^{th} component being equal to one and the only non-zero component. For each $f_D \in \mathcal{F}_D$, $\boldsymbol{A}(f_D) = (a_{ij}(f_D)) \in \mathbb{R}^{S \times S}$, the transition matrix for Markov chain $\mathcal{H}_n$ is defined as

$$\begin{aligned} a_{ij}(f_D) &= P\left(H_{n+1} = e_j | H_n = e_i, f_D^n = f_D\right) \\ &= P\left(H_1 = e_j | H_0 = e_i, f_D^0 = f_D\right), \end{aligned}$$

where $i, j \in \{1, \cdots, S\}$, and f_D^n and f_D^0 denote the state of Markov chain $\mathcal{D}_f$ at time-slot n and zero, respectively. For $f_D \in \mathcal{F}_D$, $\boldsymbol{A}(f_D)$ is irreducible and aperiodic. It ensures that for each $f_D \in \mathcal{F}_D$ there exists unique stationary distribution $\boldsymbol{\pi}(f_D) \in \mathbb{R}^{S \times 1}$ such that $\boldsymbol{\pi}(f_D) = \boldsymbol{\pi}(f_D)\boldsymbol{A}(f_D)$ and $\boldsymbol{\pi}(f_D)\mathbb{1}_S = 1$, where $\mathbb{1}_S$ is an S-length vector with all its entries equal to one.

To track the Doppler frequency, f_{D_n}, we track the invariant distribution $\boldsymbol{\pi}(f_{D_n})$ using a least mean-square type adaptive algorithm with constant step size μ, where $\mu = O(\epsilon)$. In particular, we construct a sequence of estimates $\hat{\boldsymbol{\pi}}_n$ of the time-varying distribution $\boldsymbol{\pi}(f_{D_n})$ as follows

$$\hat{\boldsymbol{\pi}}_{n+1} = \hat{\boldsymbol{\pi}}_n + \mu\left(\boldsymbol{e}_n - \hat{\boldsymbol{\pi}}_n\right), \tag{5}$$

where $\boldsymbol{e}_n$ denotes the state of Markov chain $\mathcal{H}_n$, at the time-slot n. It can be shown that for $\epsilon^2 \ll \mu$, the mean-squared error in the estimate is given as [24, Thm. 3.1]

$$\mathbb{E}\left|\hat{\boldsymbol{\pi}}_n - \mathbb{E}\boldsymbol{\pi}(f_{D_n})\right|^2 = O\left(\mu + \epsilon + \frac{\epsilon^2}{\mu}\right). \tag{6}$$

In the above, $\mathbb{E}\{\cdot\}$ denotes expectation operator. Using the above algorithm, we construct an estimate of the invariant distribution of the channel gain and declare a change in the Doppler frequency of the channel iff

$$||\hat{\boldsymbol{\pi}}_n - \hat{\boldsymbol{\pi}}_m||^2 > \delta, \tag{7}$$

where $\delta > 0$ is a constant. In particular, the change in Doppler frequency is declared when the difference between the current estimate of the invariant distribution $\hat{\boldsymbol{\pi}}_n$ from the estimate of the invariant distribution at the last switching instant m, $\hat{\boldsymbol{\pi}}_m$, is greater than a threshold δ. Using this algorithm with CRL, we have developed a CRL scheme for a scenario when the information about the Doppler frequency switching instants is not known accurately. In the following section, we analyze the performance of the algorithms developed in this paper.

IV. Simulation Results

We consider a UAV moving at a time-varying speed which results in the Doppler frequencies equal to 5 Hz. and 50 Hz. over the carrier frequency of 3 GHz. This corresponds to the channel coherence time values of 50 msec. and 5 msec., respectively. The slot duration is chosen to be equal to 5 msec. The UAV transmits its observations to a ground receiver. The delay deadline for packets equals 50 msec, i.e., $T = 10$, and maximum number of re-transmissions, K, for each packet equal 2. Total blocklength for HARQ-IR codeword, N, equals 300, and each codeword is divided into sub-codewords of length $n_\ell = 150$ for $\ell = 1, 2$. The set of discrete transmit power levels are chosen from the set $\mathcal{P} \triangleq \{0, 5, 10, \cdots, 50\}$ mw. For a unit bandwidth and ambient temperature of 300 K, this corresponds to $\{-40$ dB, 0 dB, 13 dB, $\cdots, 34$ dB$\}$ received signal-to-noise ratio (SNR). We evaluate the performance for two scenarios, namely when the transition probability matrix (TPM) for Doppler frequencies is $\begin{bmatrix} 0.9999 & 0.0001 \\ 0.0001 & 0.9999 \end{bmatrix}$ and $\begin{bmatrix} 0.99 & 0.01 \\ 0.01 & 0.99 \end{bmatrix}$.

TABLE I: Hyper-parameter values: the exploration parameter ϵ gradually decays from $\epsilon_{\max}$ to $\epsilon_{\min}$, with decay equal to ϵ_{decay}.

Parameter	**Value**
Replay buffer size	2×10^5
$\epsilon_{\max}$	1
$\epsilon_{\min}$	0.0001
ϵ_{decay}	0.9999

TABLE II: seed=25, TPM = $\begin{bmatrix} 0.9999 & 0.0001 \\ 0.0001 & 0.9999 \end{bmatrix}$

Rate	PDP		Average Energy	
	DDQN	Oracle	DDQN	Oracle
0.1	1.4×10^{-4}	1.3×10^{-4}	41.46	41.43
0.2	9.33×10^{-5}	1.2×10^{-4}	41.49	41.78
0.3	1×10^{-4}	1.4×10^{-4}	41.52	41.07
0.4	1.4×10^{-4}	1.08×10^{-4}	41.53	41.65
0.5	1.1×10^{-4}	9.78×10^{-4}	41.57	41.83

The value of parameter $\gamma = 0.1$ is used for defining the reward function for simulations.

For CRL, we implement individual RL agents using double deep Q-network based RL (DDQN-RL) [25]. Each DDQN-RL agent is designed using a deep neural network (DNN) with 10 fully connected hidden layers with each consisting of a non-linearity implemented by LeakyRelu with $\alpha = 0.01$, in addition to an input and output layer. In the n^{th} slot, the state of the system consists of h_n, q_n, transmission index of the packet $\in \{1, 2\}$, and the remaining time for the HOL packet. Therefore the dimension of the input layer is four, equal to the dimension of the system state. The dimension of the output layer equals the number of possible actions, i.e., equal to the cardinality of $\mathcal{P}$. The first hidden layer is composed of 24 neurons, and the number of neurons in each successive even-indexed hidden layer remains same as the previous odd-indexed hidden layer, and each successive odd-index hidden layer contains two neurons lesser than the number of neurons in previous odd-indexed hidden layer. The training is performed using the mean-squared-error as the loss function, and the Adam optimizer with batch size 64. Further, the weights of the target Q-network are updated using Polyak's update method. The hyper-parameter values used for implementing DDQN-RL and CRL are summarized in Table I, and are fixed throughout the simulations. However, for each algorithm, we perform an optimal hyper-parameter search for the learning rate and the Polyak's constant, over a finite set, and report the best performance here.

We term the CRL agent with adaptive tracking algorithm as CRL-tracking. We compare the performance of oracle agent and CRL-tracking agent against the conventional DDQN-RL agent. For DDQN-RL agent we compute the performance over 1.5×10^5 packets, while for both the oracle agent and CRL-tracking we compute the performance by averaging over 3×10^5 packets, as they both have two DDQN-RL agents each corresponding to a distinct context. It ensures that all the DNNs are trained using the same number of data points. We note that the conventional DDQN-RL agent contains only a single buffer which stores the data from both the environments. Thus, the DNN of the conventional DDQN-RL agent is trained using the data obtained from the channels corresponding to both the Doppler frequencies. Therefore, a comparison with conventional DDQN-RL agent provides a view on the performance gain obtained by designing specialized algorithms for the non-stationary channel.

TABLE III: seed=15, TPM = $\begin{bmatrix} 0.9999 & 0.0001 \\ 0.0001 & 0.9999 \end{bmatrix}$

Rate	PDP		Average Energy	
	DDQN	Oracle	DDQN	Oracle
0.1	2×10^{-4}	2.13×10^{-4}	26.80	26.82
0.2	2×10^{-4}	2.06×10^{-4}	26.98	26.86
0.3	2.13×10^{-4}	2.13×10^{-4}	26.88	26.90
0.4	2.2×10^{-4}	2.13×10^{-4}	26.93	26.94
0.5	2.3×10^{-4}	2.1×10^{-4}	26.97	26.68

In Tables II and III, we compare the PDP and average energy consumption values computed for DDQN-RL and oracle agent, by varying the rate $\frac{B}{N}$. The performance is compared when the TPM for the Doppler frequency of the non-stationary channel is $\begin{bmatrix} 0.9999 & 0.0001 \\ 0.0001 & 0.9999 \end{bmatrix}$. We observe that the performance gain obtained by oracle agent in comparison of the conventional DDQN-RL is negligible. Thus, it can be concluded that in this scenario the conventional DDQN-RL algorithm is able to learn efficiently over the non-stationary channel. Therefore, we do not compare the performance of CRL-tracking in this scenario.

Next, we consider the scenario when the TPM = $\begin{bmatrix} 0.99 & 0.01 \\ 0.01 & 0.99 \end{bmatrix}$, i.e., the Doppler frequency variations are faster in comparison to the previous scenario. In Tables IV and V, we compare the performance of the conventional DDQN-RL against both the proposed schemes for two different simulation runs with a different random initialization. We observe that in this case the energy consumption of the oracle agent is approximately $15 - 20$ % lesser compared to the DDQN-RL. This illustrates the advantage of using specialized algorithms for the non-stationary channel. Although, the energy consumption of CRL-tracking is only marginally better compared to the DDQN-RL, the PDP performance is significantly better. We note that optimizing the step size μ for CRL-tracking would result in further performance improvement. This will be investigated as a part of future work.

V. Conclusions

In this paper, we developed contextual RL based online schemes for learning an optimal power control policy for URLLC over a non-stationary channel for the scenarios when the switching instants for the Doppler frequency are

TABLE IV: seed=25, TPM = $\begin{bmatrix} 0.99 & 0.01 \\ 0.01 & 0.99 \end{bmatrix}$

Rate	PDP			Average Energy		
	DQN	Oracle	CRL-tracking	DQN	Oracle	CRL-tracking
0.1	1.06×10^{-4}	1.2×10^{-4}	7.66×10^{-5}	41.45	33.84	40.83
0.2	1.26×10^{-4}	1.1×10^{-4}	7×10^{-5}	41.49	34.53	40.883
0.3	1×10^{-4}	9.67×10^{-5}	8.33×10^{-5}	41.53	34.32	40.899
0.4	1.5×10^{-4}	1.3×10^{-4}	9.66×10^{-5}	41.53	34.32	40.915
0.5	1.1×10^{-4}	2.46×10^{-4}	8.33×10^{-5}	41.35	35.42	40.944

TABLE V: seed=15, TPM = $\begin{bmatrix} 0.99 & 0.01 \\ 0.01 & 0.99 \end{bmatrix}$

Rate	PDP			Average Energy		
	DQN	Oracle	CRL-tracking	DQN	Oracle	CRL-tracking
0.1	1.34×10^{-4}	1.4×10^{-4}	9.67×10^{-5}	41.77	26.84	40.9
0.2	1.5×10^{-4}	1.03×10^{-4}	7×10^{-5}	41.81	34.53	40.93
0.3	1.13×10^{-4}	1.6×10^{-5}	7.67×10^{-5}	41.53	34.32	41.03
0.4	1.67×10^{-4}	1.7×10^{-4}	8.23×10^{-5}	41.55	34.67	40.35
0.5	1.13×10^{-4}	1.7×10^{-4}	5.1×10^{-4}	41.57	34.46	39.03

known perfectly as well as imperfectly. Through simulations, we benchmarked the performance of the developed schemes against the conventional deep RL algorithms. Our results provide several new insights in the design of contextual RL based schemes for a non-stationary environment.

REFERENCES

[1] 3GPP, "Study on physical layer enhancements for NR ultra-reliable and low latency case (URLLC) (release 16)," *TR 38.824 V16.0.0*, 2019.

[2] S. R. Pokhrel, N. Kumar, and A. Walid, "Towards ultra reliable low latency multipath TCP for connected autonomous vehicles," *IEEE Trans. Veh. Technol.*, vol. 70, no. 8, pp. 8175–8185, 2021.

[3] M. Khoshnevisan, V. Joseph, P. Gupta, F. Meshkati, R. Prakash, and P. Tinnakornsrisuphap, "5G industrial networks with comp for URLLC and time sensitive network architecture," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 4, pp. 947–959, 2019.

[4] S. K. Sharma, I. Woungang, A. Anpalagan, and S. Chatzinotas, "Toward tactile internet in beyond 5G era: Recent advances, current issues, and future directions," *IEEE Access*, vol. 8, pp. 56 948–56 991, 2020.

[5] A. Anand, G. de Veciana, and S. Shakkottai, "Joint scheduling of URLLC and eMBB traffic in 5G wireless networks," *IEEE/ACM Trans. Netw.*, vol. 28, no. 2, pp. 477–490, 2020.

[6] A. Avranas, M. Kountouris, and P. Ciblat, "Energy-latency tradeoff in ultra-reliable low-latency communication with retransmissions," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 11, pp. 2475–2485, 2018.

[7] C. She, C. Yang, and T. Q. S. Quek, "Joint uplink and downlink resource configuration for ultra-reliable and low-latency communications," *IEEE Trans. Commun.*, vol. 66, no. 5, pp. 2266–2280, 2018.

[8] C. She, C. Sun, Z. Gu, Y. Li, C. Yang, H. V. Poor, and B. Vucetic, "A tutorial on ultrareliable and low-latency communications in 6G: Integrating domain knowledge into deep learning," *Proc. IEEE*, vol. 109, no. 3, pp. 204–246, 2021.

[9] A. A. Esswie, K. I. Pedersen, and P. E. Mogensen, "Online radio pattern optimization based on dual reinforcement-learning approach for 5G urllc networks," *IEEE Access*, vol. 8, pp. 132 922–132 936, 2020.

[10] N. B. Khalifa, M. Assaad, and M. Debbah, "Risk-sensitive reinforcement learning for URLLC traffic in wireless networks," in *Proc. WCNC*, 2019, pp. 1–7.

[11] B. Khodapanah, T. Hößler, B. Yuncu, A. N. Barreto, M. Simsek, and G. Fettweis, "Coexistence management for URLLC in campus networks via deep reinforcement learning," in *Proc. WCNC*, 2020, pp. 1–6.

[12] A. T. Z. Kasgari, W. Saad, M. Mozaffari, and H. V. Poor, "Experienced deep reinforcement learning with generative adversarial networks (GANs) for model-free ultra reliable low latency communication," *IEEE Trans. Commun.*, vol. 69, no. 2, pp. 884–899, 2021.

[13] Q. Huang, X. Xie, and M. Cheriet, "Reinforcement learning-based hybrid spectrum resource allocation scheme for the high load of URLLC services," *EURASIP J. Wireless Commun. Networks*, no. 250, pp. 1–21, 2020.

[14] S. Padakandla, K. J. Prabhuchandran, and S. Bhatnagar, "Reinforcement learning algorithm for non-stationary environments," *Springer J. on Applied Intell.*, no. 50, p. 3590–3606, 2020.

[15] N. Mehrnia and S. Coleri, "Non-stationary wireless channel modeling approach based on extreme value theory for ultra-reliable communications," *IEEE Trans. Veh. Technol.*, vol. 70, no. 8, pp. 8264–8268, 2021.

[16] C. She, C. Liu, T. Q. S. Quek, C. Yang, and Y. Li, "Ultra-reliable and low-latency communications in unmanned aerial vehicle communication systems," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3768–3781, 2019.

[17] C.-X. Wang, J. Huang, H. Wang, X. Gao, X. You, and Y. Hao, "6G wireless channel measurements and models: Trends and challenges," *IEEE Veh. Technol. Mag.*, vol. 15, no. 4, pp. 22–32, 2020.

[18] D. W. Matolak and R. Sun, "Air–ground channel characterization for unmanned aircraft systems—part iii: The suburban and near-urban environments," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 6607–6618, Aug. 2017.

[19] R. M. Rao, V. Marojevic, and J. H. Reed, "Adaptive pilot patterns for CA-OFDM systems in nonstationary wireless channels," *IEEE Trans. Veh. Technol.*, vol. 67, no. 2, pp. 1231–1244, 2018.

[20] B. Makki, T. Svensson, and M. Zorzi, "Finite block-length analysis of the incremental redundancy HARQ," *IEEE Wireless Commun. Lett.*, vol. 3, no. 5, pp. 529–532, 2014.

[21] Y. Li, M. C. Gursoy, and S. Velipasalar, "Throughput of HARQ-IR with finite blocklength codes and qos constraints," in *Proc. IEEE Int. Symp. Inf. Theory*, 2017, pp. 276–280.

[22] A. Anand and G. de Veciana, "Resource allocation and HARQ optimization for urllc traffic in 5G wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 11, pp. 2411–2421, 2018.

[23] R. Kotaba, C. N. Manchón, T. Balercia, and P. Popovski, "How URLLC can benefit from NOMA-based retransmissions," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1684–1699, 2021.

[24] G. Yin, V. Krishnamurthy, and C. Ion, "Regime switching stochastic approximation algorithms with application to adaptive discrete stochastic optimization," *SIAM J. Optim.*, vol. 14, no. 4, pp. 1187–1215, 2014.

[25] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," *arXiv:1509.06461v3 [cs.LG]*, 2015.