Tracking of quantized signals based on online kernel regression

Emilio Ruiz-Moreno and Baltasar Beferull-Lozano

***Abstract* — Kernel-based approaches have achieved noticeable success as non-parametric regression methods under the framework of stochastic optimization. However, most of the kernel-based methods in the literature are not suitable to track sequentially streamed quantized data samples from dynamic environments. This shortcoming occurs mainly for two reasons: first, their poor versatility in tracking variables that may change unpredictably over time, primarily because of their lack of flexibility when choosing a functional cost that best suits the associated regression problem; second, their indifference to the smoothness of the underlying physical signal generating those samples. This work introduces a novel algorithm constituted by an online regression problem that accounts for these two drawbacks and a stochastic proximal method that exploits its structure. In addition, we provide tracking guarantees by analyzing the dynamic regret of our algorithm. Finally, we present some experimental results that support our theoretical analysis and show that our algorithm has a favorable performance compared to the state-of-the-art.**

A.1 Introduction

Regression problems are some of the most important problems due to their numerous applications and relevance in a wide range of fields. In practice, regression problems are usually formulated as convex optimization problems with strongly convex objectives over feasible convex sets. Besides being one of the most benign settings, this formulation includes significant instances of interest, such as those arising in regularized regression [19], for example, to reduce the complexity of the reconstruction by promoting smoothness. Because of this, such strongly convex objectives are commonly set as the sum of a convex loss, which reflects how far the solution lies from the data samples, and a strongly convex regularizer, which controls the complexity of the solution. On the other hand, most real-world scenarios where regression techniques may be useful occur under dynamic environments. This fact motivates the design of online methods. They allow tracking over time the underlying target signals in a recursive manner with reduced memory and computational needs.

In particular, this paper focuses on sequentially streamed quantized signals. When the underlying physical process generating the signal data samples is unknown, as usual in practice, instead of blindly selecting a certain ad-hoc parametric regression model, the target signal can be estimated from the data samples. This can be done by means of non-parametric regression methods at the expense of a certain memory and computational cost that can be controlled.

Under the mathematical framework of Reproducing Kernel Hilbert Spaces (RKHSs) and thanks to the Representer Theorem [80], such a non-parametric estimate can be constructed from a pre-selected reproducing kernel with a complexity that grows linearly with the number of data samples. Regression with kernels and its online variants have been widely studied in the literature [60, 22]. Their main strength is that they are able to find non-linear patterns at a reasonable computational cost. The Naive Online regularized Risk Minimization Algorithm (NORMA) [12] is arguably the most representative algorithm from the stochastic approximation kernel-based perspective. In its standard form, it concentrates all the novelty in the new expansion coefficient of the signal estimate. However, intuitively, it seems reasonable to distribute the novelty among several expansion coefficients that contribute to the signal estimate instead. In this way, the novelty and correction of previous estimate errors are integrated, more ergonomically, in the signal estimate.

To the best of our knowledge, most of the existing literature has focused on controlling the signal estimate complexity rather than focusing on strategies to control the error in the estimates. Examples of research works controlling complexity are truncation [12] and model-order control via dictionary refining [81], among others [82, 83]. Only some works have studied reducing the signal estimate errors by means of a sliding window scheme [84, 23, 85]. However, in [84], the *selection criterion* to choose among all possible function estimates is least squares making it unsuitable for more general settings, such as incorporating quantization intervals instead of signal values. Similarly, in [23], even though its selection criterion allows certain freedom, regularization is not encouraged and therefore, the smoothness of the underlying physical signal is not fully promoted. Lastly, in [85], the selection criterion is constructed as a regularized augmentation of instantaneous loss-data pairs. As a result, it naturally extends NORMA in a sliding window scheme. Nonetheless, in this work, we present a novel algorithm constituted by a robust selection criterion alongside a conveniently engineered optimization method that outperforms all these algorithms for the task of regression-based tracking of quantized signals.

The paper is structured as follows: Sec. A.2 presents the *windowed cost* and formulates the problem from a learner-adversary perspective. Then, in Sec. A.3, we provide our main contribution: a novel method to minimize the windowed cost via proximal average functional gradient descent. The resulting approach, a novel algorithm called WORM, is used for the practical use case of regression-based tracking of quantized signals. Next, in Sec. A.4, we provide its tracking guarantees through a dynamic regret analysis. Finally, in Sec. A.5, we analyze the experimental performance of our algorithm using synthetic data, and Sec. A.6 concludes the paper.

A.2 Problem formulation

Given a possibly endless sequence $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^N$ of data samples $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^2$, with strictly increasing and non necessarily equispaced timestamps $\{x_i\}_{i=1}^N$, consider an online learning setting where the data samples become available sequentially. Under this scenario, online regression problems can be understood as an interplay between an algorithm (or learner) and an adversary (or environment) [86, 87]. At each step $n \in \mathbb{N}$, an algorithm proposes a function estimate, which we denote as f_n, from an RKHS $\mathcal{H}$. In response, an adversary selects a functional cost $\mathcal{C}_n : \mathcal{H} \to \mathbb{R}$ and penalizes the proposed function estimate with the incurred cost $\mathcal{C}_n(f_n)$. Then, the adversary reveals relevant information about the form of $\mathcal{C}_n$ that is used by the algorithm at the next step.

Unlike most of the previous work, which uses an instantaneous functional cost, i.e., a functional evaluated over one data sample, we formulate the *hypothesis* that a concurrent functional cost, i.e., a functional that considers up to $L \in \mathbb{N}$ data samples simultaneously, may lead to better performance at the expense of a higher but bounded computational cost.

In order to test our hypothesis, we first consider a proper convex *instantaneous loss* $\ell_n : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ given by

$$\ell_n(f) \triangleq \ell(\langle f, k(x_n, \cdot)\rangle_{\mathcal{H}}, y_n) = \ell(f(x_n), y_n), \tag{A.1}$$

where $k(x_n, \cdot)$ is the reproducing kernel associated with the RKHS $\mathcal{H}$ centered at x_n. Notice that the equality in (A.1) holds thanks to the reproducing property [12]. Consequently, we define the so-called windowed cost as a composite of a weighted arithmetic mean of an instantaneous loss as in (A.1), computed over L consecutive data samples and the squared Hilbert norm associated to $\mathcal{H}$ as the regularizer, i.e.,

$$\mathcal{C}_n(f) \triangleq \mathcal{L}_n(f) + \frac{\lambda}{2}\|f\|^2_{\mathcal{H}}, \tag{A.2}$$

with regularization parameter $\lambda > 0$ and where the *windowed loss* $\mathcal{L}_n : \mathcal{H} \to \mathbb{R}\cup\{\infty\}$ is given by

$$\mathcal{L}_n(f) = \sum_{i=l_n}^{n} \omega_i^{(n)} \ell_i(f), \tag{A.3}$$

where $l_n = \max\{1, n - L + 1\}$ and $\sum_{i=l_n}^{n} \omega_i^{(n)} = 1$ with $\omega_i^{(n)} \geq 0$. Finally, the RKHS $\mathcal{H}$, the instantaneous loss ℓ, the regularizer parameter λ and the tuning routine of the convex weights $\{\omega_i^{(n)}\}_{i=l_n}^{n}$ are specified by the user.

A.2.1 Performance analysis

The performance of an online algorithm can be measured by comparing the total cost incurred by the algorithm, given by $\sum_{n=1}^{N} \mathcal{C}_n(f_n)$, and the total corresponding cost incurred by a genie that knows all the costs in advance, that is, $\sum_{n=1}^{N} \mathcal{C}_n(f_n^*)$, where $f_n^* = \arg\min_{f\in\mathcal{H}} \mathcal{C}_n(f)$. Such a metric, referred to as *dynamic regret*, is defined as

$$\mathbf{Reg}_N \triangleq \sum_{n=1}^{N} \mathcal{C}_n(f_n) - \mathcal{C}_n(f_n^*). \tag{A.4}$$

The dynamic regret captures how well the sequence of function estimates $\{f_n\}_{n=1}^{N}$ matches the sequence of optimal decisions in environments that may change unpredictably over time. In general, obtaining a bound on the dynamic regret may not be possible [66]. However, under some mild assumptions on the sequence of functional costs, it is possible to derive worst-case bounds in terms of the cumulative variation of the optimal function estimates

$$\mathrm{C}_N = \sum_{n=2}^{N} \|f_n^* - f_{n-1}^*\|_{\mathcal{H}}. \tag{A.5}$$

In fact, some interesting bounds can be derived if we consider specific rates of variability [86], namely, from zero cumulative variation to a steady tracking error.

A.3 Proposed solution

The smoothness of a windowed cost, as in (A.2), depends on whether or not its instantaneous loss is smooth. Steepest descent methods have been traditionally used for differentiable problems. As for non-differentiable problems, they can be handled, in principle, via the subgradient method and its variants. However, when the cost consists of a composite of a smooth and a non-smooth term, proximal gradient descent methods are preferable because they provide faster convergence as compared to subgradient methods [37]. Therefore, if the instantaneous loss is a proximable[1] non-smooth functional, we propose to minimize the windowed cost via proximal average functional gradient descent due to its favorable convergence performance [88].

A.3.1 Stochastic proximal average functional gradient descent

Our proposed algorithm, the Windowed Online Regularized cost Minimization (WORM), makes use of the stochastic proximal average functional gradient descent. Encouraged by the windowed loss in (A.3), it exploits the concept of the so-called *proximal average functional*. Let us first to introduce some definitions to motivate our algorithm:

Given an RKHS $\mathcal{H}$, a closed proper convex functional $\ell : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ and a real parameter $\eta > 0$, the Moreau envelope of ℓ with smoothing parameter η, is defined as

$$\mathrm{M}_\ell^\eta(h) \triangleq \inf_{g \in \mathcal{H}} \left\{ \ell(g) + \frac{1}{2\eta} \|g - h\|_\mathcal{H}^2 \right\}, \tag{A.6}$$

for all $h \in \mathcal{H}$. The Moreau envelope is a smooth functional that is continuously differentiable (even if ℓ is not), and such that the set of minimizers of ℓ and M_ℓ^η are the same. Thus, the problems of minimizing ℓ and M_ℓ^η can be shown to be equivalent [37]. In addition, a derivative step with respect to the Moreau envelope corresponds to a proximal step with respect to the original function, i.e.,

$$\partial \mathrm{M}_\ell^\eta = \eta^{-1} \left(\mathrm{I} - \mathrm{prox}_\ell^\eta \right), \tag{A.7}$$

where $\mathrm{I} : \mathcal{H} \to \mathcal{H}$ is the identity operator and $\mathrm{prox}_\ell^\eta : \mathcal{H} \to \mathcal{H}$ is the proximal operator defined as

$$\mathrm{prox}_\ell^\eta(h) \triangleq \arg\min_{g \in \mathcal{H}} \left\{ \ell(g) + \frac{1}{2\eta} \|g - h\|_\mathcal{H}^2 \right\}, \tag{A.8}$$

for all $h \in \mathcal{H}$. Notice that since the objective in (A.8) is strongly convex, the proximal map is single-valued.

Next, we denote by $\mathcal{L}_n^\eta$ the so-called proximal average functional of the windowed loss in (A.3) at instant n with real parameter $\eta > 0$, as the unique closed proper convex functional such that

$$\mathrm{M}_{\mathcal{L}_n^\eta}^\eta = \sum_{i=l_n}^{n} \omega_i^{(n)} \mathrm{M}_{\ell_i}^\eta, \tag{A.9}$$

[1] Its proximal operator can be computed efficiently.

where $\ell_i \triangleq \ell(f(x_i), y_i)$ for all $f \in \mathcal{H}$. Even though it is possible to derive an explicit expression for the proximal average functional from its definition (definition 4.1, [65]), for the sake of clarity, and since only its existence is needed for the algorithm, we do not include its explicit form here.

At each iteration n, our algorithm executes the steps:

$$\bar{f}_n = f_n - \eta \partial_f \left. \frac{\lambda}{2} \|f\|_{\mathcal{H}}^2 \right|_{f=f_n}, \tag{A.10a}$$

$$f_{n+1} = \operatorname{prox}_{\mathcal{L}_n^\eta}^\eta (\bar{f}_n), \tag{A.10b}$$

with $0 \leq \eta < \lambda^{-1}$. The first algorithm step (A.10a) is equivalent to $\bar{f}_n = \rho f_n$ with $\rho \triangleq (1 - \eta\lambda) \in [0, 1)$. The proximal operator $\operatorname{prox}_{\mathcal{L}_n^\eta}^\eta : \mathcal{H} \to \mathcal{H}$, can be readily computed by differentiating both sides of the definition in (A.9) while applying the Moreau envelope property given by (A.7), getting

$$\operatorname{prox}_{\mathcal{L}_n^\eta}^\eta = \sum_{i=l_n}^{n} \omega_i^{(n)} \operatorname{prox}_{\ell_i}^\eta. \tag{A.11}$$

The remaining steps depend on the choice of the instantaneous loss. In particular, since we are interested in quantized signals, an adequate functional instantaneous loss must not penalize the function estimates that pass through the intervals. We develop further this reasoning in Sec. A.3.2.

A.3.2 Application to online regression of quantized signals

Consider the sequence of quantization intervals, where the ith quantization interval is given by its timestamp $x_i \in \mathbb{R}$, center $y_i \in \mathbb{R}$ and quantization half step-size $\epsilon \in \mathbb{R}_+$. Subsequently, we can construct its associated sequence of closed hyperslabs, each one of them defined as the convex set

$$H_i \triangleq \{f \in \mathcal{H} : |f(x_i) - y_i| \leq \epsilon\}, \tag{A.12}$$

that contains all the functions in $\mathcal{H}$ passing through the ith quantization interval, and use the metric distance functional to the ith hyperslab

$$d_i(f) \triangleq \inf_{h \in H_i} \|f - h\|_{\mathcal{H}} = \|f - P_{H_i}(f)\|_{\mathcal{H}}, \tag{A.13}$$

as an instantaneous loss to discern between all possible function candidates $f \in \mathcal{H}$. The mapping $P_{H_i} : \mathcal{H} \to H_i$ stands for the metric projection onto H_i and can be expressed as $P_{H_i}(f) = f - \beta_i k(x_i, \cdot)$ (example 38, [23]), where every coefficient β_i is computed as

$$\beta_i = \begin{cases} \frac{f(x_i) - y_i - \epsilon}{k(x_i, x_i)}, & \text{if } f(x_i) > y_i + \epsilon, \\ 0, & \text{if } |f(x_i) - y_i| \leq \epsilon, \\ \frac{f(x_i) - y_i + \epsilon}{k(x_i, x_i)}, & \text{if } f(x_i) < y_i - \epsilon. \end{cases} \tag{A.14}$$

For practical purposes, the relation in (A.13) can be equivalently computed as $d_i(f) = |\beta_i| k(x_i, x_i)^{\frac{1}{2}}$.

Regarding the tuning routine of the convex weights in (A.3), recall that if the set $\{i \in [l_n, n] : \bar{f}_n \notin H_i\} = \emptyset$, any choice of convex weights incurs zero windowed loss. If not, each convex weight is tuned as

$$\omega_i^{(n)} = \frac{d_i(\bar{f}_n)^m}{\sum_{j=l_n}^{n} d_j(\bar{f}_n)^m} = \frac{|\bar{\beta}_i^{(n)}|^m k(x_i, x_i)^{\frac{m}{2}}}{\sum_{j=l_n}^{n} |\bar{\beta}_j^{(n)}|^m k(x_j, x_j)^{\frac{m}{2}}}, \tag{A.15}$$

where $\bar{\beta}_i^{(n)}$ comes from the metric projection map $P_{H_i}(\bar{f}_n)$ and m is a user predefined non-negative real power. In this way, if $m = 0$ the convex weights are all equal. On the other hand, when m tends to infinity, only the weight associated to the largest distance is considered. Thus, the power m allows, with a range of flexibility, to weigh more those windowed loss terms in which the intermediate update $\bar{f}_n$ incurs a larger loss.

Accordingly, from the proximal operator of the metric distance (Chapter 6, [89]) with parameter η, i.e.,

$$\text{prox}_{d_i}^{\eta}(\bar{f}_n) = \bar{f}_n + \min\left\{1, \frac{\eta}{d_i(\bar{f}_n)}\right\}(P_{H_i}(\bar{f}_n) - \bar{f}_n) \tag{A.16}$$

and the proximal average decomposition in (A.11), we can rewrite the algorithm step (A.10b) as

$$f_{n+1} = \bar{f}_n - \sum_{i=l_n}^{n} \omega_i^{(n)} \min\left\{1, \frac{\eta}{d_i(\bar{f}_n)}\right\} \bar{\beta}_i^{(n)} k(x_i, \cdot). \tag{A.17}$$

Finally, assuming that the algorithm does not have access to any a priori information when it encounters the first data sample, we can set $f_1 = 0$. Then, from the algorithm step (A.10a), substituting each function estimate by its kernel expansion, i.e., $f_n = \sum_{i=1}^{n-1} \alpha_i^{(n)} k(x_i, \cdot)$ and identifying terms in (A.17), we obtain the following closed-form update rule for the non-parametric coefficients

$$\alpha_i^{(n+1)} = \begin{cases} \rho\alpha_i^{(n)} - \omega_i^{(n)}\Gamma_{\eta,i}^{(n)} & \text{if } i \in [1, n-1], \\ -\omega_i^{(n)}\Gamma_{\eta,i}^{(n)} & \text{if } i = n, \end{cases} \tag{A.18}$$

where $\Gamma_{\eta,i}^{(n)} \triangleq \min\left\{|\bar{\beta}_i^{(n)}|, \eta k(x_i, x_i)^{-\frac{1}{2}}\right\} \text{sign}(\bar{\beta}_i^{(n)})$ if $i \in [l_n, n]$ and equals zero otherwise.

A.3.2.1 Sparsification

The WORM algorithm, like many other kernel-based algorithms, suffers from the curse of kernelization [82], i.e., unbounded linear growth in model size and update time with the amount of data. For the considered application in Sec. A.3.2, a simple complexity control mechanism as kernel series truncation allows to preserve, to some extent, both performance as well as theoretical tracking guarantees, as we show in Secs. A.4 and A.5. Thus, given a user-defined truncation parameter $\tau \in \mathbb{N}$, such that $\tau > L$, if the number of effective coefficients constituting the function estimate f_n exceeds τ, we remove the older expansion term, i.e.,

$$e_n = \alpha_{n-\tau}^{(n)} k(x_{n-\tau}, \cdot), \tag{A.19}$$

where $\alpha_{n-\tau}^{(n)} = \rho^{n-\tau+L}\alpha_{n-\tau}^{(n-\tau+L)}$. For the sake of illustration, consider a Gaussian reproducing kernel, i.e., $k(x,t) = \exp(-(x-t)^2/(2\sigma^2))$ with positive width σ. Then, the contribution of the truncated term e_n at timestamp x_n depends on the ratio $(x_{n-\tau} - x_n)^2/\sigma^2$; hence, a bigger ratio implies a smaller *truncation error*. Truncating before the algorithm step (A.10a) allows distributing the effects of the truncation error among the elements within the window. **Algorithm 2** describes in pseudocode our truncated WORM algorithm.

Algorithm 2 truncated WORM

Input: The data tuples $\{(x_n, y_n)\}_n^N$, the quantization half step-size ϵ, an RKHS $\mathcal{H}$, the window length L, the regularization parameter λ, the learning rate η, the power m and the truncation parameter τ.

Set $\boldsymbol{\alpha} := \text{queue}([\,], \text{maxlen} = \tau)$.
for $n = 1, 2, \ldots$ **do**
    Append one zero to the queue $\boldsymbol{\alpha}$.
    Set $\boldsymbol{\alpha} := (1 - \eta\lambda)\boldsymbol{\alpha}$.
    Set $\zeta_L := \{\max\{1, n - L + 1\}, \ldots, n\}$ and
    $\zeta_\tau := \{\max\{1, n - \tau + 1\}, \ldots, n\}$.
    for i in ζ_L **do**
        Compute $\bar{f}(x_i) := \sum_{j \in \zeta_\tau} \alpha_{j-\max\{n-\tau,0\}} k(x_j, x_i)$
        Compute $\bar{\beta}_i$ w.r.t. $\bar{f}$ as in (A.14).
    end for
    Set $\zeta_{\bar{f}} := \{i \in \zeta_L : \bar{\beta}_i \neq 0\}$.
    for i in $\zeta_{\bar{f}_n}$ **do**
        Compute the convex weights ω_i as in (A.15).
        Compute $\Gamma_{\eta,i} := \min\{|\bar{\beta}_i|, \eta k(x_i, x_i)^{-\frac{1}{2}}\}\text{sign}(\bar{\beta}_i)$.
        Update $\alpha_i := \alpha_i - \omega_i \Gamma_{\eta,i}$.
    end for
Output: The vector $\boldsymbol{\alpha}$, which yields the function estimate
    $f_n = \sum_{i=\max\{1,n-\tau+1\}}^{n-1} \alpha_{i-\max\{n-\tau,0\}} k(x_i, \cdot)$.
end for

A.4 Dynamic regret analysis

In this section, we derive a theoretical upper bound for the dynamic regret incurred by the truncated WORM algorithm. As a standard assumption [87], suppose that the norms $\|\mathcal{C}_n'(f_n)\|_{\mathcal{H}}$ are bounded by a positive constant G, i.e.,

$$\sup_{f_n \in \mathcal{H}, n \in [1,N]} \|\mathcal{C}_n'(f_n)\|_{\mathcal{H}} \leq G. \tag{A.20}$$

For the sake of notation, we omit the sub-index $\mathcal{H}$ in inner products and norms since the RKHS is clear by context. Considering the assumption in (A.20) and the first order convexity condition of the windowed cost,

$$
\begin{aligned}
\mathbf{Reg}_N &= \sum_{n=1}^{N} \mathcal{C}_n(f_n) - \mathcal{C}_n(f_n^*) \leq \sum_{n=1}^{N} \langle \mathcal{C}_n'(f_n), f_n - f_n^* \rangle \\
&\leq \sum_{n=1}^{N} \|\mathcal{C}_n'(f_n)\| \|f_n - f_n^*\| \leq G \sum_{n=1}^{N} \|f_n - f_n^*\|,
\end{aligned}
\tag{A.21}
$$

it is clear that the dynamic regret is bounded above.

Consider the distance between the function estimate f_{n+1} and the optimal estimate $f_n^* = \arg\min_{f \in \mathcal{H}} \mathcal{C}_n(f_n)$, i.e.,

$$
\|f_{n+1} - f_n^*\| = \|\text{prox}^{\eta}_{\mathcal{L}_n^{\eta}}(\bar{f}_n) - \text{prox}^{\eta}_{\mathcal{L}_n^{\eta}}(\bar{f}_n^*)\|. \tag{A.22}
$$

Hence, from the relation (A.22), the firmly non-expansiveness of the proximal operator [37], and the method step (A.10a) with truncation, we achieve the following inequality

$$
\|f_{n+1} - f_n^*\| \leq \rho \|f_n - e_n - f_n^*\| \leq \rho \|f_n - f_n^*\| + \rho \|e_n\|, \tag{A.23}
$$

with coefficient $\rho \triangleq (1 - \eta\lambda) \in [0, 1)$. Finally, we can rewrite

$$
\begin{aligned}
&\sum_{n=1}^{N} \|f_n - f_n^*\| = \|f_1 - f_1^*\| + \sum_{n=2}^{N} \|f_n - f_n^*\| \\
&= \|f_1 - f_1^*\| + \sum_{n=2}^{N} \|f_n - f_{n-1}^* + f_{n-1}^* - f_n^*\| \\
&\leq \|f_1 - f_1^*\| + \sum_{n=2}^{N} \|f_n - f_{n-1}^*\| + \sum_{n=2}^{N} \|f_n^* - f_{n-1}^*\| \\
&\leq \|f_1 - f_1^*\| + \rho \sum_{n=2}^{N} \|f_{n-1} - f_{n-1}^*\| + \mathrm{C}_N + \rho \mathrm{E}_N \qquad \text{(A.24a)} \\
&\leq \|f_1 - f_1^*\| + \rho \sum_{n=1}^{N} \|f_n - f_n^*\| + \mathrm{C}_N + \rho \mathrm{E}_N, \qquad \text{(A.24b)}
\end{aligned}
$$

where the step (A.24a) comes after using the relation (A.23), the definition of cumulative variation in (A.5), and renaming the *cumulative truncation error* $\mathrm{E}_N \triangleq \sum_{n=2}^{N} \|e_n\|$. In step (A.24b), we rename the summation index and add the positive term $\rho\|f_N - f_N^*\|$ to the right hand-side of the inequality.

Regrouping the terms in (A.24), leads to

$$
\sum_{n=1}^{N} \|f_n - f_n^*\| \leq \frac{1}{1-\rho} \left(\|f_1 - f_1^*\| + \mathrm{C}_N + \rho \mathrm{E}_N \right) \tag{A.25}
$$

and substituting the relation obtained in (A.25) into the inequality (A.21), allows to upper-bound the dynamic regret as

$$
\mathbf{Reg}_N \leq \frac{G}{1-\rho} \left(\|f_1 - f_1^*\| + \mathrm{C}_N + \rho \mathrm{E}_N \right). \tag{A.26}
$$

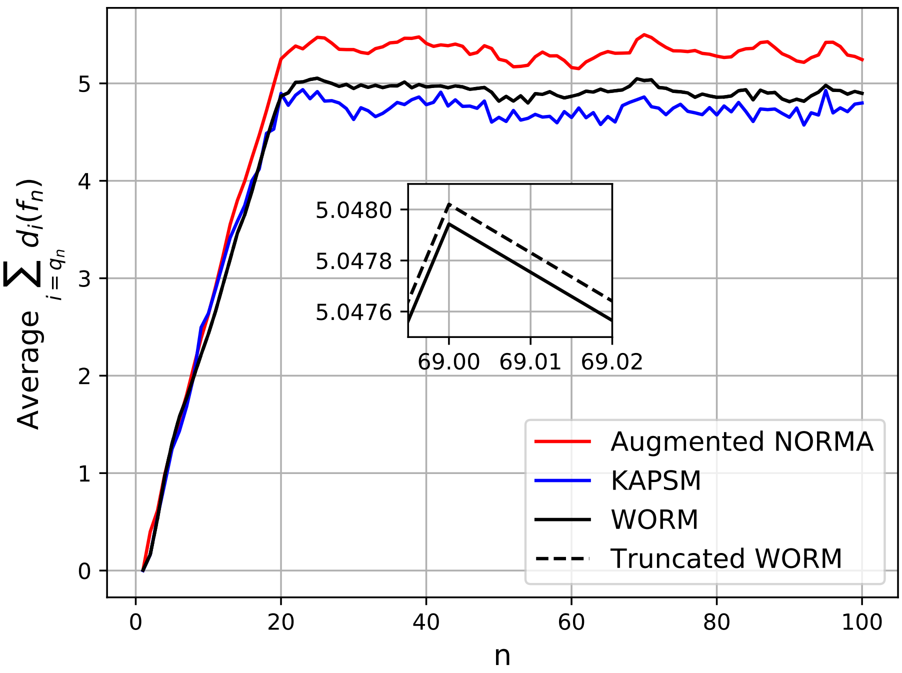

Figure A.1: Average q-inconsistency of the sequence of function estimates $\{f_n\}_{n=1}^{100}$ over 500 different quantized signals.

This result explicitly shows the trade-off between tracking accuracy and model complexity [85]. In other words, without truncation, the dynamic regret reduces to $\mathbf{Reg}_N \leq \mathcal{O}(1 + \mathrm{C}_N)$, depending entirely on the environment. On the other hand, if we control the complexity of the function estimates via any truncation strategy such that the norm of the truncation error is upper bounded by a positive constant, i.e., $\sup_{n\in[1,N]} \|e_n\| \leq \delta$, the dynamic regret reduces to $\mathbf{Reg}_N \leq \mathcal{O}(\mathrm{C}_N + \delta T)$, leading to a steady tracking error in well-behaved environments.

A.5 Experimental results

As suggested in Sec. A.1, we compare the performance of our algorithm WORM with the KAPSM algorithm [23] and the augmented version of NORMA proposed in [85]. Moreover, since complexity control methods aim to limit the model order of the function estimate by lower-order approximations, we do not consider here any of them in order to isolate their effects on the performance of the algorithms. However, we have considered the truncated version of the WORM algorithm in our experiments to show that even a low complexity control technique such as truncation may lead to competitive performance.

Considering the application described in Section A.3.2, we have generated quantized versions of 500 realizations of a given AR(1) process. Each realization has been carried out for 100 data samples. In turn, each sample has been computed recurrently via $z_n = \varphi z_{n-1} + u_n$ with $z_0 = 0$, parameter $\varphi = 0.9$ and Gaussian noise $u_n \sim \mathcal{N}(0, 1)$. The center of the quantization intervals is computed by means of $y_n = \mathrm{round}(z_n/\epsilon) \cdot \epsilon$ with quantization half-step $\epsilon = 0.5$. The corresponding times-

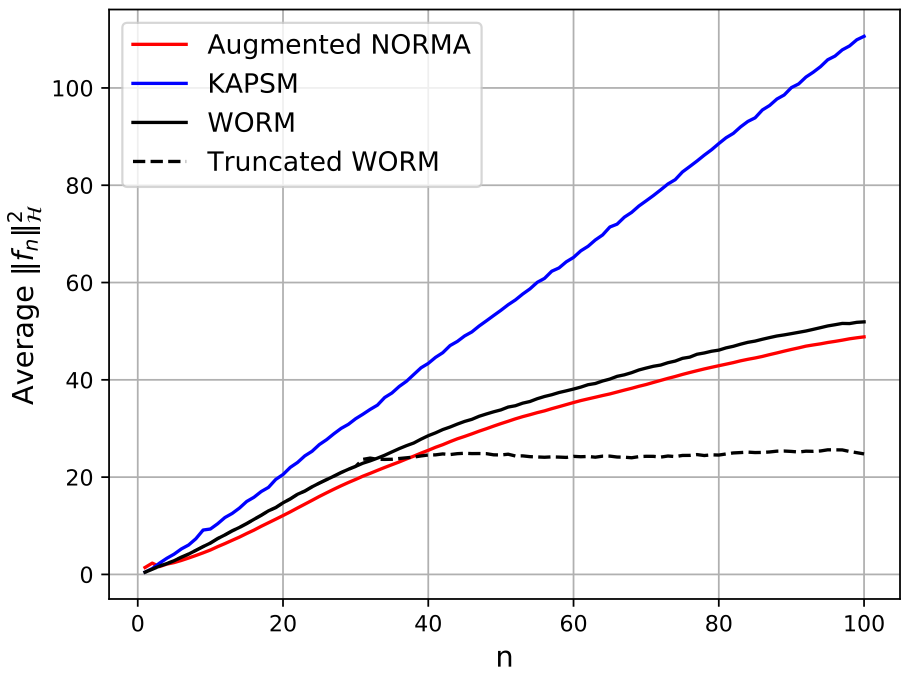

Figure A.2: Average complexity of the sequence of function estimates $\{f_n\}_{n=1}^{100}$ over 500 different quantized signals.

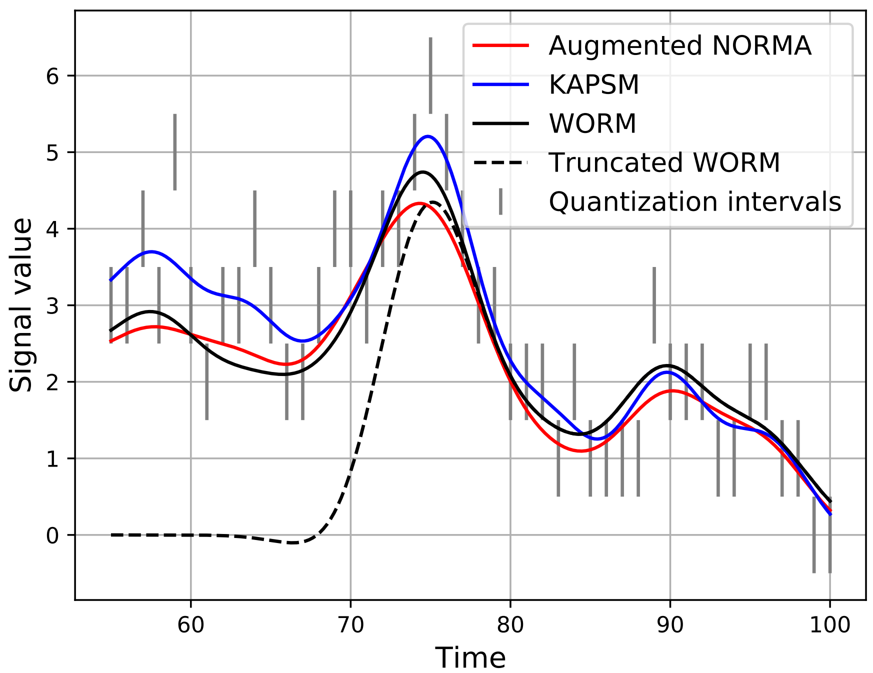

Figure A.3: Comparison of regression plots for the last function estimate f_{100}, over the last 45 data samples of a synthetically generated quantized signal.

tamps $\{x_n\}_{n=1}^{100}$ are uniformly arranged. For the sake of illustration, we use a Gaussian reproducing kernel, i.e., $k(x,t) = \exp(-(x-t)^2/(2\sigma^2))$, with $\sigma = 3$. All four algorithms use the same window length $L = 10$. As to the augmented NORMA, the WORM algorithm and its truncated version, all use the same learning rate $\eta = 1.5$ and regularization parameter $\lambda = 0.005$. We restrict the truncated WORM function estimates expansion to a maximum of 30 terms, i.e., $\tau = 30$. Both versions of WORM use the power $m = 2$. For the augmented NORMA, the instantaneous loss terms within the nth window are equally weighted with the weight $\min\{n, L\}^{-1}$ and $\partial_f d_i(f_n) = \mathrm{sign}(\beta_i^{(n)})k(x_i,x_i)^{-\frac{1}{2}}k(x_i,\cdot)$ is used as a valid functional subgradient. We also define the *q-inconsistency*, i.e., $\sum_{i=q_n}^{n} d_i(f_n)$, with $q_n = \max\{n-q+1, 1\}$ and $q = 20$, and use the squared Hilbert norm, $\|f_n\|_{\mathcal{H}}^2 = \sum_{i,j=\tau_n}^{n} \alpha_i^{(n)}\alpha_j^{(n)}k(x_i,x_j)$, with $\tau_n = \max\{n-\tau+1, 1\}$, as performance metrics for the function estimates. The first metric measures how far is the function estimate of falling into the last q received quantization intervals. The second metric measures the function estimate complexity.

As shown in Fig. A.1 and Fig. A.2, there is a trade-off between q-inconsistency and complexity. The WORM algorithm successfully balances both altogether. As to its truncated version, the same experimental results show that the complexity can be successfully controlled at the expense of little accuracy. Finally, Fig. A.3 shows a snapshot of the last function estimate f_{100} for each algorithm.

A.6 Conclusion

In this paper, we propose a novel algorithm, WORM, for regression-based tracking of quantized signals. We derive a theoretical dynamic regret bound for WORM that ensures tracking guarantees. Our experiment shows that WORM provides better signal reconstruction in terms of consistency and smoothness altogether compared to the state-of-the-art.