

Interference Prediction for Low-Complexity Link Adaptation in Beyond 5G Ultra-Reliable Low-Latency Communications

Alessandro Brighente*, Jafar Mohammadi†,
 Paolo Baracca†, Silvio Mandelli†, and Stefano Tomasin*
 *Department of Information Engineering, University of Padova, Italy
 †Nokia Bell Labs, Nokia, Germany

Abstract—Traditional link adaptation (LA) schemes in cellular network must be revised for networks beyond the fifth generation (b5G), to guarantee the strict latency and reliability requirements advocated by ultra reliable low latency communications (URLLC). In particular, a poor error rate prediction potentially increases retransmissions, which in turn increase latency and reduce reliability. In this paper, we present an interference prediction method to enhance LA for URLLC. To develop our prediction method, we propose a kernel based probability density estimation algorithm, and provide an in depth analysis of its statistical performance. We also provide a low complexity version, suitable for practical scenarios. The proposed scheme is compared with state-of-the-art LA solutions over fully compliant 3rd generation partnership project (3GPP) calibrated channels, showing the validity of our proposal.

Index Terms—Beyond 5G, kernel distribution estimation, statistical link adaptation, ultra-reliable low-latency communications.

I. INTRODUCTION

Among the different application scenarios for networks beyond the fifth generation (b5G), ultra reliable low latency communications (URLLC) have drawn significant attention from both industrial and academic research. The strict requirements on latency (1 – 10 ms) and reliability (first transmission block error rate (FT-BER) $< 10^{-5}$) will enable new use cases, such as factory automation [1], autonomous vehicles [2], and tactile Internet [3]. Even more stringent requirements with latency down to 100 μ s are expected to be supported by the sixth generation (6G) [4]. In order to meet the aforementioned requirements, different communication solutions have been investigated. In fact, short packets [5], shorter transmission time intervals, and grant-free access schemes [6] are enablers of URLLC. The reader is referred to [7] for an overview of available solutions.

In this paper, we focus on link adaptation (LA), i.e., the choice of a proper modulation and coding scheme such that a certain FT-BER is met. LA drastically reduces the number of required retransmissions, as the choice of the modulation and coding scheme (MCS) is based on the channel quality at transmission time, therefore adapting transmissions to the actual channel quality. However, in order to meet very low target FT-BER, a poor LA may yield too conservative transmission rates, with waste of resources. Instead, the LA algorithm should

guarantee the target FT-BER, while at the same time avoiding too conservative behaviors.

The baseline LA algorithm is the outer loop link adaptation (OLLA) [8], which exploits the feedback from the receiver about the success of the previous transmissions to adapt the MCS, reaching a target FT-BER. This solution is very efficient in many scenarios, and has been further improved for scenarios with very sporadic traffic [9]. However, OLLA is not well suited for URLLC, mainly because of the latency introduced by the feedback, and several solutions have been proposed to address this issue. In [10] LA is performed based on a filtered version of the signal to interference plus noise ratio (SINR), where the interference power (IP) for the next transmission is predicted by low-pass filtering past IPs. Similarly, in [11], prediction is obtained by two different low-pass filters, designed on the difference between current and previous filtered IP. However, this approach is sensitive to high oscillations of IPs, yielding a sub-optimal LA.

Many research papers reported the benefits of modeling the LA problem as either a reinforcement learning or multi-arm bandit technique. The authors in [12] provide an overview on how machine learning can be applied to meet URLLC requirements; however, the reviewed methodologies do not extract statistical information from the time correlation of the interference to optimize LA. In [13], a reinforcement learning technique is proposed to select the MCS from a discrete set, based on the feedback from the environment, i.e., the ACK/NACK and channel state information (CSI). Multi-arm bandits have been proposed in [14]–[16], in the form of sampling from a posterior distribution of SINRs by different modified Thompson sampling (TS) approaches. The modified TS method [14] further reduces the computational complexity of the conventional TS method. As an improvement, a two-stage TS is proposed in [15]. Lastly, the goal in [16] is to design a *tuning-free* LA scheme by TS. However, none of these works are tailored to URLLC communications. In [17], a LA solution has been proposed to attain ultra-reliability. However, this is obtained by means or retransmission, which increase the overall latency. When delay constraints allow for retransmissions, proper schemes can achieve the desired reliability with higher spectral efficiency [18]. When delay constraints becomes strict, instead, the FT-BER at the first transmission becomes the key enabler for URLLC. In [19] a conservative

LA algorithm has been proposed, where the MCS is chosen on the basis of the estimated strongest channel degradation at the packet transmission time. Although retransmissions may not be needed using a conservative MCS, this solution does not fully exploit channel conditions at transmission time. In [20], a joint LA and retransmission policy is obtained, based on the average SINR value, which however cannot guarantee a small FT-BER in the short term. In [21], LA is implemented based on IPs statistics. In particular, the probability density function (p.d.f.) of IP is predicted using kernel density estimator (KDE) [22], and then used in the LA algorithm.

In this paper, following the approach in [21], we propose an algorithm for the estimation of the IP p.d.f. and its use for LA. To meet the requirements of URLLC, we propose a solution wherein latency is minimized thanks to the absence of retransmissions, while the reliability requirement is met by minimizing the transmission failure probability. Thus, by allowing no retransmission and targeting failure probabilities as low as 10^{-5} , our scheme represents a viable solution for URLLC. The proposed solution is evaluated in a cellular network scenario with channels following either the Rice channel model or a 3rd generation partnership project (3GPP) calibrated 3D Urban micro (UMi) model [23]. It turns out that the proposed LA framework is more accurate and entails a lower complexity than state-of-the-art solutions.

With respect to the literature, and in particular to [21], the contribution of this paper are the following:

- we propose a novel density estimation algorithm, i.e., subsets-based KDE (SB-KDE), and compare its performance with the state-of-the-art KDE;
- we propose a low-complexity version of SB-KDE, i.e., low-complexity SB-KDE (LC-SB);
- we show how density estimation methods can be leveraged for LA, and assess their performance, comparing it with KDE (for p.d.f. prediction), OLLA and a log-normal approximation for LA;
- we test our solution in a 3GPP calibrated 3D UMi scenario, further assessing the validity of the proposed framework.

The rest of the paper is organized as follows. In Section II we introduce the system model for the considered cellular network, and review LA. In Section III we present interference prediction for LA, and review KDE. In Section IV we introduce SB-KDE and its low-complexity version. Section V describes the evaluation framework. The computational complexity analysis of all algorithms is derived in Section VI. In Section VII we assess the validity of the proposed solution, by comparing it with state-of-the-art algorithms in both a Rice channel model and a 3GPP calibrated 3D UMi scenario. Lastly, in Section VIII we draw the conclusions.

II. SYSTEM MODEL

We consider a cellular network with C cells wherein, for each cell, a single next-generation node base (gNB) equipped with N_a antennas serves K_{tot} single-antenna user equipments (UEs). Each cell is populated by a random number of UEs uniformly located in space. We denote as \mathcal{U}_c the set of UEs

indexes in cell c . We assume that each gNB serves a single UE in a resource block, and we denote as $\mathbf{z}(c, t)$ the $\mathbb{C}^{1 \times N_a}$ channel from gNB c toward the UE served at transmission time interval (TTI) t , in the considered resource block.

We consider two different channel models. We first consider the Rice channel model [24, Ch. 2.4.2], where each link is a linear combination of a line of sight (LOS) and non LOS links. While this model is widely used in the literature for its implementation simplicity, we also consider a more realistic spatial 3D UMi 3GPP calibrated [23] scenario.

As in [10], in this work we assume that the gNB perfectly tracks the channel of the scheduled UE, since it changes slowly in the medium-to-low mobility scenarios typical of URLLC. Moreover, the gNB does not know: a) the interference power experienced by the scheduled UE in the actual transmission, and b) the UEs scheduled by the other gNBs and their channels, thus preventing the usage of advanced coordinated multi-point techniques [25]. We assume that downlink transmissions are performed using the maximal ratio transmission (MRT) precoder

$$\mathbf{g}(c, t) = \frac{\mathbf{z}^H(c, t)}{\|\mathbf{z}(c, t)\|}, \quad (1)$$

where $[\cdot]^H$ denotes the Hermitian of a vector.

The signal received by the served UE suffers from the interference caused by all gNBs with index $\ell = 1, \dots, C$, $\ell \neq c$, transmitting toward their scheduled UEs. The SINR measured at UE in cell c at TTI t is given by

$$\rho(c, t) = \frac{|\mathbf{z}(c, t)\mathbf{g}(c, t)|^2 P}{\phi(t) + \sigma^2}, \quad (2)$$

where P is the transmitted power, σ^2 is the noise power, $\phi(t)$ is the IP due to other scheduled users at TTI t , i.e.,

$$\phi(t) = \sum_{\ell=1, \ell \neq c}^C |\mathbf{z}(\ell, t)\mathbf{g}(\ell, t)|^2 P. \quad (3)$$

The system model assumes no specific scheduling policy. Furthermore, the correlation is exploited in the design of our proposed LA scheme. Therefore, different schedulers will impact only in terms of periodicity of the interference patterns. Moreover, because of the strict latency requirements of URLLC traffic, we assume that no retransmission is allowed. Therefore, packets that are successfully received at the UEs always meet the latency constraint, whereas, when transmission fails, the packet is dropped. Finally, we assume that each gNB is fully loaded, i.e., in each TTI there is always a certain UE that needs to be scheduled by each gNB.

In this paper we focus on LA, i.e., on the problem of choosing a proper MCS, subject to a target FT-BER. Whilst for each MCS index the values of modulation order, target code rate, and spectral efficiency are given (see, e.g., [26]), FT-BER values are usually obtained by simulations or data collection. For a given FT-BER, the minimum SINR needed for each MCS is stored in a look-up table. Since now we will focus on the LA of a single gNB, we drop index c from the notation. In particular, considering the set \mathcal{M} of available MCSs and assuming that MCSs are ordered by their increasing rates, the

LA problem for TTI $t + 1$ can be written as

$$M^*(t + 1) = \max_{i \in \mathcal{M}} \{M_i : \hat{\rho}(t + 1) > \rho_i\}, \quad (4)$$

where ρ_i is the minimum SINR for which the target FT-BER is achieved with MCS i , and $\hat{\rho}(t + 1)$ is an estimate of the SINR at $t + 1$. We see that the LA problem selects, among the MCSs which guarantee a certain target FT-BER, the one which also maximizes the rate.

In the baseline OLLA the predicted SINR [8] is obtained only from the last SINR $\rho(t)$ measured at the user's side. The idea is that, if the last transmission was successful with the previously selected MCS and an acknowledgment (ACK) packet is sent back to the receiver, the estimated SINR $\rho_{\text{OLLA}}(t + 1)$ is increased. If instead the previous transmission failed and a non-acknowledgment (NACK) packet is sent back to the transmitter, and then $\rho_{\text{OLLA}}(t + 1)$ is reduced. In particular, the SINR value used for MCS selection at TTI $t + 1$ is

$$\hat{\rho}_{\text{OLLA}}(t + 1) = \rho(t) + \Delta(t + 1), \quad (5)$$

where the offset is computed as

$$\Delta(t + 1) = \begin{cases} \Delta(t) + \Delta_{\text{ACK}}, & \text{if ACK at } t; \\ \Delta(t) + \Delta_{\text{NACK}}, & \text{if NACK at } t. \end{cases} \quad (6)$$

The values $\Delta_{\text{ACK}} > 0$ and Δ_{NACK} are suitably selected, such that the target FT-BER ε is met, with $\Delta_{\text{NACK}} = -\frac{1-\varepsilon}{\varepsilon} \Delta_{\text{ACK}}$ [8]. Therefore, in (5) the SINR estimate is typically reduced in order to have a more conservative approach in LA.

We notice that the basic OLLA design is not suitable for URLLC for two reasons. First, due to the very low FT-BER requirements of URLLC, adjusting the estimated SINR via Δ_{NACK} would lead to a conservatively low MCS. Indeed, to recover from a loss, OLLA needs a number of steps proportional to the inverse of the required reliability, significantly increasing latency. The second drawback of OLLA is that the target FT-BER is guaranteed over a window of duration $1/\varepsilon$, whereas the instantaneous level may be extremely different.

III. INTERFERENCE PREDICTION FOR LA

In this paper, similarly to [21], we compute the predicted SINR $\hat{\rho}(t + 1)$ using the last N_{prev} measured $\rho(t)$, i.e., given the vector

$$\phi_{N_{\text{prev}}}(t) = [\phi(t), \dots, \phi(t - N_{\text{prev}} + 1)]. \quad (7)$$

The underlying assumption is that the SINR and, in turn, the interference power in the previous transmissions are perfectly estimated by the gNB. In a practical system, the two quantities are estimated by the UE on pilots received from the gNB, and their quantized versions are fed back by the UE through the channel quality indicator (CQI). In this respect, note that CQI feedback mechanisms of current cellular systems already provide a precise knowledge of the SINR in previous transmissions [27]. Moreover, it has been shown that the variation of the measured interference power over time with URLLC in a realistic environments are mainly due to issues here properly modeled, as the flashlight effect, with interfering gNBs changing their beamformer to serve different UEs in

different slots [10, Sect. III-D]. Therefore, here, we neglect the effects of quantization in designing the LA technique. However, in Section VII we numerically assess its impact on the proposed technique.

Our aim is to predict the outage SINR, i.e., an SINR value that will be exceeded with very high probability, thus reducing the probability of transmission failures. Thus, we resort to the maximum quantile (MQ) method proposed in [21], so that the SINR prediction reduces to predicting the outage IP $\hat{\phi}(t + 1)$, i.e., the value $\hat{\phi}(t + 1)$ that satisfies

$$\mathbb{P} \left[\hat{\phi}(t + 1) < \phi(t + 1) | \phi_{N_{\text{prev}}}(t) \right] \leq \varepsilon, \quad (8)$$

where ε is the target FT-BER.

In order to compute $\hat{\phi}(t + 1)$, we estimate the conditional p.d.f. of $\phi(t + 1)$, given the observations $\phi_{N_{\text{prev}}}(t)$, i.e., $\hat{f}_{\phi(t+1)|\phi_{N_{\text{prev}}}(t)}(a|\phi)$. Then, (8) becomes

$$\begin{aligned} & \mathbb{P} \left[\hat{\phi}(t + 1) < \phi(t + 1) | \phi_{N_{\text{prev}}}(t) \right] \\ &= \int_0^{\phi(t+1)} f_{\phi(t+1)|\phi_{N_{\text{prev}}}(t)}(a|\phi) da \\ &\approx \int_0^{\phi(t+1)} \hat{f}_{\phi(t+1)|\phi_{N_{\text{prev}}}(t)}(a|\phi) da, \end{aligned} \quad (9)$$

where in the last line we approximated the distribution f with its estimate \hat{f} . The integral in (9) is computed via numerical integration.

Now, we still have the problem of obtaining an estimate of the conditional p.d.f. $\hat{f}_{\phi(t+1)|\phi_{N_{\text{prev}}}(t)}(a|\phi)$. To this end, we consider

$$L \gg N_{\text{prev}} \quad (10)$$

samples of $\phi(t)$, i.e., $\phi(t - 1), \dots, \phi(t - L)$, which are used to estimate the conditional p.d.f.. Then, we decompose the conditional p.d.f. into the ratio of the joint and marginal p.d.f.s $f_{\phi(t+1), \phi_{N_{\text{prev}}}(t)}(a, \phi)$ and $f_{\phi_{N_{\text{prev}}}(t)}(\phi)$, i.e.,

$$f_{\phi(t+1)|\phi_{N_{\text{prev}}}(t)}(a|\phi) = \frac{f_{\phi(t+1), \phi_{N_{\text{prev}}}(t)}(a, \phi)}{f_{\phi_{N_{\text{prev}}}(t)}(\phi)}. \quad (11)$$

In the following we will propose techniques to estimate the joint p.d.f. of multiple random variables, which will be used to estimate both the joint and marginal p.d.f.s in (11). In order to simplify notation, we will consider a generic random vector \mathbf{x} , with p.d.f. $f(\mathbf{x})$. The estimated p.d.f. will be obtained using set $\mathcal{S} = \{\mathbf{s}_1, \dots, \mathbf{s}_N\}$ of observed realizations of \mathbf{x} , where each element \mathbf{s}_n is given by $N_{\text{prev}} + 1$ successive elements $\phi(\cdot)$. Note that in (11) the numerator and denominator of the p.d.f. differ only for the first element in the training sequence. Therefore, the numerator training set elements are computed as $\mathbf{s}_n = [\phi(t - 1 - n), \dots, \phi(t - 1 - n - N_{\text{prev}})]$, whereas the denominator elements are $\mathbf{s}_n = [\phi(t - 2 - n), \dots, \phi(t - 2 - n - N_{\text{prev}})]$. Notice that, since each \mathbf{s}_n is obtained by grouping identically distributed elements, elements in \mathcal{S} are identically distributed. Furthermore, notice that our proposal envisions the offline estimation of the p.d.f., without impact on the latency.

A. p.d.f. Estimation by Cumulative Density Function

The simplest p.d.f. estimator is obtained by means of histogram, i.e.,

$$\hat{f}_h(\mathbf{x}) = \frac{\sum_{n=1}^N \delta(\mathbf{s}_n - \mathbf{x})}{N}, \quad (12)$$

where $\delta(\mathbf{x})$ is the Dirac delta, defined as $\delta(\mathbf{x}) = 0$ if $\mathbf{x} \neq \mathbf{0}$ and such that $\int_{\mathbb{R}^{N_{\text{prev}}+1}} \delta(\mathbf{x}) d\mathbf{x} = 1$.

Let us split the length $N_{\text{prev}} + 1$ vectors \mathbf{x} and \mathbf{s}_n into $\mathbf{x} = [a, \bar{\mathbf{x}}]$ and $\mathbf{s}_n = [b, \bar{\mathbf{s}}_n]$, such that a and b denote the first element, whereas $\bar{\cdot}$ denotes the remaining N_{prev} values. Based on (12), we have the following

Lemma 1. *The empirical conditional cumulative density function (CDF) of a given $\bar{\mathbf{x}}$ is given by*

$$\hat{F}(a|\bar{\mathbf{x}}) = \frac{\sum_{n=1}^N \mathbb{1}(b - a|\bar{\mathbf{x}})}{\sum_{n=1}^N \delta(\bar{\mathbf{s}}_n - \bar{\mathbf{x}})}, \quad (13)$$

where $\mathbb{1}(s_n(1) - x(1)|\bar{\mathbf{x}}) = 1$ if $s_n(1) < x(1)$ given $\bar{\mathbf{s}}_n = \bar{\mathbf{x}}$, 0 otherwise.

Proof. The result is given by substitution of (12) in (11), and by applying the integration in (9). \square

A drawback of this approach is that, when L is not large enough, the entries with lower probability will not appear in \mathcal{S} , and the value of their p.d.f. will be zero. This problem becomes more prominent for URLLC, as targeting 10^{-5} or lower FT-BER requires a precise p.d.f. estimate.

B. KDE

A more accurate p.d.f. estimator is KDE, firstly introduced in [22], namely

$$\hat{f}(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^N K\left(\frac{\mathbf{s}_n - \mathbf{x}}{h}\right), \quad (14)$$

where K is the *kernel function* and h is the *kernel bandwidth*, i.e., a parameter of the kernel function which must be suitably selected.

We consider the multivariate Gaussian kernel with uncorrelated dimensions in [28], i.e.,

$$K\left(\frac{\mathbf{s}_n - \mathbf{x}}{h}\right) = \frac{1}{\sqrt{(2\pi)^D h}} \exp\left(-\frac{\|\mathbf{s}_n - \mathbf{x}\|^2}{2h}\right), \quad (15)$$

where D is the dimension. Bandwidth h can be obtained by minimizing the asymptotic mean integrated squared error (AMISE) [28]. In particular, when considering a Gaussian kernel, the AMISE can be written as [28]

$$\text{AMISE}(h) = \frac{1}{4} \Upsilon(f'') h^4 + \frac{1}{2N\sqrt{\pi}h}, \quad (16)$$

where f'' denotes the second derivative of the true p.d.f. f , and

$$\Upsilon(f'') = \int_{-\infty}^{\infty} [f''(\mathbf{u})]^2 d\mathbf{u}. \quad (17)$$

The minimum AMISE value is obtained for the bandwidth

$$h^* = \left(\frac{1}{2N\sqrt{\pi}\Upsilon(f'')}\right)^{\frac{1}{5}}, \quad (18)$$

which is the standard deviation of the Gaussian kernel. The assumption behind (14) is that values are independent identically distributed. We already discussed the latter condition when creating sets. However, due to the round robin (RR) scheduler, values in \mathcal{S} are correlated. This will be exploited to improve the quality of the proposed estimator, as detailed in Section IV-B.

Notice that, by exploiting Gaussian kernels, a kernel-based estimate of the CDF can be obtained by substituting the Gaussian function with the Gaussian Q function. However, we here focus on p.d.f. estimation.

C. Variable-Bandwidth KDE

The choice of a single bandwidth value may not be accurate enough. For instance, it could be useful to have a larger bandwidth in intervals wherein few samples have been collected, and a smaller bandwidth where many samples has been collected. In order to obtain a better estimate of the IP's p.d.f., we consider the variable bandwidth KDE (VB-KDE), i.e., a KDE with multiple bandwidths. Among VB-KDEs we find two different classes [29]: the *balloon estimator* and the *sample smoothing estimator*.

In the balloon estimator, the bandwidth h is a function of the target point \mathbf{x} , i.e.,

$$\hat{f}(\mathbf{x}) = \frac{1}{Nh(\mathbf{x})} \sum_{n=1}^N K\left(\frac{\mathbf{s}_n - \mathbf{x}}{h(\mathbf{x})}\right). \quad (19)$$

In the sample smoothing estimator, the bandwidth h is a function of the measured sample point \mathbf{s}_n , i.e.

$$\hat{f}(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^N \frac{1}{h(\mathbf{s}_n)} K\left(\frac{\mathbf{s}_n - \mathbf{x}}{h(\mathbf{s}_n)}\right), \quad (20)$$

Both estimators present drawbacks: the balloon estimator has been showed to perform better than the fixed bandwidth KDE only when dealing with a multidimensional p.d.f. with more than three dimensions [29]. On the other hand, the sample smoothing estimator has been showed to be highly dependent on the distance between sample points.

IV. SUBSETS-BASED SAMPLE SMOOTHING ESTIMATOR

In this paper, we focus on the second class of VB-KDE estimators and propose a new method which deals with the drawback of sample smoothing estimators.

Let us consider the sample space (i.e., the list of possible outcomes) \mathcal{X} associated to the p.d.f. $f(\mathbf{x})$. We split \mathcal{X} in B subsets, such that $\mathcal{X}_i \cap \mathcal{X}_j = \emptyset$, $\bigcup_{i=1}^B \mathcal{X}_i = \mathcal{X}$. Subsets are created according to a predefined rule, homogeneous in all dimensions, i.e., denoting as $\mathcal{X}^{(d)} = \{\mathcal{X}_1^{(d)}, \dots, \mathcal{X}_B^{(d)}\}$ the group of subsets created along the d^{th} dimension, \mathcal{X}_i is the i^{th} elements of the cartesian product

$$\mathcal{X}^{(1)} \times \mathcal{X}^{(2)} \times \dots \times \mathcal{X}^{(D)}. \quad (21)$$

The rule deciding how subsets are created may be based on different factors, such as the value assumed by the elements in the set or the number of samples with close values. Both policies will be discussed in Section V-A.

Then, we consider a bandwidth value h_i for each subsets, and define the SB-KDE estimator

$$\tilde{f}(\mathbf{x}) = \frac{1}{B} \sum_{i=1}^B \frac{1}{|\mathcal{X}_i| h_i} \sum_{\mathbf{s} \in \mathcal{X}_i} K\left(\frac{\mathbf{s} - \mathbf{x}}{h_i}\right). \quad (22)$$

Let $\mathbb{P}(\mathcal{X}_i)$ be the probability of the i^{th} subset, and $f(\mathbf{x}|\mathcal{X}_i)$ the p.d.f. of sample \mathbf{x} in the i^{th} subset. Assuming that the value B is given and that sets are created according to a certain policy, we have the following results on the SB-KDE.

Lemma 2. *Given the number of subsets B , the bias of the SB-KDE estimator is given by*

$$\begin{aligned} & \text{Bias}(\tilde{f}(\mathbf{x})) \\ &= \omega_2(K) \sum_{i=1}^B \frac{1}{2} f''(\mathbf{x}|\mathcal{X}_i) h_i^2 \mathbb{P}(\mathcal{X}_i) + o(h_i^2), \end{aligned} \quad (23)$$

where $o()$ is the small o Bachmann–Landau notation, and $\omega_2(K)$ is the second moment of the kernel function K , i.e.,

$$\omega_2(K) = \int_{-\infty}^{\infty} \mathbf{u}^2 K(\mathbf{u}) d\mathbf{u}. \quad (24)$$

Proof. See Appendix A. \square

Lemma 3. *Given the number of subsets B , the variance of the SB-KDE estimator is given by*

$$\text{Var}(\tilde{f}(\mathbf{x})) = \Upsilon(K) f(\mathbf{x}) \frac{1}{B} \sum_{i=1}^B \frac{1}{h_i} + O\left(\frac{1}{N}\right),$$

where N is the total number of samples.

Proof. See Appendix A. \square

Exploiting the results in Lemma 2 and Lemma 3, we also obtain the following two results.

Theorem 4. *For a given number of subsets B , and bandwidths $\mathbf{h} = [h_1, \dots, h_B]$, the AMISE obtained via SB-KDE exploiting a Gaussian kernel is given by*

$$\begin{aligned} & \text{AMISE}(\mathbf{h}) = \\ & \frac{1}{4} \sum_{i=1}^B \left(\Upsilon(f''(\mathbf{x}|\mathcal{X}_i)) h_i^4 \mathbb{P}(\mathcal{X}_i)^2 + \frac{1}{2B^2 |\mathcal{X}_i| \sqrt{\pi} h_i} \right). \end{aligned} \quad (25)$$

Proof. See Appendix A. \square

Theorem 5. *For a given number of subsets B and considering a Gaussian kernel, the minimum AMISE is obtained via the SB-KDE for the i^{th} bandwidth value given by*

$$h_i^* = \left(\frac{1}{2B^2 |\mathcal{X}_i| \sqrt{\pi} \Upsilon(f''(\mathbf{x}|\mathcal{X}_i)) \mathbb{P}(\mathcal{X}_i)^2} \right)^{\frac{1}{5}}. \quad (26)$$

Proof. See Appendix A. \square

A. Optimal Bandwidth Computation

The computation of (26) requires the knowledge of the true p.d.f., as for (18). Therefore, to compute the optimal bandwidth, we resort to an iterative algorithm. The optimal bandwidth for SB-KDE has been proven to depend on both the probability $\mathbb{P}(\mathcal{X}_i)$ and the second derivative of the p.d.f.,

Algorithm 1: Bandwidth computation algorithm for SB-KDE

Data: $\mathcal{X}, B, \mathcal{X}_i \forall i = 1, \dots, B, \iota$
Result: \tilde{f}

- 1 initialize $\tilde{f}_0 = 0$;
- 2 compute $h_i^{(0)}$ (18) via algorithm in [28];
- 3 compute $\hat{f}(\mathbf{x}|\mathcal{X}_i)$ via KDE algorithm in [28];
- 4 compute $\hat{\mathbb{P}}(\mathcal{X}_i)$ via (27) $\forall i = 1, \dots, B$;
- 5 $h_i^{(1)} = h_i^{(0)} (N/(B^2 |\mathcal{X}_i| \mathbb{P}(\mathcal{X}_i)^2))^{1/5}, \forall i = 1, \dots, B$;
- 6 compute \tilde{f}_1 using $h_i^{(1)}$;
- 7 $q = 1$;
- 8 **while** $D_{\text{KL}}(\tilde{f}_q || \tilde{f}_{q-1}) > \epsilon$ **do**
- 9 $q = q + 1$;
- 10 compute $\hat{f}(\mathbf{x}|\mathcal{X}_i)$ via KDE using $h_i^{(q-1)}$;
- 11 compute $\hat{\mathbb{P}}(\mathcal{X}_i)$ via (27) $\forall i = 1, \dots, B$;
- 12 $h_i^{(q)} = h_i^{(q-1)} (N/(B^2 |\mathcal{X}_i| \mathbb{P}(\mathcal{X}_i)^2))^{1/5}, \forall i = 1, \dots, B$;
- 13 compute \tilde{f}_q using $h_i^{(q)}$;
- 14 **end**

$f''(\mathbf{x}|\mathcal{X}_i)$. However, again, the true p.d.f. is not known. Therefore, we propose a heuristic algorithm, splitting the SB-KDE design into the design of one KDE for each subset $i = 1, \dots, B$, obtaining a first estimate of the bandwidths h_i , then refined iteratively.

In particular, we note that (26) and (18) are similar, except for a multiplicative factor given by $(N/(B^2 |\mathcal{X}_i| \mathbb{P}(\mathcal{X}_i)^2))^{1/5}$. Therefore, we exploit the result obtained via KDE to obtain the optimal bandwidth for SB-KDE.

Once the i^{th} bandwidth has been obtained from KDE, a first estimate of the probability $\hat{\mathbb{P}}(\mathcal{X}_i)$ is obtained by integration over the i^{th} subset, i.e.,

$$\hat{\mathbb{P}}(\mathcal{X}_i) = \int_{\mathbf{x} \in \mathcal{X}_i} \hat{f}(\mathbf{x}|\mathcal{X}_i) d\mathbf{x}. \quad (27)$$

The bandwidth values are then updated according to the estimate (27), and the procedure is repeated until convergence. Given the p.d.f. estimates \tilde{f}_q and \tilde{f}_{q-1} respectively at iteration q and $q-1$, the algorithm reaches convergence when the two p.d.f.s are similar in terms of their Kullback-Leibler divergence [30], i.e., when for a suitable parameter ι

$$D_{\text{KL}}(\tilde{f}_q || \tilde{f}_{q-1}) = \sum_{\mathbf{x} \in \mathcal{X}} \tilde{f}_q(\mathbf{x}) \log_2 \frac{\tilde{f}_q(\mathbf{x})}{\tilde{f}_{q-1}(\mathbf{x})} \leq \iota, \quad (28)$$

where $\log_2(\cdot)$ represents the base-2 logarithm.

The algorithm steps are presented in Algorithm 1.

B. Low-complexity SB-KDE

The proposed SB-KDE algorithm has the drawback of a high computational complexity than KDE, as it requires B KDEs for each iteration. In order to reduce its complexity, we replace the bandwidth estimation algorithms with the estimation of a covariance matrix. In particular, since the measured IPs are correlated due to the round robin scheduler, we compute the bandwidth as the correlation matrix of the IP, thus

capturing time correlation. We consider the multidimensional Gaussian kernel,

$$\Xi \left(\frac{\mathbf{s}_n - \mathbf{x}}{\mathbf{H}_i} \right) = \frac{1}{\sqrt{(2\pi)^D \det \mathbf{H}_i}} \exp \left(-\frac{(\mathbf{s}_n - \mathbf{x})^T \mathbf{H}_i^{-1} (\mathbf{s}_n - \mathbf{x})}{2} \right), \quad (29)$$

where $\mathbf{H}_i = \mathbb{E}(\mathbf{s}_n \mathbf{s}_n^H)$ is the covariance matrix of the IP sequence in i^{th} subset and $\det \mathbf{H}_i$ its determinant. For each subset of \mathcal{X} , we estimate the D -dimensional sample covariance matrix. We denote as \mathbf{A}^ℓ the $N_\ell \times D$ matrix whose rows are given by the IP sequence of length D jointly belonging to the ℓ^{th} element of \mathcal{X} . Denoting as μ_d the sample mean value of the d -th column, the (q, j) element of the i^{th} sample covariance matrix is given by

$$[\mathbf{H}_i]_{q,j} = \frac{1}{N_\ell - 1} \sum_{n=1}^{N_\ell} (\mathbf{A}_{n,q}^\ell - \mu_q)^* (\mathbf{A}_{n,j}^\ell - \mu_j), \quad (30)$$

where $\mathbf{A}_{n,q}^\ell$ denotes the element of row n and column q of matrix \mathbf{A}^ℓ . The resulting SB-KDE density estimator is obtained as

$$\tilde{f}(\mathbf{x}) = \frac{1}{B} \sum_{i=1}^B \frac{1}{|\mathcal{X}_i| \det(\mathbf{H}_i)} \sum_{\mathbf{s} \in \mathcal{X}_i} \Xi \left(\frac{\mathbf{s} - \mathbf{x}}{\mathbf{H}_i} \right). \quad (31)$$

By replacing the iterative algorithm with an estimate of the sample covariance matrix, we reduced the computational complexity to a linear function of the number of measured IPs. We henceforth denote the SB-KDE with bandwidth given by (30) as LC-SB.

V. EVALUATION FRAMEWORK

A. Link Adaptation

In this work, we assume that the LA can match the FT-BER target if the predicted IP is larger than that experienced at transmission time. Therefore, we assume a correct transmission if the predicted IP is larger than the actual one at the successive time instant, whereas otherwise a failure occurs. This models a system where uncontrolled retransmissions are to be avoided, for instance due to very strict latency constraints in URLLC. For each IP test sequence of T TTIs, we evaluate the reliability of the different solutions by counting the number of events in which the predicted IP is below the actual one. We hence define the reliability of the system (between 0 and 1) as

$$1 - \theta = 1 - \frac{\sum_{t=1}^{T-1} \chi(t)}{T-1}, \quad (32)$$

where $\chi(t)$ is the indicator function

$$\chi(t) = \begin{cases} 1 & \text{if } \hat{\phi}(t) < \phi(t); \\ 0 & \text{otherwise.} \end{cases} \quad (33)$$

Notice that, θ represents the unreliability. Indeed, θ is the FT-BER obtained when the error correcting code used on the block achieves the channel capacity. Since we are interested in high reliability, we aim at small θ values.

For each UE we assume that, if a failure happens, the experienced data rate (DR) is zero, due to the packet's transmission's failure. Therefore, considering short packets, the instantaneous DR $R(t)$ at TTI t is [31]

$$R(t) \approx (1 - \chi(t)) \left(\log_2(1 + \hat{\rho}(t)) - \sqrt{\frac{V(\hat{\rho}(t))}{M}} Q^{-1}(\varepsilon) \right), \quad (34)$$

where $1 - \chi(t)$ accounts for the transmission success as defined in (33), $V(\hat{\rho}(t)) = \frac{\hat{\rho}(t)(2+\hat{\rho}(t))}{(1+\hat{\rho}(t))^2}$ is the channel dispersion [12, pag. 215], M the blocklength, Q the Gaussian complementary CDF, Q^{-1} its inverse, and $\hat{\rho}(t)$ the predicted SINR. Notice that we used \approx in (34) as we neglect terms with $O(\log M/M)$. We then denote as \bar{R} the average DR in time. Computing the DR as in (34) corresponds to having an infinite set \mathcal{M} of available MCSs, allowing to exactly match the desired ε with the theoretically maximal coding rate possible in additive white Gaussian noise (AWGN) channels. The same approach could be used with a finite set of MCSs, whose $\rho_i, \forall i \in \mathcal{M}$ are known. The impact on performance of practical MCSs, like data channel codes in 3GPP standards [32], is postponed to a future work.

For the MQ-based methods, we consider that prediction exploits the conditional p.d.f. (11), where prediction is based on the previous measured IP, i.e., $N_{\text{prev}} = 1$. We compare two different heuristic policies for creating LC-SB subsets. The first is based on the values assumed by the IPs in the time series. In particular, assuming an IP sequence with minimum value m and maximum value M , the b^{th} subset is given by $\mathcal{X}_d^{(b)} = x_d$, s.t. $x_d \geq (b-1)\frac{M-m}{B} + 1$ and $x_d < b\frac{M-m}{B}$. We denote this method by LC-SBV. The latter policy is based on the number of different IPs in the time series. In particular, we populate each subset with the same number of sample points. We denote this method as LC-SBN.

B. Baseline Algorithms

In order to better assess the performance of the SB-KDE and its low-complexity version, we compare the results with four state-of-the-art algorithms. Two baselines are given by the MQ method proposed in [21], based on the empirical CDF (13) (henceforth referred to as ECDF) and on KDE. The third method is the based on OLLA, described in Section II. Since OLLA does not exploit any interference prediction method, for a fair comparison, we exploit the low-pass filtering of the IPs proposed in [10]. We will hereafter denote this method as OLLA-LPP. In particular, given the previously predicted IP $\hat{\phi}(t)$ and the current measured IP $\phi(t)$, the IP to be used for the next transmission is given by [10]

$$\hat{\phi}(t+1) = \alpha \phi(t) + (1 - \alpha) \hat{\phi}(t-1), \quad (35)$$

where α is a constant real value, which is usually small. The predicted value is then used to compute the SINR $\rho_{\text{OLLA}}^{(t+1)}$. The last baseline method assumes the IPs' distribution to be log-normal. Its first and second (order) moments are estimated as a log-normal random variable [33]. We propose to set the first and second moments of the p.d.f. as the mean $\mu_{\mathcal{X}}$ and

variance $\sigma_{\mathcal{X}}^2$ of the training set given by \mathcal{X} . The predicted IP is hence given by

$$\hat{\phi}(t+1) = \exp\left(\Theta^{-1}(2\varepsilon - 1)\sqrt{2}\sigma_{\mathcal{X}} + \mu_{\mathcal{X}}\right), \quad (36)$$

where $\Theta(x)$ represents the error function

$$\Theta(x) = \frac{1}{\sqrt{\pi}} \int_{-x}^x e^{-t^2} dt. \quad (37)$$

VI. COMPUTATIONAL COMPLEXITY

We here consider the computational complexity in terms of total number of addition and multiplication operations. The implementation of the KDE and of its optimal bandwidth in [28] requires the computation of the discrete cosine transform (DCT) of the data in the training set. The computational complexity of a two-dimensional DCT is given by [34]

$$\mathcal{C}_{\text{DCT}}(\mu) = \frac{7}{3}\mu\beta - \frac{20}{9}\mu + \frac{2}{9}(-1)^\beta + 2, \quad (38)$$

where μ is the length of the DCT and $\beta = \log_2 \mu$. Successively, it requires to find the root of the obtained function, with a total number of operations given by [35, Th. 2.1]

$$\mathcal{C}_{\text{root}} = 2 \log_2 \frac{\eta_0}{\eta}, \quad (39)$$

where η_0 is the difference between the maximum and minimum value in the training set, and η is the error tolerance. The overall complexity is hence given by

$$\mathcal{C}_{\text{KDE}}(\mu) = \mathcal{C}_{\text{DCT}}(\mu) + \mathcal{C}_{\text{root}}. \quad (40)$$

The SB-KDE algorithm requires an initial KDE estimate for each subset. Then, it requires, for each iteration until convergence, the computation of a summation over the number of samples in each subset and 4 multiplications for each subset. The overall complexity is hence given by

$$\mathcal{C}_{\text{SB-KDE}} = B\mathcal{C}_{\text{KDE}}(\mu/B) + n_{\text{it}}4B \sum_{i=1}^B |\mathcal{X}_i|, \quad (41)$$

where n_{it} denotes the number of iterations for convergence.

The LC-SB requires the estimation of the set variance for each subset, which is linear in the cardinality of the considered subset. Therefore, the overall complexity can be expressed as

$$\mathcal{C}_{\text{LC-SB}} = \sum_{i=1}^B |\mathcal{X}_i|. \quad (42)$$

VII. NUMERICAL RESULTS

To assess the performance of our proposal, we focus on the most extreme URLLC cases in Industry 4.0 (e.g., motion control) that are characterized by deterministic periodic traffic [36, Sect. 5.2] and assume that a) packets for a given UE appear at periodic TTIs, and b) UEs are served by the gNBs in a deterministic fashion, according to a RR scheduler taking into account each packet latency constraint. This assumption represents a snapshot of industrial control loop operations, where semi-persistent scheduling can be performed in advance due to the periodic nature of UE traffic [37]. The use of an

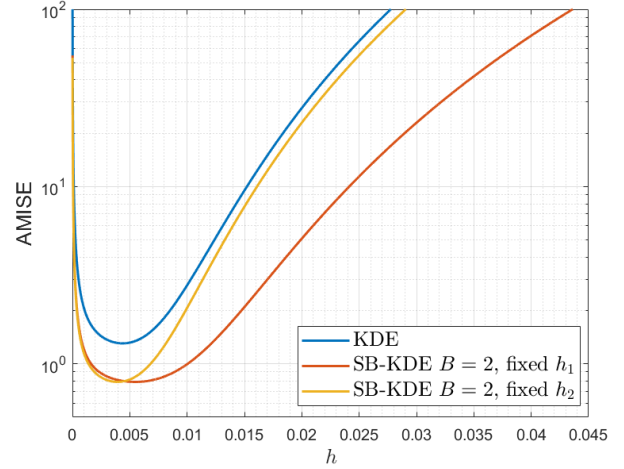


Fig. 1. AMISE vs bandwidth for KDE and SB-KDE. Notice that, being AMISE a function of $B = 2$ bandwidths for SB-KDE, curves are obtained by fixing one bandwidth value and varying the other. We assumed $S_{\text{max}} = 100$, and true p.d.f. with $\mathbf{h} = [1.5, 2.7, 1.2]$.

RR scheduler implies that ρ values are correlated in time, since each user will be periodically affected by the same subset of interferers.

A. SB-KDE Optimality

In order to assess the performance of the SB-KDE density estimator, we compare the AMISE obtained for both KDE and SB-KDE for a fixed and known p.d.f. $f(x)$ of a random variable. In particular, we assume that the true p.d.f. is given by (22) for $B = 3$ and $\mathbf{h} = [1.5, 2.7, 1.2]$. This allows us to compute the p.d.f. $f(x)$ for each x in a predefined dataset \mathcal{S} . We assume that $\mathcal{S} = \{0, 1, \dots, S_{\text{max}}\}$, and we randomly create sets \mathcal{X}_i , $i = 1, \dots, B$, such that $\bigcup_{i=1}^B \mathcal{X}_i = \mathcal{S}$.

Fig. 1 shows the AMISE obtained with KDE as in (16) and SB-KDE as in (25) with $B = 2$. Notice that both estimates are approximations of the true p.d.f., which is based on 3 subsets. Furthermore, since the AMISE for SB-KDE is a function of two bandwidths, results are shown by fixing one bandwidth to the optimal value, and letting the other vary. By using multiple bandwidths, the AMISE is reduced, validating our approach. Furthermore, we notice that the proposed solution is less sensitive to sub-optimal bandwidth values. In fact, we notice that near-to-optimal AMISE values are obtained for a range of bandwidth values wider for SB-KDE than KDE.

B. Rice Channel Model

We here consider a scenario with $N = 9$ square cells, with each gNB located at the center of it at a distance of 200 m from neighboring gNBs. Each gNB is equipped with $N_a = 16$ antennas linearly spaced by $d = \lambda/2$. The IPs are measured in the central cell, and the number of UEs in each surrounding cell is given by the realization of a uniform random variable in range [2, 8]. We consider a noise power of $\sigma^2 = -101$ dBm, a transmitted power at each gNB of $P = 46$ dBm, Rice K factor $\Psi = 10$ dB, path loss exponent $\nu = 3.5$ and a cell edge

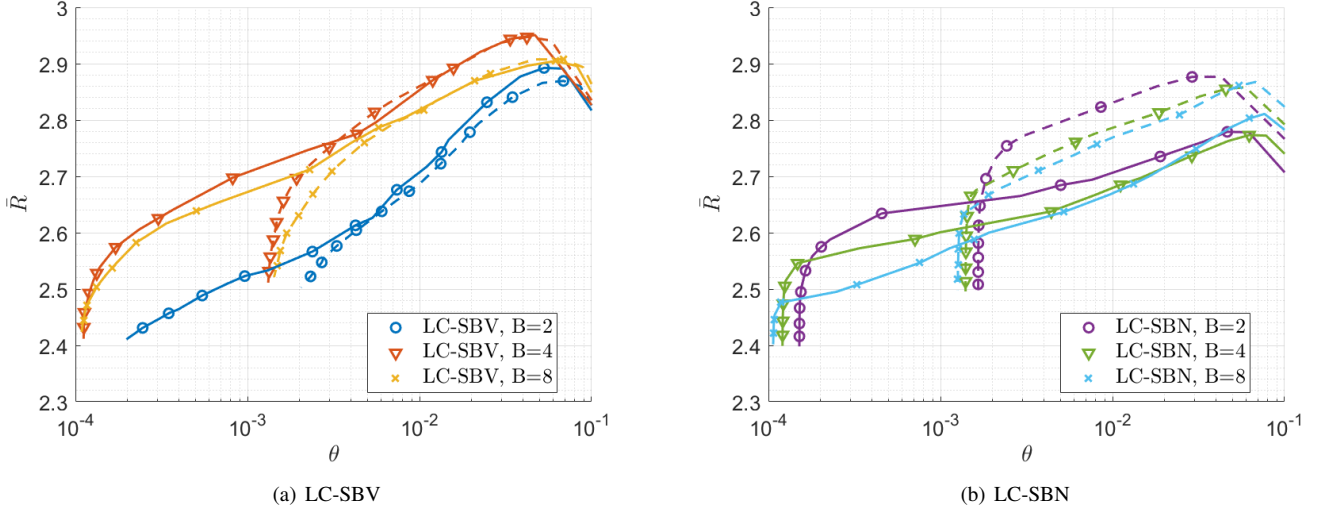


Fig. 2. Average DR vs θ for LC-SBV (left) and LC-SBN (right). In both figures same marker denotes the same number of subsets. Results are reported for a training set of $L = 10^3$ samples (dashed lines) and a training set of $L = 10^4$ samples (solid lines).

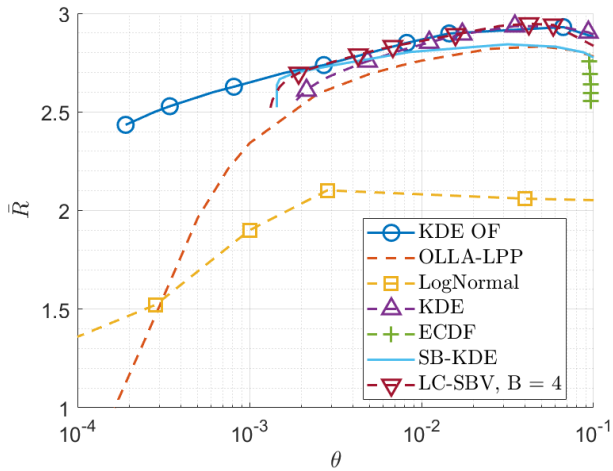


Fig. 3. Average DR vs θ for the best performing policy chosen from Fig. 2 and the baseline methods. Estimation performed with $L = 10^3$ samples.

signal to noise ratio (SNR) without interference of 20 dB. This corresponds to a typical highly interference limited scenario in practical deployments [23].

Fig. 2 shows the average DR vs θ for the LC-SBV (left) and LC-SBN (right). Results have been obtained by fixing the number L of training samples used for p.d.f. estimation and changing the parameter ε , yielding different values of θ . The values of θ in the figures are obtained by simulations, counting the number of outage events over all the predictions via (32). Figures show both the impact of increasing L and the effect of the number of subsets B . About the sensitivity with respect to L , we denote with dashed line the results obtained estimating the p.d.f. with $L = 10^3$ samples and with solid line the results obtained estimating the p.d.f. with $L = 10^4$ samples. Moreover, for each curve in each figure, a higher reliability, i.e., a smaller θ , is obtained by decreasing ε . We notice that a larger L helps in reaching higher reliability values

with both policies, since having a larger number of training samples yields a higher precision in the estimated p.d.f.. This however comes at the cost of a decreased maximum rate. We also notice that for LC-SBN a lower L leads to higher DR for high θ . Indeed, a lower precision in the p.d.f. estimation leads to a less conservative behavior, with higher rates and a lower reliability. We now consider the effect of the number of subsets. We notice that more subsets entail a lower θ and hence a higher reliability for both policies, whereas in terms of rate results do not show the same behavior. In fact, for LC-SBV the best trade-off between DR and reliability is obtained when considering $B = 4$ subsets, whereas for LC-SBN is obtained with $B = 2$ subsets. We can hence identify the number of subsets as a hyper-parameter, to be optimized based on collected data. Comparing the two policies, we notice that LC-SBV attains higher DR as the reliability increases. In the following, we choose LC-SBV $B = 4$ as the policy to be compared with baseline algorithms, as it achieves the best performance both in terms of average DR and reliability.

Fig. 3 shows the average DR vs θ for the best policy chosen from Fig. 2 and the baseline methods, when considering $L = 10^3$ training samples. For KDE we show both the performance obtained with $L = 10^3$ training samples and the optimal results obtained by over-fitting (denoted as KDE OF), i.e., using all samples in the dataset for training. The ECDF method based on (13) attains a poor approximation of the CDF, since not enough points are available to match the strict reliability targets. The comparison with the other methods motivates the need for kernel-based density estimators. About OLLA-LPP we notice that, although it works well with high θ values, the attained DR rapidly degrades for decreasing θ , due to the highly conservative behavior of OLLA which favors reliability to DR. Considering a URLLC scenario targeting high reliability, and hence small θ , we notice that OLLA-LPP is not able to guarantee both high reliability and high DR, therefore not being a suitable solution if the load offered by URLLC becomes relevant. A similar behavior is obtained with

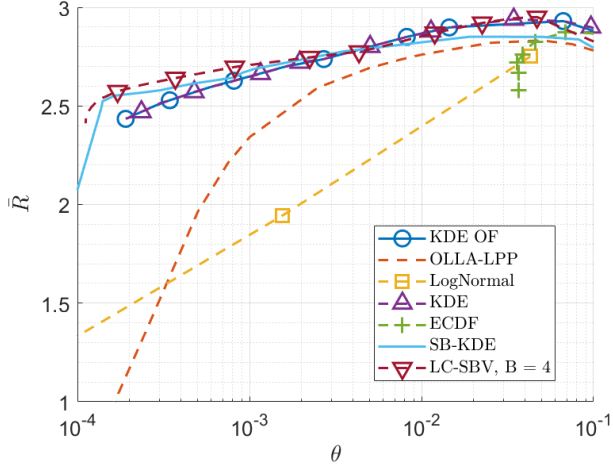


Fig. 4. Average DR vs θ for the best performing policy chosen from Fig. 2 and the baseline methods. Estimation performed with $L = 10^4$ samples.

the LogNormal approximation, where DR rapidly degrades with increasing reliability values. However, differently from OLLA-LPP, the DR has a slower decrease and does not tend to zero for $\theta < 10^{-4}$. Furthermore, we notice that, compared to the other p.d.f.-based methods, LogNormal reaches smaller values of θ . Indeed, this method relies on a closed-form equation of the IP p.d.f. and does not suffer for an insufficient number of training data, being able to reach infinite precision. On the other hand, we notice that the LogNormal approximation attains a higher DR than OLLA-LPP for $\theta < 2 \cdot 10^{-4}$. The DR for the proposed SB-KDE degrades as θ decreases. However, it attains a higher DR when it's able to match the desired reliability when compared to LogNormal and OLLA-LPP, whereas, when compared to KDE, it attains higher DR only for $\theta \leq 5 \cdot 10^{-3}$. Furthermore, compared to KDE, it also attains lower θ , for $\theta \leq 2 \cdot 10^{-3}$, confirming that a subset-based approach is advantageous over the plain KDE. Comparing the low-complexity LC-SBV with all other approaches, we notice that it attains the highest DR for all $\theta \geq 2 \cdot 10^{-3}$. However, LC-SBV is limited by the demanding amount of data needed to estimate the p.d.f..

Fig. 4 shows the average DR vs θ , for the best performing policy chosen from Fig. 2 and the baseline methods, when considering $L = 10^4$ training samples. Note that both KDE and KDE OF have similar performance, as convergence is already obtained with 10^4 training samples. For the ECDF method, a larger training set does not improve its performance. About SB-KDE, we notice a DR degradation smaller than 0.2 bit/s/Hz, up to $\theta = 0.5 \cdot 10^{-4}$, then the DR rapidly decreases, still being higher than that of both OLLA-LPP and the Log-normal approximation. Furthermore, we notice that the subset-based approaches attain smaller θ values than KDE and LC-SBV is the best performing method down to $\theta = 10^{-4}$. Therefore, LC-SBV achieves a higher DR for a given θ with a lower computational complexity with respect to SB-KDE.

C. 3GPP Channel Model

In order to test the proposed algorithm in a more realistic scenario, we consider in this section a three-dimensional spatial 3D Urban Micro (UMi) channel, calibrated with the results obtained by 3GPP [23]. In detail, we consider $C = 21$ cells, organized in 7 sites, each with 120 degrees subsets per cell, inter-site distance of 200 m, and wraparound. An average of 5 UEs is deployed per cell, which move with a speed of 3 km/h. Each gNB is equipped with 64 antennas, organized in a uniform planar array, with 8 rows, 4 columns and with cross-polarized antenna elements. The gNBs serve UEs at a carrier frequency of 3.7 GHz on a system bandwidth of 10 MHz and using a transmit power of 41 dBm. We follow the scenario in [23], where the reader can find more details.

Results have been obtained with $2 \cdot 10^6$ samples of which L are used for estimating the p.d.f. and the others for evaluation. By varying ε we obtain different θ values. As for the Rice channel model, we show both the effect of increasing L and varying the number B of subsets.

Fig. 5 shows the DR vs θ for the two considered policies for the LC-SB algorithm. Dashed lines refer to training sequences of $L = 10^2$ samples, while solid lines to training sequences of 10^4 samples. We notice that, if the training set does not have enough samples, both methods are unable to attain small θ values with a poor reliability. About LC-SBV, we notice the trade-off between DR and reliability when considering the number of subsets B needed to estimate the p.d.f.. In fact, on one hand a larger number of subsets yields higher DR values, on the other hand a smaller number of subsets means we reach smaller θ values. Hence, this parameter will be set according to the specific users' need. About LC-SBN, we instead notice that increasing the number of subsets does not improve performance. However, the LC-SBN with $B = 2$ is the best performing method among all policies and number of subsets, and is hence compared with the baseline methods.

Fig. 6 shows the average DR vs θ for all the baseline methods, the proposed SB-KDE, and LC-SBN with $B = 2$ subsets. Results are obtained with $L = 10^5$ training samples. As for Fig. 4, for both OLLA-LP and the LogNormal approximations the DR increases with θ . About KDE, it converges to its optimal performance already with 10^5 training samples. Although this is the best result obtainable with KDE, it does not represent the overall optimum, as the estimated p.d.f. may be inaccurate. In fact, with ECDF we attain a higher DRs at high θ values. However, KDE can reach $\theta = 10^{-5}$, showing that kernel methods provide good performance at low reliability target. About the proposed SB-KDE and LC-SBN methods, they achieve higher DR values than KDE. Note also that, with $L = 10^5$ training samples, the order of magnitude of attained θ values is not lower than 10^{-5} , while our proposed methods achieve this tight reliability value required by URLLC. Therefore, although LC-SBN has a reduced computational complexity and obtains higher DR values for a given θ , it requires a larger number of training samples to attain smaller θ values.

We have assessed the robustness against quantization errors

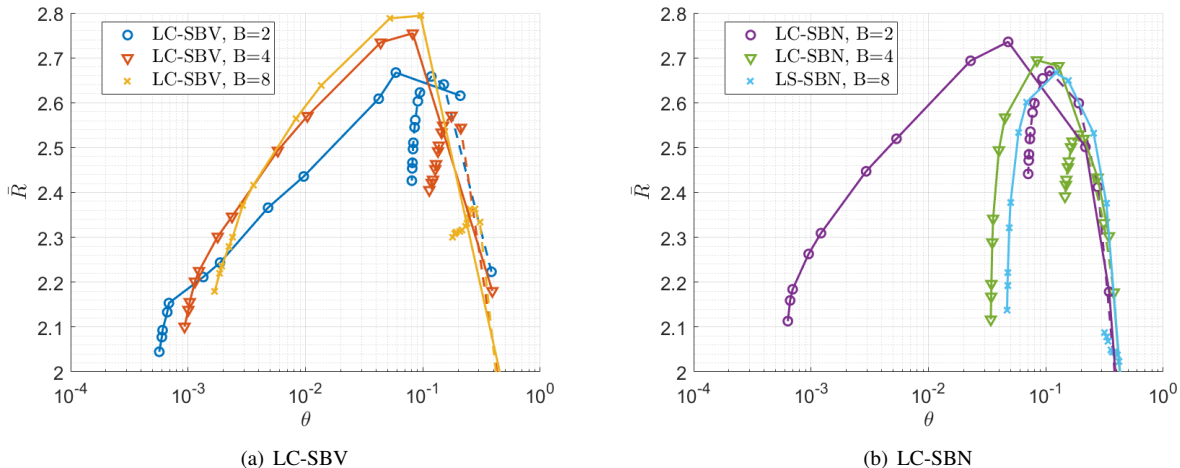


Fig. 5. DR vs θ for LC-SBV (left) and LC-SBN (right) obtained with the 3GPP channel model. In both figures same marker shape denotes the same number of subsets. Results are reported for a training set of $L = 10^2$ samples (dashed lines) and a training set of $L = 10^4$ samples (solid lines).

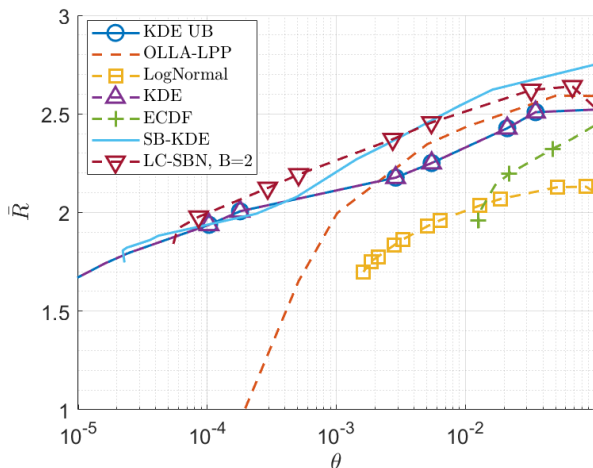


Fig. 6. Average DR vs θ for the best performing policy chosen from Fig. 5 and the baseline methods. Results obtained with $L = 10^5$ training samples.

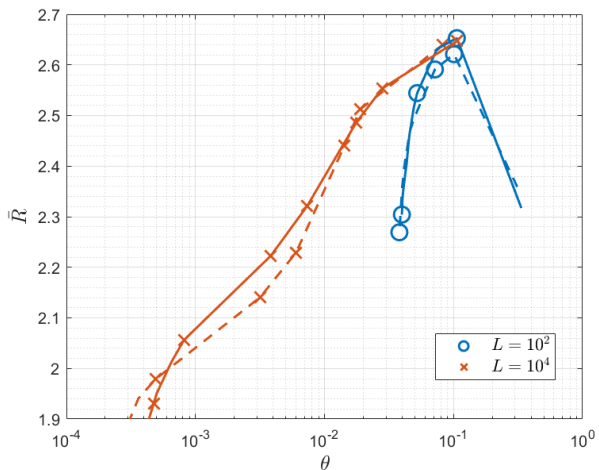


Fig. 7. Average DR vs θ for LC-SBV, $B = 2$. Comparison of the results obtain with quantization error (dashed lines) and without quantization error (solid lines) for $L = 10^2$ and $L = 10^4$.

on the channel estimates of LC-SBV with $B = 2$. In particular, given the CDF of the interference to noise ratio (INR), we designed a 5-bit a non-uniform quantizer where all quantized values are taken with the same probability. Fig. 7 shows the DR vs θ , for training set sizes $L = 10^2$ and $L = 10^4$, with and without the quantization error. We observe that the LC-SBV scheme is robust against quantization errors, showing a negligible impact on the DR. Note also that results considering the quantization error show a higher DR for some extreme values of θ . This is due to the fact that the LC-SBV scheme is sub-optimal, and indeed it shows slightly worse performance at these θ values. Indeed, we conclude that the quantization error unbalances the prediction, making it more conservative, and thus improving the performance.

Fig. 8 shows the computational complexity in terms of number of additions and multiplications vs the number of training samples L for the KDE and the two policies for the low-complexity LC-SB. The best performing method is the LC-SBN, due to the fact that it linearly depends on the number of samples per subset, which decreases as the number of subset increases.

VIII. CONCLUSIONS AND FUTURE WORKS

In this paper we considered the problem of predicting interference power to enable efficient LA for URLLC. We first derived the optimal bandwidth for the SB-KDE, and based on the optimal solution for KDE, we proposed a heuristic algorithm to estimate a p.d.f. based on the optimal bandwidths. Motivated by the considerable computational complexity of the proposed solution, we then proposed a low-complexity version of the SB-KDE, namely the LC-SB. By means of extensive simulations in cellular networks, considering realistic 3GPP 3D UMi channel models, we showed through numerical evaluations that the proposed solutions attain at the same time higher DR and a better matching of the reliability targets than state-of-the-art solutions. By jointly looking at results in Figs. 4, 6 and 8, we can conclude that the proposed LC-SB method attains the best DR with the lower computational complexity

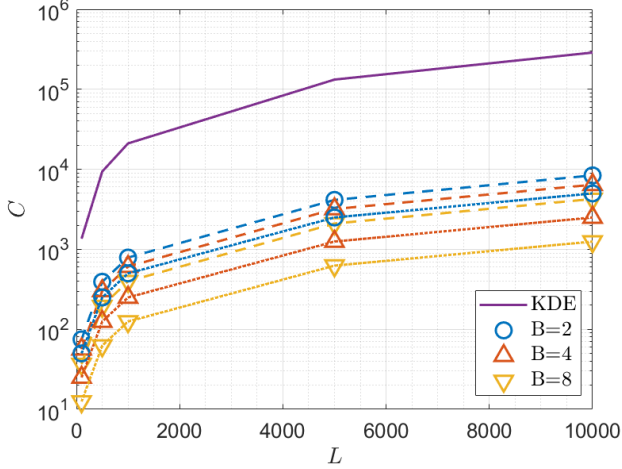


Fig. 8. Computational complexity vs number of training data for the kernel based density estimators. Dashed curves show results for LC-SBV, whereas dotted curves show results for LC-SBN.

and achieves extremely high reliability targets. Therefore, it is an effective algorithm for LA with URLLC in the considered low mobility scenario.

Future study will include potential enhancements targeting URLLC in high mobility scenarios, for instance with an online learning p.d.f. update in order to keep up with the non-ergodic nature of the channel in such scenarios: a simple approach can use a data buffer where parts of the older data, i.e., outdated information, is periodically replaced with more recent ones to keep the p.d.f. up-to-date in a dynamic environment.

APPENDIX A

DERIVATION OF THE OPTIMAL BANDWIDTH FOR SB-KDE

We henceforth consider operations between vector and scalar as element-wise. We also recall that the kernel function of a vector is a scalar value. We assume that $K(\cdot)$ is a kernel function as defined in [22], therefore $\int K(u)du = 1$ and that $\int uK(u)du = 0$.

A. Bias

Define the local KDE as

$$\hat{f}(\mathbf{x}) = \frac{1}{B} \sum_{i=1}^B \frac{1}{|\mathcal{X}_i|} \sum_{\mathbf{z} \in \mathcal{X}_i} \frac{1}{h_i} K\left(\frac{\mathbf{z} - \mathbf{x}}{h_i}\right). \quad (43)$$

The mean value of the kernel function can be computed as

$$E\left[\frac{1}{h_i} K\left(\frac{\mathbf{z} - \mathbf{x}}{h_i}\right)\right] = \int_{-\infty}^{\infty} \frac{1}{h_i} K\left(\frac{\boldsymbol{\zeta} - \mathbf{x}}{h_i}\right) f(\boldsymbol{\zeta}) d\boldsymbol{\zeta}. \quad (44)$$

From the total probability law we have

$$f(\boldsymbol{\zeta}) = \sum_{\ell=1}^B f(\boldsymbol{\zeta}|\mathcal{X}_\ell) \mathbb{P}(\mathcal{X}_\ell), \quad (45)$$

and by substitution in (44) we obtain

$$E\left[\frac{1}{h_i} K\left(\frac{\mathbf{z} - \mathbf{x}}{h_i}\right)\right] = \int_{-\infty}^{\infty} \frac{1}{h_i} K\left(\frac{\boldsymbol{\zeta} - \mathbf{x}}{h_i}\right) \sum_{\ell=1}^B f(\boldsymbol{\zeta}|\mathcal{X}_\ell) \mathbb{P}(\mathcal{X}_\ell) d\boldsymbol{\zeta}. \quad (46)$$

By performing the change of variables $\mathbf{u} = (\mathbf{z} - \mathbf{x})/h_i$ we obtain

$$E\left[\frac{1}{h_i} K\left(\frac{\mathbf{z} - \mathbf{x}}{h_i}\right)\right] = \int_{-\infty}^{\infty} K(\mathbf{u}) \sum_{\ell=1}^B f(\mathbf{u}h_i + \mathbf{x}|\mathcal{X}_\ell) \mathbb{P}(\mathcal{X}_\ell) d\mathbf{u}. \quad (47)$$

Let us consider the second order Taylor series expansion of $f(\mathbf{u}h_i + \mathbf{x}|\mathcal{X}_i)$, obtaining

$$f(\mathbf{u}h_i + \mathbf{x}|\mathcal{X}_i) \approx f(\mathbf{x}|\mathcal{X}_i) + f^{(1)}(\mathbf{x}|\mathcal{X}_i)h_i\mathbf{u} + \frac{1}{2}f''(\mathbf{x}|\mathcal{X}_i)h_i^2\mathbf{u}^2 + o(h_i^2). \quad (48)$$

By substitution of (48) in (47) we obtain

$$\sum_{\ell=1}^B \int_{-\infty}^{\infty} K(\mathbf{u}) f(\mathbf{u}h_i + \mathbf{x}|\mathcal{X}_i) \mathbb{P}(\mathcal{X}_\ell) d\mathbf{u} = \quad (49)$$

$$\sum_{\ell=1}^B \int_{-\infty}^{\infty} K(\mathbf{u}) \left(f(\mathbf{x}|\mathcal{X}_\ell) + f^{(1)}(\mathbf{x}|\mathcal{X}_\ell)h_i\mathbf{u} + \right. \quad (50)$$

$$\left. \frac{1}{2}f''(\mathbf{x}|\mathcal{X}_\ell)h_i^2\mathbf{u}^2 \right) \mathbb{P}(\mathcal{X}_\ell) d\mathbf{u} =$$

$$\sum_{\ell=1}^B \left(f(\mathbf{x}|\mathcal{X}_\ell) + \frac{1}{2}f''(\mathbf{x}|\mathcal{X}_\ell)h_i^2\omega_2(K) \right) \mathbb{P}(\mathcal{X}_\ell),$$

where the second term involving the first derivative is zero as the mean of $K(\mathbf{u})$ is zero and where $\omega_2(K)$ is defined in (24).

Therefore

$$\begin{aligned} E[\hat{f}(\mathbf{x})] &= E\left[\frac{1}{B} \sum_{i=1}^B \frac{1}{|\mathcal{X}_i|} \sum_{\mathbf{z} \in \mathcal{X}_i} \frac{1}{h_i} K\left(\frac{\mathbf{z} - \mathbf{x}}{h_i}\right)\right] \\ &= \frac{1}{B} \sum_{i=1}^B \frac{1}{|\mathcal{X}_i|} \sum_{\mathbf{z} \in \mathcal{X}_i} E\left[\frac{1}{h_i} K\left(\frac{\mathbf{z} - \mathbf{x}}{h_i}\right)\right] \\ &= \frac{1}{B} \sum_{i=1}^B \frac{1}{|\mathcal{X}_i|} \sum_{\mathbf{z} \in \mathcal{X}_i, \ell=1}^B \left(f(\mathbf{x}|\mathcal{X}_\ell) + \frac{1}{2}f''(\mathbf{x}|\mathcal{X}_\ell)h_i^2\omega_2(k) \right) \mathbb{P}(\mathcal{X}_\ell) \\ &= f(\mathbf{x}) + \frac{1}{B} \sum_{i=1}^B \left(\sum_{\ell=1}^B \frac{1}{2}f''(\mathbf{x}|\mathcal{X}_\ell)h_i^2\omega_2(k) \mathbb{P}(\mathcal{X}_\ell) \right) \\ &= f(\mathbf{x}) + \sum_{\ell=1}^B \frac{1}{2}f''(\mathbf{x}|\mathcal{X}_\ell)h_i^2\omega_2(k) \mathbb{P}(\mathcal{X}_\ell). \end{aligned}$$

Therefore, the bias is

$$\begin{aligned} \text{Bias}(\hat{f}(\mathbf{x})) &= E[\hat{f}(\mathbf{x})] - f(\mathbf{x}) \\ &= \omega_2(k) \sum_{\ell=1}^B \frac{1}{2}f''(\mathbf{x}|\mathcal{X}_\ell)h_i^2\mathbb{P}(\mathcal{X}_\ell) + o(h_i^2). \end{aligned} \quad (51)$$

B. Variance

From the bias analysis we saw how the kernel function of a vector can be treated as a random variable. The variance of a random variable (r.v.) x can be computed as

$$\text{Var}(x) = E[x^2] - E[x]^2. \quad (52)$$

From the bias analysis, focusing on the first order Taylor approximation we have

$$E[\hat{f}(\mathbf{x})] \approx f(\mathbf{x}) + o(1), \quad (53)$$

where the second term is $O\left(\frac{1}{N}\right)$, being N the number of samples used for density estimation. Then, following the approach used for the bias computation we have

$$E\left[\frac{1}{h_i}K\left(\frac{\mathbf{z}-\mathbf{x}}{h_i}\right)^2\right] = \quad (54)$$

$$\begin{aligned} & \frac{1}{h_i} \int_{-\infty}^{\infty} K(\mathbf{u})^2 \sum_{\ell=1}^B f(\mathbf{u}h_i + \mathbf{x}|\mathcal{X}_\ell)\mathbb{P}(\mathcal{X}_\ell) d\mathbf{u} \\ &= \int_{-\infty}^{\infty} K(\mathbf{u})^2 f(\mathbf{u}h_i + \mathbf{x}) d\mathbf{u}, \end{aligned} \quad (55)$$

where the second equality comes from the fact that we considered a first order Taylor series and the total probability law. Therefore, recalling that the kernel estimator is a linear estimator and that $K\left(\frac{\mathbf{z}-\mathbf{x}}{h_i}\right)$ is independent identically distributed (i.i.d)

$$\begin{aligned} & \text{Var}(\hat{f}(\mathbf{x})) \\ &= \frac{1}{B^2} \sum_{i=1}^B \frac{1}{|\mathcal{X}_i|^2} \sum_{\mathbf{z} \in \mathcal{X}_i} \frac{1}{h_i} \Upsilon(K) f(\mathbf{x}) + O\left(\frac{1}{N}\right), \\ &= \Upsilon(K) f(\mathbf{x}) \frac{1}{B^2} \sum_{i=1}^B \frac{1}{|\mathcal{X}_i| h_i} + O\left(\frac{1}{N}\right), \end{aligned} \quad (56)$$

where $\Upsilon(K)$ is defined in (17).

C. AMISE

The MSE can be expressed as

$$\text{MSE} = \text{Bias}(\hat{f}(\mathbf{x}))^2 + \text{Var}(\hat{f}(\mathbf{x})). \quad (57)$$

The asymptotic mean squared error (AMSE) is obtained by the asymptotic derivation of bias and variance (51) and (56) as

$$\begin{aligned} \text{AMSE} &= \left(\frac{1}{2} \omega_2(K) \sum_{i=1}^B f''(\mathbf{x}|\mathcal{X}_i) h_i^2 \mathbb{P}(\mathcal{X}_i) \right)^2 \\ &+ \Upsilon(K) f(\mathbf{x}) \frac{1}{B^2} \sum_{i=1}^B \frac{1}{|\mathcal{X}_i| h_i}. \end{aligned}$$

By integrating the AMSE we obtain the AMISE

$$\begin{aligned} & \text{AMISE} \\ &= \int_{-\infty}^{\infty} \left[\left(\frac{1}{2} \omega_2(K) \sum_{i=1}^B f''(\mathbf{x}|\mathcal{X}_i) h_i^2 \mathbb{P}(\mathcal{X}_i) \right)^2 + \right. \\ & \left. \Upsilon(K) f(\mathbf{x}) \frac{1}{B^2} \sum_{i=1}^B \frac{1}{|\mathcal{X}_i| h_i} \right] d\mathbf{x}. \end{aligned} \quad (58)$$

Since $f(\mathbf{x}|\mathcal{X}_i) = 0 \forall \mathbf{x} \notin \mathcal{X}_i$ we obtain

$$\begin{aligned} & \text{AMISE} \\ &= \int_{-\infty}^{\infty} \left(\frac{1}{4} \omega_2(k)^2 \sum_{i=1}^B f''(\mathbf{x}|\mathcal{X}_i)^2 h_i^4 \mathbb{P}(\mathcal{X}_i)^2 \right. \\ & \left. + \Upsilon(K) f(\mathbf{x}) \frac{1}{B^2} \sum_{i=1}^B \frac{1}{|\mathcal{X}_i| h_i} \right) d\mathbf{x} \\ &= \sum_{i=1}^B \left(\frac{1}{4} \Upsilon(f''(\mathbf{x}|\mathcal{X}_i)) h_i^4 \mathbb{P}(\mathcal{X}_i)^2 \omega_2(k)^2 + \frac{\Upsilon(K)}{B^2 |\mathcal{X}_i| h_i} \right). \end{aligned}$$

D. Optimal bandwidth

The optimal value h_i^* can be computed as

$$\frac{\partial \text{AMISE}}{\partial h_i} = \Upsilon(f''(\mathbf{x}|\mathcal{X}_i)) h_i^3 \mathbb{P}(\mathcal{X}_i)^2 \omega_2(K)^2 - \frac{\Upsilon(K)}{B^2 |\mathcal{X}_i| h_i^2} = 0,$$

from which the AMISE optimal bandwidth value is given by

$$h_i^* = \left(\frac{\Upsilon(K)}{B^2 |\mathcal{X}_i| \Upsilon(f''(\mathbf{x}|\mathcal{X}_i)) \mathbb{P}(\mathcal{X}_i)^2 \omega_2(k)^2} \right)^{\frac{1}{5}}. \quad (59)$$

Considering a Gaussian kernel we have

$$\text{AMISE} = \frac{1}{4} \sum_{i=1}^B \left(\Upsilon(f''(\mathbf{x}|\mathcal{X}_i)) h_i^4 \mathbb{P}(\mathcal{X}_i)^2 + \frac{1}{B^2 |\mathcal{X}_i| 2\sqrt{\pi} h_i} \right),$$

and therefore

$$h_i^* = \left(\frac{1}{2B^2 |\mathcal{X}_i| \sqrt{\pi} \Upsilon(f''(\mathbf{x}|\mathcal{X}_i)) \mathbb{P}(\mathcal{X}_i)^2} \right)^{\frac{1}{5}}. \quad (60)$$

REFERENCES

- [1] H. Ren, C. Pan, Y. Deng, M. El-kashlan, and A. Nallanathan, "Joint power and blocklength optimization for URLLC in a factory automation scenario," *IEEE Trans. on Wireless Commun.*, vol. 19, no. 3, pp. 1786–1801, Dec. 2019.
- [2] X. Ge, "Ultra-reliable low-latency communications in autonomous vehicular networks," *IEEE Trans. on Vehicular Technology*, vol. 68, no. 5, pp. 5005–5016, Mar. 2019.
- [3] K. S. Kim *et al.*, "Ultrareliable and low-latency communication techniques for tactile internet services," *Proceedings of the IEEE*, vol. 107, no. 2, pp. 376–393, Feb. 2018.
- [4] H. Viswanathan and P. E. Mogensen, "Communications in the 6G era," *IEEE Access*, vol. 8, pp. 57 063–57 074, 2020.
- [5] M. Shirvanimoghaddam *et al.*, "Short block-length codes for ultra-reliable low latency communications," *IEEE Communications Magazine*, vol. 57, no. 2, pp. 130–137, Dec. 2018.
- [6] Z. Zhou, R. Ratasuk, N. Mangalvedhe, and A. Ghosh, "Resource allocation for uplink grant-free ultra-reliable and low latency communications," in *Proc. 87th Vehicular Technology Conference (VTC Spring)*, Jul. 2018, pp. 1–5.
- [7] Z. Li, M. A. Uusitalo, H. Shariatmadari, and B. Singh, "5G URLLC: Design challenges and system concepts," in *Proc. Int. Symp. on Wireless Commun. Systems (ISWCS)*, Aug. 2018, pp. 1–6.

- [8] A. Sampath, P. S. Kumar, and J. M. Holtzman, "On setting reverse link target SIR in a CDMA system," in *Proc. Vehicular Technology Conf. (VTC)*, vol. 2, May 1997, pp. 929–933.
- [9] F. Blaquez-Casado, G. Gomez, M. d. C. Aguayo-Torres, and J. T. Entrambasaguas, "eOLLA: an enhanced outer loop link adaptation for cellular networks," *EURASIP Journal on Wireless Comm. and Networking*, pp. 1–16, Jan. 2016.
- [10] G. Pocovi, K. I. Pedersen, and P. Mogensen, "Joint link adaptation and scheduling for 5G ultra-reliable low-latency communications," *IEEE Access*, vol. 6, pp. 28912–28922, 2018.
- [11] G. Pocovi, A. A. Esswie, and K. I. Pedersen, "Channel quality feedback enhancements for accurate URLLC link adaptation in 5G systems," in *Proc. Vehicular Technology Conf. (VTC2020-Spring)*, May 2020, pp. 1–6.
- [12] 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," *Proceedings of the IEEE*, vol. 109, no. 3, pp. 204–246, 2021.
- [13] S. K. Pulliyakode and S. Kalyani, "Reinforcement learning techniques for outer loop link adaptation in 4G/5G systems," 2017.
- [14] H. Gupta, A. Eryilmaz, and R. Srikant, "Low-complexity, low-regret link rate selection in rapidly-varying wireless channels," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018, pp. 540–548.
- [15] J. Park and S. Baek, "Two-stage thompson sampling for outer-loop link adaptation," *IEEE Wireless Communications Letters*, vol. 10, no. 9, pp. 2004–2008, 2021.
- [16] V. Saxena, H. Tullberg, and J. Jaldén, "Reinforcement learning for efficient and tuning-free link adaptation," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2021.
- [17] H. Shariatmadari, Z. Li, M. A. Uusitalo, S. Iraj, and R. Jäntti, "Link adaptation design for ultra-reliable communications," in *Proc. International Conference on Communications (ICC)*, May 2016, pp. 1–5.
- [18] L. Buccheri, S. Mandelli, S. Saur, L. Reggiani, and M. Magarini, "Hybrid retransmission scheme for QoS-defined 5G ultra-reliable low-latency communications," in *Proc. IEEE Wireless Commun. and Networking Conf. (WCNC)*, Apr. 2018, pp. 1–6.
- [19] A. Belogaev, E. Khorov, A. Krasilov, D. Shmelkin, and S. Tang, "Conservative link adaptation for ultra reliable low latency communications," in *Proc. International Black Sea Conference on Communications and Networking (BlackSeaCom)*, Jun. 2019, pp. 1–5.
- [20] M. Deghel, S. E. Elayoubi, A. Galindo-Serrano, and R. Visoz, "Joint optimization of link adaptation and harq retransmissions for URLLC services," in *Proc. 25th International Conference on Telecommunications (ICT)*, Jun. 2018, pp. 21–26.
- [21] A. Brighente, J. Mohammadi, and P. Baracca, "Interference distribution prediction for link adaptation in ultra-reliable low-latency communications," in *Proc. 91st Vehicular Technology Conference (VTC2020-Spring)*, May 2020, pp. 1–7.
- [22] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, Sep. 1962.
- [23] 3GPP, "TR 38.901 V14.0.0, Study on channel model for frequencies from 0.5 to 100 GHz," 3GPP, Tech. Rep., May 2017.
- [24] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.
- [25] D. Gesbert, S. Hanly, H. Huang, S. Shamai Shitz, O. Simeone, and W. Yu, "Multi-cell MIMO cooperative networks: A new look at interference," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 9, pp. 1380–1408, Dec. 2010.
- [26] 3GPP, "TR 38.214 V15.3.0, Physical layer procedures for data," 3GPP, Tech. Rep., Oct. 2018.
- [27] V. Kumar and N. B. Mehta, "Modeling and analysis of differential CQI feedback in 4G/5G ofdm cellular systems," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2361–2373, 2019.
- [28] Z. I. Botev, J. F. Grotowski, D. P. Kroese *et al.*, "Kernel density estimation via diffusion," *The annals of Statistics*, vol. 38, no. 5, pp. 2916–2957, Aug. 2010.
- [29] G. R. Terrell and D. W. Scott, "Variable kernel density estimation," *The Annals of Statistics*, pp. 1236–1265, 1992.
- [30] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.
- [31] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Trans. on Information Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.
- [32] 3GPP, "TS 38.212 V16.3.0, Multiplexing and channel coding," Tech. Rep., Oct. 2020.
- [33] K. Sung, H. Haas, and McLaughlin, "A semianalytical PDF of downlink SINR for femtocell networks," *EURASIP Journal on Wireless Comm. and Networking*, pp. 1–9, Feb. 2010.
- [34] G. Plonka and M. Tasche, "Fast and numerically stable algorithms for discrete cosine transforms," *Linear algebra and its applications*, vol. 394, pp. 309–345, Jan. 2005.
- [35] R. L. Burden and D. J. Faires, *Numerical analysis*. PWS Publishing Company, 1985.
- [36] 3GPP, "TS 22.104 V17.4.0, Service requirements for cyber-physical control applications in vertical domains," 3GPP, Tech. Rep., Sep. 2020.
- [37] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. ElBakoury, "Ultra-low latency (ULL) networks: The IEEE TSN and IETF DetNet standards and related 5G ULL research," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 88–145, 2019.