entropy

MDPI

Article

Kernel Risk-Sensitive Mean p-Power Error Algorithms for Robust Learning

Tao Zhang [1,2], Shiyuan Wang [1,2,*], Haonan Zhang [1,2], Kui Xiong [1,2] and Lin Wang [1,2]

[1] College of Electronic and Information Engineering, Southwest University, Chongqing 400715, China; zhangtao1996@email.swu.edu.cn (T.Z.); zhanghaonan1@email.swu.edu.cn (H.Z.); xiongk@email.swu.edu.cn (K.X.); irenewang@swu.edu.cn (L.W.)

[2] Chongqing Key Laboratory of Nonlinear Circuits and Intelligent Information Processing, Chongqing 400715, China

* Correspondence: wsy@swu.edu.cn

Received: 8 May 2019; Accepted: 12 June 2019; Published: 13 June 2019

Abstract: As a nonlinear similarity measure defined in the reproducing kernel Hilbert space (RKHS), the correntropic loss (C-Loss) has been widely applied in robust learning and signal processing. However, the highly non-convex nature of C-Loss results in performance degradation. To address this issue, a convex kernel risk-sensitive loss (KRL) is proposed to measure the similarity in RKHS, which is the risk-sensitive loss defined as the expectation of an exponential function of the squared estimation error. In this paper, a novel nonlinear similarity measure, namely kernel risk-sensitive mean p-power error (KRP), is proposed by combining the mean p-power error into the KRL, which is a generalization of the KRL measure. The KRP with $p = 2$ reduces to the KRL, and can outperform the KRL when an appropriate p is configured in robust learning. Some properties of KRP are presented for discussion. To improve the robustness of the kernel recursive least squares algorithm (KRLS) and reduce its network size, two robust recursive kernel adaptive filters, namely recursive minimum kernel risk-sensitive mean p-power error algorithm (RMKRP) and its quantized RMKRP (QRMKRP), are proposed in the RKHS under the minimum kernel risk-sensitive mean p-power error (MKRP) criterion, respectively. Monte Carlo simulations are conducted to confirm the superiorities of the proposed RMKRP and its quantized version.

Keywords: correntropic; quantized; kernel risk-sensitive mean p-power error; recursive; kernel adaptive filters

1. Introduction

Online kernel-based learning is to extend the kernel methods to online settings where the data arrives sequentially, which has been widely applied in signal processing thanks to its excellent performance in addressing nonlinear issues [1]. The development of kernel methods is of great significance for practical applications. In kernel methods, the input data are transformed from the original space into the reproducing kernel Hilbert space (RKHS) using the kernel trick [2]. As the representative of the kernel methods, kernel adaptive filters (KAFs) provide an effective way to transform a nonlinear problem into a linear one, which have been widely introduced in system identification and time-series prediction [3–5]. Generally, KAFs are designed for Gaussian and non-Gaussian noises from the aspect of cost function, respectively.

For Gaussian noises, the second-order similarity measures of errors are generally used as a cost function of KAFs to achieve desirable filtering accuracy. Therefore, in the Gaussian noise environment, KAFs based on the second-order similarity measures of errors are mainly divided into three categories, i.e., the kernel least mean square (KLMS) algorithm [6], the kernel affine

projection algorithm (KAPA) [7], and the kernel recursive least square algorithm (KRLS) [8]. However, the network size of KAFs increases linearly with the length of training, leading to large computational and storage burdens. To curb this structure growth, many sparsification methods are required, such as the surprise criterion (SC) [9], novelty criterion (NC) [10], coherence criterion [11], and approximate linear dependency (ALD) criterion [8]. However, these sparsification methods only discard the redundant data, leading to reduction of filtering accuracy. Unlike the aforementioned sparsification methods, the vector quantization (VQ) utilizes the redundant data to update the weights for accuracy improvement. The VQ is combined into KAFs to generate quantized KAFs, e.g., the quantized kernel least mean square algorithm (QKLMS) [12] and quantized kernel recursive least squares algorithm (QKRLS) [13].

However, the second-order similarity measures used in the aforementioned algorithms merely contain the second order statistics of errors, which cannot address non-Gaussian noises or outliers, efficiently [14]. Thus, it is very important to design a cost function beyond the second-order statistics of errors for combating non-Gaussian noises. The non-second order similarity measures can be divided into three categories, i.e., the mean p-power error (MPE) criterion [15], information theoretic learning (ITL) [14], and risk-sensitive loss (RL) based criteria [16,17]. The MPE criterion based on the pth absolute moment of the error can deal with non-Gaussian data with a proper p-value, efficiently. In general, MPE is robust to large outliers when $p < 2$ [15], generating robust adaptive filters [15], e.g., the kernel least mean p-power (KLMP) algorithm [18] and the kernel recursive least mean p-power (KRLP) algorithm [18]. ITL can incorporate the complete distribution of errors into the learning process, resulting in the improvement of filtering precision and robustness to outliers. The most widely used ITL criterion is the maximum correntropy criterion (MCC) [19–24]. As a local similarity measure defined as a generalized correlation in the RKHS, the correntropy used in MCC can leverage higher order statistics of data to combat outliers [25]. However, the performance surface of the correntropic loss (C-Loss) is highly non-convex, which may lead to poor convergence performance. In the RL-based criteria, e.g., minimum risk-sensitive loss [16] and minimum kernel risk-sensitive loss (MKRL) [17,26], the risk-sensitive loss in the RKHS is convex extremely, which is more efficient for combating non-Gaussian noises or outliers than MCC [17,26]. However, since the MKRL uses the stochastic gradient descent (SGD) method to update its weights, the desirable filtering performance cannot be achieved for some complex nonlinear issues. The recursive update rules with excellent tracking ability can improve the filtering performance of adaptive filtering algorithms [21]. For example, KRLS based on the recursive update rule can improve the filtering performance of KLMS based on the SGD, significantly. To the best of our knowledge, however, it has not yet been proposed to design a recursive MKRL algorithm for desirable filtering performance in the RKHS by a recursive update rule.

In this paper, to inherit the advantages of both KRL and MPE for robustness improvement, we propose the risk-sensitive mean p-power error (RP) defined as the expectation of an exponential function of the pth absolute moment of the estimation error, and its kernel RP (KRP). The KRP can outperform the KRL by setting an appropriate p-value for robust learning, and the KRP with $p = 2$ reduces to the KRL. The proposed KRP criterion is used to derive a novel recursive minimum kernel risk-sensitive mean p-power error (RMKRP) algorithm for desirable filtering performance by combining the weighted output information. Furthermore, to curb the growth of network size in the RMKRP, the VQ is combined into RMKRP to generate quantized RMKRP (QRMKRP).

The rest of this paper is organized as follows. In Section 2, we define the KRP, and give some basic properties. The KRP criterion is derived to develop a recursive adaptive algorithm by combining the weighted output information, called RMKRP algorithm in Section 3. To further reduce the network size of RMKRP, the vector quantization method is applied in RMKRP, thus generating the quantized RMKRP (QRMKRP) in Section 3. In Section 4, Monte Carlo simulations are conducted to validate the superiorities of the proposed algorithms in nonlinear examples. The conclusion is summarized in Section 5.

2. Kernel Risk-Sensitive Mean p-Power Error

2.1. Definition

According to [17], the risk-sensitive loss is defined in RKHS, called the kernel risk-sensitive loss (KRL). Given two arbitrary scalar random variables X and Y, where $X, Y \in \mathbb{R}$, the KRL is defined by

$$\begin{aligned} L_\lambda(X,Y) &= \frac{1}{\lambda}\mathbf{E}\left[\exp\left(\lambda\left(\frac{1}{2}\left\|\varphi(X)-\varphi(Y)\right\|_{\mathbb{F}}^2\right)\right)\right] \\ &= \frac{1}{\lambda}\mathbf{E}\left[\exp\left(\lambda\left(\frac{1}{2}\langle\varphi(X)-\varphi(Y),\varphi(X)-\varphi(Y)\rangle_{\mathbb{F}}\right)\right)\right] \\ &= \frac{1}{\lambda}\mathbf{E}\left[\exp\left(\lambda\left(\frac{1}{2}\left(\langle\varphi(X),\varphi(X)\rangle_{\mathbb{F}}+\langle\varphi(Y),\varphi(Y)\rangle_{\mathbb{F}}-2\langle\varphi(X),\varphi(Y)\rangle_{\mathbb{F}}\right)\right)\right)\right] \\ &= \frac{1}{\lambda}\mathbf{E}[\exp(\lambda(1-\kappa_\sigma(X-Y)))] \\ &= \frac{1}{\lambda}\int\exp(\lambda(1-\kappa_\sigma(X-Y)))dF_{XY}(x,y), \end{aligned} \tag{1}$$

where $\lambda > 0$ is a risk-sensitive scalar parameter; $\varphi(X) = \kappa(X,.)$ is a nonlinear mapping induced by a Mercer kernel $\kappa_\sigma(.)$, which transforms the data from the original space into the RKHS $\mathbb{F}$ equipped with an inner product $\langle .,. \rangle_{\mathbb{F}}$ satisfying $\langle\varphi(X),\varphi(Y)\rangle_{\mathbb{F}} = \varphi^{\mathrm{T}}(X)\varphi(Y) = \kappa_\sigma(X-Y)$; $\mathbf{E}$ denotes the mathematical expectation; $\|.\|_{\mathbb{F}}$ denotes the norm in RKHS $\mathbb{F}$; $F_{XY}(x,y)$ denotes the joint distribution function of (X,Y). A shift-invariant Gaussian kernel $\kappa_\sigma(.)$ with bandwidth σ is given as follows:

$$\kappa_\sigma(x,y) = \kappa_\sigma(x-y) = \exp\left(-\frac{(x-y)^2}{2\sigma^2}\right). \tag{2}$$

However, the joint distribution of (X,Y) is usually unknown, and only N samples $\{x(i),y(i)\}_{i=1}^N$ are available. Hence, the nonparametric estimate of $L_\lambda(X,Y)$ is obtained by applying the Parzen windows [19] as $\hat{L}_\lambda(X,Y) = \frac{1}{N\lambda}\sum_{i=1}^{N}\exp(\lambda(1-\kappa_\sigma(x(i)-y(i))))$. Note that the inner product in the RKHS for the same input is calculated by using kernel trick and (2), i.e, $\varphi^{\mathrm{T}}(X)\varphi(X) = \exp\left(-\frac{(X-X)^2}{2\sigma^2}\right) = 1$.

In this paper, we define a new non-second order similarity measure in the RKHS, i.e., the kernel risk-sensitive mean p-power error (KRP) loss. Given two random variables X and Y, the KRP loss is defined by

$$\begin{aligned} L_{\lambda,p}(X,Y) &= \frac{1}{\lambda}\mathbf{E}\left[\exp\left(\lambda 2^{-p/2}\left(\left\|\varphi(X)-\varphi(Y)\right\|_{\mathbb{F}}^p\right)\right)\right] \\ &= \frac{1}{\lambda}\mathbf{E}\left[\exp\left(\lambda 2^{-p/2}\left(\left\|\varphi(X)-\varphi(Y)\right\|_{\mathbb{F}}^2\right)^{p/2}\right)\right] \\ &= \frac{1}{\lambda}\mathbf{E}[\exp(\lambda 2^{-p/2}(2-2\kappa_\sigma(X-Y))^{p/2})] \\ &= \frac{1}{\lambda}\mathbf{E}[\exp(\lambda(1-\kappa_\sigma(X-Y))^{p/2})] \\ &= \frac{1}{\lambda}\int\exp(\lambda(1-\kappa_\sigma(X-Y))^{p/2})dF_{X,Y}(x,y), \end{aligned} \tag{3}$$

where $p > 0$ is the power parameter. Note that the KRL can be regarded as a special case of the KRP with $p = 2$.

However, the joint distribution of X and Y is usually unknown in practice. Hence, the empirical KRP is defined as follows:

$$\hat{L}_{\lambda,p}(X,Y) = \frac{1}{N\lambda}\sum_{i=1}^{N}\exp(\lambda(1-\kappa_\sigma(x(i)-y(i)))^{p/2}), \tag{4}$$

where $\{x(i),y(i)\}_{i=1}^{N}$ denotes the available finite number of samples. The empirical KRP can be regarded as a distance between both $\mathbf{X} = [x(1),x(2),...,x(N)]^{\mathrm{T}}$ and $\mathbf{Y} = [y(1),y(2),...,y(N)]^{\mathrm{T}}$.

2.2. Properties

In the following, we give some important properties of the proposed KRP.

Property 1. *$L_{\lambda,p}(X,Y)$ is symmetric that is $L_{\lambda,p}(X,Y) = L_{\lambda,p}(Y,X)$.*

Proof. Straightforward since $\kappa_\sigma(X-Y) = \kappa_\sigma(Y-X)$. □

Property 2. *$L_{\lambda,p}(X,Y)$ is positive and bounded, i.e., $\frac{1}{\lambda} \le L_{\lambda,p}(X,Y) \le \frac{1}{\lambda}\exp(\lambda)$, and reaches its minimum if $X = Y$.*

Proof. Straightforward since $0 < \kappa_\sigma(X-Y) \le 1$, and $\kappa_\sigma(X-Y) = 1$ if $X = Y$. □

Property 3. *As λ is small enough, it holds that $L_{\lambda,p}(X,Y) \approx \frac{1}{\lambda} + \mathbf{E}[(1-\kappa_\sigma(X-Y))^{p/2}]$.*

Proof. For a small enough λ, we have $\lambda(1-\kappa_\sigma(X-Y))^{p/2} \to 0$, i.e.,

$$\exp(\lambda(1-\kappa_\sigma(X-Y))^{p/2}) \approx 1+\lambda(1-\kappa_\sigma(X-Y))^{p/2}. \tag{5}$$

Therefore, we can obtain

$$\begin{aligned} L_{\lambda,p}(X,Y) &= \frac{1}{\lambda}\mathrm{E}[\exp(\lambda(1-\kappa_\sigma(X-Y))^{p/2})] \\ &\overset{(5)}{\approx} \frac{1}{\lambda}\mathrm{E}[1+\lambda(1-\kappa_\sigma(X-Y))^{p/2}] \\ &= \frac{1}{\lambda} + \mathrm{E}[(1-\kappa_\sigma(X-Y))^{p/2}]. \end{aligned} \tag{6}$$

□

Property 4. *As σ is large enough, it holds that $L_{\lambda,p}(X,Y) \approx \frac{1}{\lambda} + (2\sigma^2)^{-p/2}\mathbf{E}[|X-Y|^p]$.*

Proof. Since $\exp(x)$ is approximated by $1+x$ for a small enough x, for the case of large enough σ, i.e., $\frac{(X-Y)^2}{2\sigma^2} \to 0$. Thus, we can obtain the approximation as

$$\exp\left(-\frac{(X-Y)^2}{2\sigma^2}\right) \approx 1 - \frac{(X-Y)^2}{2\sigma^2}. \tag{7}$$

Similarly, when $\lambda\left(\frac{(X-Y)^2}{2\sigma^2}\right)^{p/2} \to 0$ for large enough σ, we can also obtain the approximation as

$$\exp\left(\lambda\left(\frac{(X-Y)^2}{2\sigma^2}\right)^{p/2}\right) \approx 1+\lambda\left(\frac{(X-Y)^2}{2\sigma^2}\right)^{p/2}. \tag{8}$$

According to (7) and (8), we have

$$\begin{aligned} L_{\lambda,p}(X,Y) &= \frac{1}{\lambda}\mathrm{E}\left[\exp\left(\lambda\left(1-\exp\left(-\frac{(X-Y)^2}{2\sigma^2}\right)\right)^{p/2}\right)\right] \\ &\overset{(7)}{\approx} \frac{1}{\lambda}\mathrm{E}\left[\exp\left(\lambda\left(\frac{(X-Y)^2}{2\sigma^2}\right)^{p/2}\right)\right] \\ &\overset{(8)}{\approx} \frac{1}{\lambda}\mathrm{E}\left[1+\lambda\left(\frac{(X-Y)^2}{2\sigma^2}\right)^{p/2}\right] \\ &= \frac{1}{\lambda}+(2\sigma^2)^{-p/2}\mathrm{E}\left[|X-Y|^p\right]. \end{aligned} \tag{9}$$

□

Remark 1. *According to Properties* 3 *and* 4*, the KRP is, approximately, equivalent to the KMPE [27] as λ is small enough, and equivalent to the MPE [15] as σ is large enough. Thus, the KMPE and MPE can be viewed as two extreme cases of the KRP.*

Property 5. *As p is small enough, it holds that $L_{\lambda,p}(X,Y) \approx \frac{1}{\lambda}\exp(\lambda(1+(p/2)\mathbf{E}[\log(1-\kappa_\sigma(X-Y))])) \approx \frac{1}{\lambda}\exp(\lambda)$.*

Proof. Property 5 holds because of $(1-\kappa_\sigma(X-Y))^{p/2} \approx 1+(p/2)\mathbf{E}[\log(1-\kappa_\sigma(X-Y))] \approx 1$. □

Property 6. *Let $\mathbf{e} = \mathbf{X}-\mathbf{Y} = [e(1), e(2), ..., e(N)]^{\mathrm{T}}$, where $e(i) = x(i)-y(i)$. The empirical KRP $\hat{L}_{\lambda,p}(\mathbf{X},\mathbf{Y})$ as a function of $\mathbf{e}$ is convex at any point satisfying $\|\mathbf{e}\|_\infty = \max_{i=1,2,...,N}|e(i)| \le \sigma$ and $p \ge 2$. When $\|\mathbf{e}\|_\infty > \sigma$, the empirical KRP $\hat{L}_{\lambda,p}(\mathbf{X},\mathbf{Y})$ is also convex if the risk-sensitive parameter $\lambda > 0$ and power parameter $p \ge \max\limits_{i=1,2,...,N}\left\{\frac{2(e^2(i)-\sigma^2)(1-\kappa_\sigma(e(i)))}{e^2(i)\kappa_\sigma(e(i))}+2\right\}$.*

Proof. Since $\hat{L}_{\lambda,p}(\mathbf{X},\mathbf{Y}) = \frac{1}{N\lambda}\sum\limits_{i=1}^{N}\exp(\lambda(1-\kappa_\sigma(e(i)))^{p/2})$, the Hessian matrix of $\hat{L}_{\lambda,p}(\mathbf{X},\mathbf{Y})$ with respect to $\mathbf{e}$ can be derived as

$$\mathrm{H}_{\hat{L}_{\lambda,p}(\mathbf{X},\mathbf{Y})}(\mathbf{e}) = \left[\frac{\partial^2\hat{L}_{\lambda,p}(\mathbf{X},\mathbf{Y})}{\partial e(i)\partial e(j)}\right] = \mathrm{diag}[\gamma_1,\gamma_2,...,\gamma_N], \tag{10}$$

where

$$\begin{aligned} \gamma_i =& \zeta_i\left(\frac{p\lambda}{2\sigma^2}(1-\kappa_\sigma(e(i)))^{(p-2)/2}\exp\left(-\frac{e^2(i)}{2\sigma^2}\right)e^2(i)+1\right. \\ &\left.+\frac{p-2}{2}(1-\kappa_\sigma(e(i)))^{-1}\exp\left(-\frac{e^2(i)}{2\sigma^2}\right)\frac{e^2(i)}{\sigma^2}-\frac{e^2(i)}{\sigma^2}\right), \end{aligned} \tag{11}$$

with $\zeta_i = \frac{p}{2N\sigma^2}\exp(\lambda(1-\kappa_\sigma(e(i)))^{p/2})(1-\kappa_\sigma(e(i)))^{(p-2)/2}\exp\left(-\frac{e^2(i)}{2\sigma^2}\right), i = 1,2,...,N$. When $p \ge 2$, we have $\mathrm{H}_{\hat{L}_{\lambda,p}(\mathbf{X},\mathbf{Y})}(\mathbf{e}) > 0$ if $\max\limits_{i=1,2,...,N}|e(i)| \le \sigma$. From (11), if $|e(i)| \le \sigma$ and $p \ge 2$, or $|e(i)| > \sigma$ and $p \ge [2(e^2(i)-\sigma^2)(1-\kappa_\sigma(e(i)))/(e^2(i)\kappa_\sigma(e(i)))]+2$, we have $\zeta_i \ge 0$. Therefore, we have $\mathrm{H}_{\hat{L}_{\lambda,p}(X,Y)}(\mathbf{e}) \ge 0$ if

$$p(\lambda(1-\kappa_\sigma(e(i)))^{p/2}+1) \ge \max_{i=1,2,...,N}\left\{\frac{2(e^2(i)-\sigma^2)(1-\kappa_\sigma(e(i)))}{e^2(i)\kappa_\sigma(e(i))}+2\right\}, \tag{12}$$

where $(\lambda(1-\kappa_\sigma(e(i)))^{p/2}+1) \geq 1$. Thus, we have $p \geq \max\limits_{i=1,2,...,N}\left\{\frac{2(e^2(i)-\sigma^2)(1-\kappa_\sigma(e(i)))}{e^2(i)\kappa_\sigma(e(i))}+2\right\}$. □

Remark 2. *According to Property 6, the empirical KRP as a function of* $\mathbf{e}$ *is convex at any point satisfying* $\|\mathbf{e}\|_\infty \leq \sigma$ *and* $p \geq 2$. *For the case* $\|\mathbf{e}\|_\infty > \sigma$, *the empirical KRP can still be convex at a point if the risk-sensitive parameter* $\lambda > 0$ *and power parameter* $p \geq \max\limits_{i=1,2,...,N}\left\{\frac{2(e^2(i)-\sigma^2)(1-\kappa_\sigma(e(i)))}{e^2(i)\kappa_\sigma(e(i))}+2\right\}$.

Property 7. *As* $\sigma \to \infty$ *or* $x(i) \to 0$, $i = 1, 2, ..., N$, *it holds that*

$$\hat{L}_{\lambda,p}(\mathbf{X},\mathbf{0}) \approx \frac{1}{\lambda} + \frac{1}{N\left(\sqrt{2}\sigma\right)^p}\|\mathbf{X}\|_p^p, \tag{13}$$

where $\mathbf{0}$ *denotes an N-dimensional zero vector.*

Proof.

$$\begin{aligned}\hat{L}_{\lambda,p}(\mathbf{X},\mathbf{0}) &= \frac{1}{N\lambda}\sum_{i=1}^{N}\exp(\lambda(1-\kappa_\sigma(x(i)))^{p/2}) \approx \frac{1}{N\lambda}\sum_{i=1}^{N}\exp\left(\lambda\left(1-\left(1-\frac{x^2(i)}{2\sigma^2}\right)\right)^{p/2}\right)\\ &= \frac{1}{N\lambda}\sum_{i=1}^{N}\exp\left(\lambda\left(\frac{x^2(i)}{2\sigma^2}\right)^{p/2}\right) \approx \frac{1}{N\lambda}\sum_{i=1}^{N}\left[1+\lambda\left(\frac{x^2(i)}{2\sigma^2}\right)^{p/2}\right]\\ &= \frac{1}{\lambda}+\frac{1}{N\left(\sqrt{2}\sigma\right)^p}\sum_{i=1}^{N}|x(i)|^p = \frac{1}{\lambda}+\frac{1}{N\left(\sqrt{2}\sigma\right)^p}\|\mathbf{X}\|_p^p.\end{aligned} \tag{14}$$

□

Remark 3. *According to Property 7, the empirical KRP* $\hat{L}_{\lambda,p}(\mathbf{X},\mathbf{0})$ *behaves like an* L_p *norm of* $\mathbf{X}$ *when kernel bandwidth* σ *is large enough.*

3. Application to Adaptive Filtering

In this section, to combat non-Gaussian noises, two recursive robust adaptive algorithms under the proposed KRP criterion are proposed in the RKHS using the kernel trick and vector quantization technique, respectively.

3.1. RMKRP

The recursive strategy is introduced into the KRP loss function, namely the recursive minimum kernel risk-sensitive mean p-power error (RMKRP) algorithm. The offline solution to minimum of the KRP loss is first obtained. Based on the obtained offline solution, the recursive solution or online solution to minimum of the KRP loss is then derived using some matrix operations, which generates the RMKRP algorithm. The details of RMKRP are shown as follows.

Consider the prediction of a continuous input-output model $f : \mathbb{U} \to \mathbb{R}$ based on adaptive filtering shown in Figure 1, where $\mathbf{u}(i) \in \mathbb{U} \subset \mathbb{R}^D$ is the ith D-dimensional input vector, $d(i) \in \mathbb{R}$ is the ith scalar desired output contaminated by a noise $v(i)$, i.e., $d(i) = f(\mathbf{u}(i)) + v(i)$. A sequence of training samples $\{\mathbf{u}(j), d(j)\}_{j=1}^{i}$ is used to perform the prediction of $f(\cdot)$ in an adaptive filter. The nonlinear mapping $\boldsymbol{\varphi}(\mathbf{u}(j))$ of input $\mathbf{u}(j)$ is denoted by $\boldsymbol{\varphi}(j)$ for simplicity. Hence, in the RKHS $\mathbb{F}$, the training samples are changed to $\{\boldsymbol{\Phi}(i), \mathbf{d}(i)\}$, where the desired output vector is $\mathbf{d}(i) = [d(1), d(2), ..., d(i)]^{\mathrm{T}}$ and the input kernel mapping matrix is $\boldsymbol{\Phi}(i) = [\boldsymbol{\varphi}(1), \boldsymbol{\varphi}(2), ..., \boldsymbol{\varphi}(i)]$. The prediction denoted by $\hat{f}(\cdot)$ in the RKHS is therefore given as $\hat{f}(\cdot) = \boldsymbol{\varphi}^{\mathrm{T}}(\cdot)\boldsymbol{\Omega}$, where $\boldsymbol{\Omega} \in \mathbb{F}$ is the weight vector in a high dimensional feature space $\mathbb{F}$.

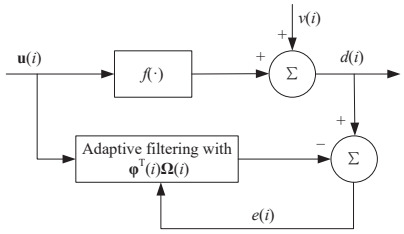

Figure 1. Block diagram of adaptive filtering.

An exponentially-weighted loss function is used here to put more emphasis on recent data and to de-emphasize data on the remote past [28]. When $\{\mathbf{\Phi}(i), \mathbf{d}(i)\}$ are available, the weight vector $\mathbf{\Omega}(i)$ is obtained as the offline solution to minimizing the following weighted cost function:

$$J(\mathbf{\Omega}(i)) = \sum_{j=1}^{i} \rho^{i-j} \frac{1}{\lambda} \exp\left(\lambda(z(j))^{\frac{p}{2}}\right) + \frac{1}{2}\rho^i \zeta \|\mathbf{\Omega}(i)\|^2, \tag{15}$$

where ρ denotes the forgetting factor in the interval $[0, 1]$, ζ is the regularization factor, $z(j) = 1 - \exp\left(-\frac{e^2(j)}{2\sigma^2}\right)$, and $e(j) = d(j) - \boldsymbol{\varphi}^{\mathrm{T}}(j)\mathbf{\Omega}(i)$ denotes the jth estimate error. The second term is a norm penalizing term, which is to guarantee the existence of the inverse of the input data autocorrelation matrix especially during the initial update stages. In addition, the regularization term is weighted by ρ, which deemphasizes regularization as time progresses. According to Property 6, the empirical KRP as a function of $\mathbf{e}$ is convex at any point satisfying $\max_{j=1,2,\ldots,i} |e(j)| \leq \sigma$, $\lambda > 0$, and $p \geq 2$. To obtain the minimum of (15), its gradient is calculated, i.e.,

$$\begin{aligned}
\frac{\partial J(\mathbf{\Omega}(i))}{\partial \mathbf{\Omega}(i)} &= -\frac{p}{2\sigma^2} \sum_{j=1}^{i} \boldsymbol{\varphi}(j)\rho^{i-j} \exp\left(\lambda(z(j))^{\frac{p}{2}}\right)(z(j))^{\frac{p-2}{2}}(d(j) - \boldsymbol{\varphi}^{\mathrm{T}}(j)\mathbf{\Omega}(i))(1 - z(j)) + \rho^i \zeta \mathbf{\Omega}(i) \\
&= -\frac{p}{2\sigma^2} \sum_{j=1}^{i} \boldsymbol{\varphi}(j)\rho^{i-j} \exp\left(\lambda(z(j))^{\frac{p}{2}}\right)(z(j))^{\frac{p-2}{2}}(1 - z(j))d(j) \\
&\quad + \frac{p}{2\sigma^2} \sum_{j=1}^{i} \boldsymbol{\varphi}(j)\rho^{i-j} \exp\left(\lambda(z(j))^{\frac{p}{2}}\right)(z(j))^{\frac{p-2}{2}}(1 - z(j))\boldsymbol{\varphi}^{\mathrm{T}}(j)\mathbf{\Omega}(i) + \rho^i \zeta \mathbf{\Omega}(i).
\end{aligned} \tag{16}$$

Setting (16) to zero, i.e., $\frac{\partial J(\mathbf{\Omega})}{\partial \mathbf{\Omega}} = 0$, we can obtain the offline solution to minimum of (15) as follows:

$$\mathbf{\Omega}(i) = (\mathbf{\Phi}(i)\mathbf{H}(i)\mathbf{\Phi}^{\mathrm{T}}(i) + \rho^i \zeta 2\sigma^2/p\mathbf{I})^{-1}\mathbf{\Phi}(i)\mathbf{H}(i)\mathbf{d}(i), \tag{17}$$

where $\mathbf{H}(i) = \mathrm{diag}[H_1(i), H_2(i), ..., H_i(i)]$ with $H_j(i) = \rho^{i-j} \exp\left(\lambda(z(j))^{\frac{p}{2}}\right)(z(j))^{\frac{p-2}{2}}(1 - z(j))$, $j = 1, 2, ..., i$, and $\mathbf{I}$ denotes an identity matrix with an appropriate dimension.

To obtain an efficient recursive solution to the minimum of (15), a Mercer kernel is used to construct the RKHS. Here, the Gaussian kernel is used as a Mercer kernel, which is denoted as $\kappa_{\sigma_1}(.)$ with σ_1 being the kernel width. The inner product in the RKHS can be calculated by using the kernel trick [28], i.e., $\kappa_{\sigma_1}(\mathbf{u}(i), \mathbf{u}(j)) = \kappa_{\sigma_1}(\mathbf{u}(i) - \mathbf{u}(j)) = \varphi^{\mathrm{T}}(\mathbf{u}(i))\varphi(\mathbf{u}(j)) = \boldsymbol{\varphi}^{\mathrm{T}}(i)\boldsymbol{\varphi}(j)$, efficiently, which can avoid the direct calculation of nonlinear mapping $\boldsymbol{\varphi}(\cdot)$.

Since the matrix inversion lemma [28] is described by $(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1} + \mathbf{DA}^{-1}\mathbf{B})^{-1}\mathbf{DA}^{-1}$, by letting $\mathbf{A} = \rho^i \zeta 2\sigma^2/p\mathbf{I}$, $\mathbf{B} = \mathbf{\Phi}(i)$, $\mathbf{C} = \mathbf{H}(i)$, and $\mathbf{D} = \mathbf{\Phi}^{\mathrm{T}}(i)$, we rewrite (17) as

$$(\mathbf{\Phi}(i)\mathbf{H}(i)\mathbf{\Phi}^{\mathrm{T}}(i) + \rho^i \zeta 2\sigma^2/p\mathbf{I})^{-1}\mathbf{\Phi}(i)\mathbf{H}(i) = \mathbf{\Phi}(i)(\mathbf{\Phi}^{\mathrm{T}}(i)\mathbf{\Phi}(i) + \rho^i \zeta 2\sigma^2/p\mathbf{H}(i)^{-1})^{-1}. \tag{18}$$

Substituting (18) into (17) yields

$$\mathbf{\Omega}(i) = \mathbf{\Phi}(i)(\mathbf{\Phi}^{\mathrm{T}}(i)\mathbf{\Phi}(i) + \rho^i \zeta 2\sigma^2 / p\mathbf{H}(i)^{-1})^{-1}\mathbf{d}(i). \tag{19}$$

Note that $\mathbf{\Phi}^{\mathrm{T}}(i)\mathbf{\Phi}(i)$ in (19) can be computed by the kernel trick, efficiently. The weight vector $\mathbf{\Omega}(i)$ is therefore described explicitly as a linear combination of the input data in the RKHS, i.e.,

$$\mathbf{\Omega}(i) = \mathbf{\Phi}(i)\boldsymbol{\alpha}(i), \tag{20}$$

where $\boldsymbol{\alpha}(i)$ denotes the coefficients vector.

It can be seen from (20) that the recursive form of $\mathbf{\Omega}(i)$ is changed to that of $\boldsymbol{\alpha}(i)$. Hence, in the following, the key for finding a recursive solution to the minimum of (15) is to obtain the recursive form of $\boldsymbol{\alpha}(i)$.

The coefficients vector $\boldsymbol{\alpha}(i)$ is calculated using the kernel trick as

$$\boldsymbol{\alpha}(i) = (\mathbf{\Phi}^{\mathrm{T}}(i)\mathbf{\Phi}(i) + \rho^i \zeta 2\sigma^2 / p\mathbf{H}(i)^{-1})^{-1}\mathbf{d}(i). \tag{21}$$

For simplicity, we obtain the update form of $\boldsymbol{\alpha}(i)$ indirectly by defining $\mathbf{\Lambda}(i)$ as

$$\mathbf{\Lambda}(i) = (\mathbf{\Phi}^{\mathrm{T}}(i)\mathbf{\Phi}(i) + \rho^i \zeta 2\sigma^2 / p\mathbf{H}(i)^{-1})^{-1}, \tag{22}$$

where $\mathbf{\Phi}(i) = \{\mathbf{\Phi}(i-1), \boldsymbol{\varphi}(i)\}$. Then, the update form of $\mathbf{\Lambda}(i)$ can be further obtained

$$\mathbf{\Lambda}(i) = \begin{bmatrix} \mathbf{\Phi}^{\mathrm{T}}(i-1)\mathbf{\Phi}(i-1) + \rho^i \zeta 2\sigma^2 / p\mathbf{H}(i-1)^{-1} & \mathbf{\Phi}^{\mathrm{T}}(i-1)\boldsymbol{\varphi}(i) \\ \boldsymbol{\varphi}^{\mathrm{T}}(i)\mathbf{\Phi}(i-1) & \boldsymbol{\varphi}^{\mathrm{T}}(i)\boldsymbol{\varphi}(i) + \rho^i \zeta 2\sigma^2 / p\nu(i) \end{bmatrix}^{-1}, \tag{23}$$

where $\nu(i) = \left(\exp\left(\lambda(z(i))^{\frac{p}{2}}\right)(z(i))^{\frac{p-2}{2}}(1-z(i))\right)^{-1}$. By using some matrix operations, we further simplify (23) as

$$\mathbf{\Lambda}(i)^{-1} = \begin{bmatrix} \mathbf{\Lambda}(i-1)^{-1} & \boldsymbol{\xi}(i) \\ \boldsymbol{\xi}^{\mathrm{T}}(i) & \rho^i \zeta 2\sigma^2 / p\nu(i) + \boldsymbol{\varphi}^{\mathrm{T}}(i)\boldsymbol{\varphi}(i) \end{bmatrix}, \tag{24}$$

where $\boldsymbol{\xi}(i) = \mathbf{\Phi}^{\mathrm{T}}(i-1)\boldsymbol{\varphi}(i)$. By using the following block matrix inversion identity [18,21,28]

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} = \begin{bmatrix} (\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & -\mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \\ -\mathbf{D}^{-1}\mathbf{C}(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & (\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \end{bmatrix}, \tag{25}$$

then, we can obtain the update equation for the inverse of the growing matrix in (24) as

$$\mathbf{\Lambda}(i) = r^{-1}(i) \begin{bmatrix} \mathbf{\Lambda}(i-1)r(i) + \boldsymbol{\theta}(i)\boldsymbol{\theta}^{\mathrm{T}}(i) & -\boldsymbol{\theta}(i) \\ -\boldsymbol{\theta}^{\mathrm{T}}(i) & 1 \end{bmatrix}, \tag{26}$$

where $\boldsymbol{\theta}(i) = \mathbf{\Lambda}(i-1)\boldsymbol{\xi}(i)$ and $r(i) = \rho^i \zeta 2\sigma^2 / p\nu(i) + \boldsymbol{\varphi}^{\mathrm{T}}(i)\boldsymbol{\varphi}(i) - \boldsymbol{\theta}^{\mathrm{T}}(i)\boldsymbol{\xi}(i)$. Combining (21) with (26), the coefficients vector $\boldsymbol{\alpha}(i)$ of the weight vector $\mathbf{\Omega}(i)$ is shown as follows:

$$\begin{aligned} \boldsymbol{\alpha}(i) = \mathbf{\Lambda}(i)\mathbf{d}(i) &= \begin{bmatrix} \mathbf{\Lambda}(i-1) + \boldsymbol{\theta}(i)\boldsymbol{\theta}^{\mathrm{T}}(i)r^{-1}(i) & -\boldsymbol{\theta}(i)r^{-1}(i) \\ -\boldsymbol{\theta}^{\mathrm{T}}(i)r^{-1}(i) & r^{-1}(i) \end{bmatrix} \begin{bmatrix} \mathbf{d}(i-1) \\ d(i) \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{\alpha}(i-1) - \boldsymbol{\theta}(i)r^{-1}(i)e(i) \\ r^{-1}(i)e(i) \end{bmatrix}, \end{aligned} \tag{27}$$

where $e(i) = d(i) - \hat{f}(i)$ denotes the difference between the desired output $d(i)$ and the system output $\hat{f}(i) = \boldsymbol{\xi}(i)^{\mathrm{T}}\boldsymbol{\alpha}(i-1) = \sum_{j=1}^{i-1} \boldsymbol{\alpha}_j(i-1)\kappa_{\sigma_1}(\mathbf{u}(j), \mathbf{u}(i))$. $\boldsymbol{\alpha}_j(i-1)$ is the jth element of $\boldsymbol{\alpha}(i-1)$ and all the previous data are the centers. The coefficients $\boldsymbol{\alpha}(i-1)$ and all the previous data should be stored at each iteration. Finally, the RMKRP algorithm is summarized in Algorithm 1.

Algorithm 1: The RMKRP Algorithm.

Initialization:

$\{\mathbf{u}(i), d(i)\}, i = 1, 2, ...$

$p, \lambda, \rho, \sigma, \sigma_1 > 0, \zeta \in [0, 1], z(1) = 1 - \exp\left(-\frac{d^2(1)}{2\sigma^2}\right)$.

$H_1(1) = \exp\left(\lambda(z(1))^{\frac{p}{2}}\right)(z(1))^{\frac{p-2}{2}}(1 - z(1))$.

$\boldsymbol{\Lambda}(1) = (\zeta\rho 2\sigma^2/p/H_1(1) + \kappa(\mathbf{u}(1), \mathbf{u}(1)))^{-1}, \boldsymbol{\alpha}(1) = \boldsymbol{\Lambda}(1)d(1)$.

Computation:

While $\{\mathbf{u}(i), d(i)\}(i > 1)$ available **do**

1) $\boldsymbol{\xi}(i) = [\kappa(\mathbf{u}(1), \mathbf{u}(1)), ..., \kappa(\mathbf{u}(i), \mathbf{u}(i-1))]^{\mathrm{T}}$

2) $e(i) = d(i) - \boldsymbol{\xi}(i)^{\mathrm{T}}\boldsymbol{\alpha}(i-1)$

3) $\boldsymbol{\theta}(i) = \boldsymbol{\Lambda}(i-1)\boldsymbol{\xi}(i)$

4) $z(i) = 1 - \exp\left(-\frac{e^2(i)}{2\sigma^2}\right)$

5) $\nu(i) = \left(\exp\left(\lambda(z(i))^{\frac{p}{2}}\right)(z(i))^{\frac{p-2}{2}}(1 - z(i))\right)^{-1}$

6) $r(i) = \rho^i\zeta 2\sigma^2/p\nu(i) + \kappa(\mathbf{u}(i), \mathbf{u}(i)) - \boldsymbol{\theta}^{\mathrm{T}}(i)\boldsymbol{\xi}(i)$

7) $\boldsymbol{\Lambda}(i) = r^{-1}(i)\begin{bmatrix} \boldsymbol{\Lambda}(i-1)r(i) + \boldsymbol{\theta}(i)\boldsymbol{\theta}^{\mathrm{T}}(i) & -\boldsymbol{\theta}(i) \\ -\boldsymbol{\theta}^{\mathrm{T}}(i) & 1 \end{bmatrix}$

8) $\boldsymbol{\alpha}(i) = \begin{bmatrix} \boldsymbol{\alpha}(i-1) - \boldsymbol{\theta}(i)r^{-1}(i)e(i) \\ r^{-1}(i)e(i) \end{bmatrix}$.

end while

3.2. QRMKRP

The RMKRP algorithm generates a linearly growing network owing to the used kernel trick. The online vector quantization (VQ) method [12] has been successfully applied in KAFs to curb its network growth efficiently. Thus, we incorporate the online VQ method into the RMKRP to develop the quantized RMKRP (QRMKRP) algorithm, which is shown as follows.

Suppose that the dictionary $\mathbf{C}(i)$ contains L vectors at discrete time i, i.e., $\mathbf{C}(i) = \{\mathbf{C}_k(i)\}_{k=1}^{L}$, $k \in Id = \{1, 2, ..., L\}$, which means that there are L distinctive quantization regions. In the RKHS, the prediction $\hat{f}(i)$ is therefore expressed as $\hat{f}(i) = \varphi^{\mathrm{T}}(\mathbf{C}_k(i))\hat{\boldsymbol{\Omega}}$, where $\hat{\boldsymbol{\Omega}} \in \mathbb{F}$ is the weight vector in RKHS $\mathbb{F}$. The cost function of QRMKRP based on $\mathbf{C}(i)$ is denoted as

$$\sum_{k=1}^{L}\sum_{n=1}^{|\mathrm{D}_k|}\rho^{i-k}\frac{1}{\lambda}\exp\left(\lambda\left(1 - \exp\left(-\frac{(d_{kn}(i) - \varphi^{\mathrm{T}}(\mathbf{C}_k(i))\hat{\boldsymbol{\Omega}}(i))^2}{2\sigma^2}\right)\right)^{\frac{p}{2}}\right) + \frac{1}{2}\rho^i\zeta\left\|\hat{\boldsymbol{\Omega}}(i)\right\|^2, \quad (28)$$

where $|\mathrm{D}_k|$ denotes the number of input data those lie in the kth quantization region of $\mathbf{C}(i)$ and satisfies $\sum_{k\in Id}|\mathrm{D}_k| = i$ and $|\mathrm{D}_k| \geq 1$, and $d_{kn}(i)$ is the desired output $d(i)$ corresponding to the nth element within the kth quantization region.

The offline solution to the minimization of (28) can be described by

$$\hat{\boldsymbol{\Omega}}(i) = \left[\hat{\boldsymbol{\Phi}}(i)\hat{\mathbf{H}}(i)\hat{\boldsymbol{\Phi}}^{\mathrm{T}}(i) + \rho^i\zeta 2\sigma^2/p\mathbf{I}\right]^{-1}\hat{\boldsymbol{\Phi}}(i)\hat{\mathbf{d}}(i), \quad (29)$$

where $\hat{\boldsymbol{\Phi}}(i) = [\varphi(\mathbf{C}_1(i)), \varphi(\mathbf{C}_2(i)), ..., \varphi(\mathbf{C}_L(i))]$ with $L \ll i$ elements; $\hat{\mathbf{H}}(i) = \mathrm{diag}[\sum_{n=1}^{|D_1|}H_{1n}(i), \sum_{n=1}^{|D_2|}H_{2n}(i), ..., \sum_{n=1}^{|D_L|}H_{Ln}(i)]$ denotes a accumulated diagonal matrix;

$\hat{\mathbf{d}}(i) = \mathrm{diag}[\sum_{n=1}^{|D_1|} H_{1n}(i)d_{1n}(i), \sum_{n=1}^{|D_2|} H_{2n}(i)d_{2n}(i), ..., \sum_{n=1}^{|D_L|} H_{Ln}(i)d_{Ln}(i)]^{\mathrm{T}}$ denotes a accumulated weighted output vector; $H_{kn}(i)$ denotes $H_i(i)$ corresponding to the nth entry of the kth quantization region; $\mathbf{I}$ denotes an identity matrix with an appropriate dimension. Since (29) has a similar form to (17), we simplify (29) as

$$\hat{\mathbf{\Omega}}(i) = \hat{\mathbf{\Phi}}(i)\left[\hat{\mathbf{H}}(i)\hat{\mathbf{K}}(i) + \rho^i \zeta 2\sigma^2/p\mathbf{I}\right]^{-1}\hat{\mathbf{d}}(i) = \hat{\mathbf{\Phi}}(i)\hat{\mathbf{Q}}(i)\hat{\mathbf{d}}(i) = \hat{\mathbf{\Phi}}(i)\hat{\boldsymbol{\alpha}}(i), \tag{30}$$

where $\hat{\mathbf{K}}(i) = \hat{\mathbf{\Phi}}^{\mathrm{T}}(i)\hat{\mathbf{\Phi}}(i)$. To obtain the recursive solution to the minimization of (28), we let $\hat{\boldsymbol{\alpha}}(i) = \hat{\mathbf{Q}}(i)\hat{\mathbf{d}}(i)$ and denote

$$\hat{\mathbf{Q}}(i) = \left[\hat{\mathbf{H}}(i)\hat{\mathbf{K}}(i) + \rho^i \zeta 2\sigma^2/p\mathbf{I}\right]^{-1}. \tag{31}$$

To update $\hat{\mathbf{\Omega}}(i)$ in (30) recursively, two cases are therefore considered.

(1) First, Case: $\mathrm{dis}(\mathbf{u}(i), \mathbf{C}(i-1)) \leq \epsilon$: In this case, we have $\mathbf{C}(i) = \mathbf{C}(i-1)$ and $\hat{\mathbf{Q}}(i) = \hat{\mathbf{Q}}(i-1)$, which means the input $\mathbf{u}(i)$ is therefore quantized to the k^*th element of dictionary $\mathbf{C}(i-1)$, where $k^* = \underset{1\leq k\leq|\mathbf{C}_k(i-1)|}{\arg\min} \|\mathbf{u}(i) - \mathbf{C}_k(i-1)\|^2$. The matrix $\hat{\mathbf{H}}(i)$ and the vector $\hat{\mathbf{d}}(i)$ have a similar form to [13]. Here, $\hat{\mathbf{H}}(i)$ and $\hat{\mathbf{d}}(i)$ can be shown as

$$\begin{cases} \hat{\mathbf{H}}(i) = \hat{\mathbf{H}}(i-1) + H_i(i)\boldsymbol{\tau}_{k^*}\boldsymbol{\tau}_{k^*}^{\mathrm{T}} \\ \hat{\mathbf{d}}(i) = \hat{\mathbf{d}}(i-1) + H_i(i)d(i)\boldsymbol{\tau}_{k^*} \end{cases}, \tag{32}$$

where $\boldsymbol{\tau}_{k^*}$ is a $|\mathbf{C}(i-1)|$-dimensional column vector whose k^*th element is 1 and all other elements are 0. Combining (32) into (31), the matrix $\hat{\mathbf{Q}}(i)$ can be expressed as $\hat{\mathbf{Q}}(i) = [\hat{\mathbf{Q}}(i-1)^{-1} + H_i(i)\boldsymbol{\tau}_{k^*}\boldsymbol{\tau}_{k^*}^{\mathrm{T}}\hat{\mathbf{K}}(i-1)]^{-1}$. By using the matrix inversion lemma [28], we obtain

$$\hat{\mathbf{Q}}(i) = \hat{\mathbf{Q}}(i-1) - \frac{\hat{\mathbf{Q}}_{k^*}(i-1)\hat{\mathbf{K}}_{k^*}^{\mathrm{T}}(i-1)\hat{\mathbf{Q}}(i-1)}{H_i^{-1}(i) + \hat{\mathbf{K}}_{k^*}^{\mathrm{T}}(i-1)\hat{\mathbf{Q}}_{k^*}(i-1)}, \tag{33}$$

where $\hat{\mathbf{Q}}_{k^*}(i-1)$ and $\hat{\mathbf{K}}_{k^*}(i-1)$ represent the k^*th columns of the matrices $\hat{\mathbf{Q}}(i-1)$ and $\hat{\mathbf{K}}(i-1)$, respectively. Therefore, $\hat{\boldsymbol{\alpha}}(i)$ in (30) can be calculated as

$$\hat{\boldsymbol{\alpha}}(i) = \hat{\mathbf{Q}}(i)\hat{\mathbf{d}}(i) = \hat{\boldsymbol{\alpha}}(i-1) + \frac{(d(i) - \hat{\mathbf{K}}_{k^*}^{\mathrm{T}}(i-1)\hat{\boldsymbol{\alpha}}(i-1))\hat{\mathbf{Q}}_{k^*}(i-1)}{H_i^{-1}(i) + \hat{\mathbf{K}}_{k^*}^{\mathrm{T}}(i-1)\hat{\mathbf{Q}}_{k^*}(i-1)}. \tag{34}$$

(2) Second Case: $\mathrm{dis}(\mathbf{u}(i), \mathbf{C}(i-1)) > \epsilon$: In this case, we have $\mathbf{C}(i) = \{\mathbf{C}(i-1), \mathbf{u}(i)\}$, $\hat{\mathbf{\Phi}}(i) = [\hat{\mathbf{\Phi}}(i-1), \varphi(\mathbf{u}(i))]$, and we have

$$\hat{\mathbf{H}}(i) = \begin{bmatrix} \hat{\mathbf{H}}(i-1) & \mathbf{0} \\ \mathbf{0}^{\mathrm{T}} & H_i(i) \end{bmatrix}, \ \hat{\mathbf{K}}(i) = \begin{bmatrix} \hat{\mathbf{K}}(i-1) & \hat{\mathbf{h}}(i) \\ \hat{\mathbf{h}}(i)^{\mathrm{T}} & \kappa_{ii} \end{bmatrix}, \tag{35}$$

where $\mathbf{0}$ is the null column vector with a compatible dimension; $\hat{\mathbf{h}}(i) = \hat{\mathbf{\Phi}}(i-1)^{\mathrm{T}}\varphi(\mathbf{u}(i))$ and $\kappa_{ii} = \kappa_{\sigma_1}(\mathbf{u}(i), \mathbf{u}(i))$. Combining (31), (35), $\hat{\mathbf{d}}(i) = [\hat{\mathbf{d}}(i-1)^{\mathrm{T}}, H_i(i)d(i)]^{\mathrm{T}}$, and the block matrix inversion identity [28], we obtain

$$\hat{\mathbf{Q}}(i) = \begin{bmatrix} \hat{\mathbf{Q}}(i-1) + \hat{r}(i)^{-1}H_i(i)\hat{\mathbf{z}}_{\hat{\mathbf{H}}(i)}\hat{\mathbf{z}}(i)^{\mathrm{T}} & -\hat{r}(i)^{-1}\hat{\mathbf{z}}_{\hat{\mathbf{H}}(i)} \\ -\hat{r}(i)^{-1}H_i(i)\hat{\mathbf{z}}(i)^{\mathrm{T}} & \hat{r}(i)^{-1} \end{bmatrix}, \tag{36}$$

where

$$\begin{cases} \hat{\mathbf{z}}_{\hat{\mathbf{H}}(i)} = \hat{\mathbf{Q}}(i-1)\hat{\mathbf{H}}(i-1)\hat{\mathbf{h}}(i) \\ \hat{\mathbf{z}}(i) = \hat{\mathbf{Q}}(i-1)^{\mathrm{T}}\hat{\mathbf{h}}(i) \\ \hat{r}(i) = \kappa_{\sigma_1}(\mathbf{u}(i),\mathbf{u}(i))H_i(i) + \rho^i\zeta 2\sigma^2/p - H_i(i)\hat{\mathbf{h}}(i)^{\mathrm{T}}\hat{\mathbf{z}}_{\hat{\mathbf{H}}(i)} \end{cases}. \tag{37}$$

Furthermore, due to $\hat{\mathbf{d}}(i) = [\hat{\mathbf{d}}(i-1)^{\mathrm{T}}, H_i(i)d(i)]^{\mathrm{T}}$, we obtain

$$\hat{\boldsymbol{\alpha}}(i) = \hat{\mathbf{Q}}(i)\hat{\mathbf{d}}(i) = \begin{bmatrix} \hat{\boldsymbol{\alpha}}(i-1) - \hat{r}(i)^{-1}H_i(i)\hat{\mathbf{z}}_{\hat{\mathbf{H}}(i)}(d(i) - \hat{\mathbf{h}}(i)^{\mathrm{T}}\hat{\boldsymbol{\alpha}}(i-1)) \\ \hat{r}(i)^{-1}H_i(i)(d(i) - \hat{\mathbf{h}}(i)^{\mathrm{T}}\hat{\boldsymbol{\alpha}}(i-1)) \end{bmatrix}. \tag{38}$$

The QRMKRP algorithm is summarized in Algorithm 2, where L denotes the dictionary size.

Algorithm 2: The QRMKRP algorithm.

Initialization:
$\{\mathbf{u}(i), d(i)\}, i = 1, 2, ...$
$\sigma, \sigma_1, p, \lambda > 0, L = 1, \mathbf{C}(1) = \{\mathbf{u}(1)\}$,
$z(1) = 1 - \exp\left(-\frac{d^2(1)}{2\sigma^2}\right)$,
$H_1(1) = \exp\left(\lambda(z(1))^{\frac{p}{2}}\right)(z(1))^{\frac{p-2}{2}}(1 - z(1)), \hat{\mathbf{H}}(1) = [H_1(1)]$.
$\hat{\mathbf{Q}}(1) = \left[\zeta\rho 2\sigma^2/p + H_1(1)\kappa_{11}\right]^{-1}, \hat{\boldsymbol{\alpha}}(1) = \hat{\mathbf{Q}}(1)H_1(1)d(1)$,
$\epsilon > 0, \rho > 0, \zeta \in [0, 1]$.
Computation:
While $\{\mathbf{u}(i), d(i)\}(i > 1)$ available **do**
1) Compute the distance between $\mathbf{u}(i)$ and $\mathbf{C}(i-1)$:
$\mathrm{dis}(\mathbf{u}(i), \mathbf{C}(i-1)) = \min\limits_{1 \le k \le |\mathbf{C}_k(i-1)|} \|\mathbf{u}(i) - \mathbf{C}_k(i-1)\|^2$,
where $k^* = \arg\min\limits_{1 \le k \le |\mathbf{C}_k(i-1)|} \|\mathbf{u}(i) - \mathbf{C}_k(i-1)\|^2$.
2) If $\mathrm{dis}(\mathbf{u}(i), \mathbf{C}(i-1)) \le \epsilon$:
Keep the dictionary unchanged: $\mathbf{C}(i) = \mathbf{C}(i-1), L \Leftarrow L$,
Update $\hat{\mathbf{H}}(i)$ by (32), $\hat{\mathbf{Q}}(i)$ by (33), $\hat{\boldsymbol{\alpha}}(i)$ by (34).
3) Otherwise:
The dictionary changes: $\mathbf{C}(i) = [\mathbf{C}(i-1), \mathbf{u}(i)], L \Leftarrow L + 1$,
Update $\hat{\mathbf{H}}(i)$ by (35), $\hat{\mathbf{Q}}(i)$ by (36), $\hat{\boldsymbol{\alpha}}(i)$ by (38).
end while

4. Simulation

In this section, to validate the performance of the proposed RMKRP algorithm and its quantized version, two examples, i.e., Mackey–Glass (MG) chaotic time series prediction and nonlinear system identification, are used to validate the performance superiorities of the proposed two algorithms.

In this example, the noise environment considered is the impulsive noise, which is modeled by the combination of two independent noise processes [17], i.e.,

$$v(i) = (1 - b(i))v_1(i) + b(i)v_2(i), \tag{39}$$

where $v_1(i)$ is an ordinary noise disturbance with small variance and $v_2(i)$ represents large outliers with large variance; $b(i)$ is of binary distribution random process over $\{0, 1\}$ with $Prob\{b(i) = 1\} = c$ and $Prob\{b(i) = 0\} = 1 - c$ ($0 \le c \le 1$ is an occurrence probability). Here, we select $c = 0.1$. The distribution of $v_1(i)$ is considered as a Binary distribution over $\{0.5, -0.5\}$ with probability mass $Prob\{x = 0.5\} = Prob\{x = -0.5\} = 0.5$. In addition, $v_2(i)$ is modeled by the α-stable process,

owing to its heavy-tailed probability density function. The α-stable process is described by the following characteristic function [29]:

$$f(t) = \exp\left\{j\delta t - \gamma|t|^{\alpha}\left[1 + j\beta \mathrm{sgn}(t) S(t,\alpha)\right]\right\}, \tag{40}$$

where

$$S(t,\alpha) = \begin{cases} \tan\frac{\alpha\pi}{2}, \text{if } \alpha \neq 1 \\ \frac{2}{\pi}\log|t|, \text{if } \alpha = 1 \end{cases}, \tag{41}$$

with $\alpha \in (0,2]$ being the characteristic factor, $\beta \in [-1,1]$ being the symmetry parameter, $\gamma > 0$ being the dispersion parameter, sgn(.) denotes the sign function, $j = \sqrt{-1}$, and $-\infty < \delta < \infty$ being the location parameter. Generally, a smaller α generates a heavier tail and a smaller γ generates fewer large outliers. The characteristic function denoted as $V_{\alpha-\text{stable}}(\alpha,\beta,\gamma,\delta)$ is chosen as $V_{\alpha-\text{stable}}(0.8,0,0.1,0)$ to model the impulse noise in the simulations.

4.1. Chaotic Time Series Prediction

The MG chaotic time series is generated from the following differential equation [9]:

$$\frac{dx(t)}{dt} = \frac{\beta x(t-\tau)}{1 + x(t-\tau)^{n}} - \gamma x(t), \tag{42}$$

where $\beta, \gamma, n > 0$. Here, we set $\beta = 0.2$, $\gamma = 0.1$, and $\tau = 30$. The time series is discretized at a sampling period of six seconds. The training set includes a segment of 2000 samples corrupted by the additive noises which are shown in (39), and another 200 samples without noise are used as the testing set. The kernel size σ_1 in the Gaussian kernel is set to 1. The filter length is set at $L = 7$, which means that $[x_t, x_{t-1}, ..., x_{t-6}]$ is used to predict x_{t+1}.

To evaluate the filtering accuracy, the testing mean square error (MSE) is defined as follows:

$$\mathrm{MSE(dB)} = \frac{1}{N} 10\log_{10}\left(\sum_{i=1}^{N}\left(d(i) - \hat{f}(i)\right)^{2}\right), \tag{43}$$

where $\hat{f}(i)$ is the estimate of $d(i)$, and N is the length of testing data.

The KLMS [6], KMCC [22], MKRL [26], KRMC [21], and KRLS [8] algorithms are chosen for performance comparison with RMKRP thanks to their excellent filtering performance. The other sparsification algorithms, i.e., the QKLMS [12], QKMCC [30], QMKRL [26], QKRLS [13], and KRMC-NC [21] algorithms are used for performance comparison with QRMKRP owing to their modest space complexities and excellent performance. All simulation results are averaged over 50 independent Monte Carlo runs.

Since power parameter p, risk-sensitive parameter λ, and kernel width σ are crucial parameters in the proposed RMKRP and QRMKRP algorithms, the influence of these parameters on the performance is first discussed. In the simulations, we take 12 points evenly in the close interval $p \in [1,6]$ and $\sigma \in [0.17,5]$, respectively. The influence of p on the steady-state performance of RMKRP is shown in Figure 2a, where the steady-state MSEs are obtained as averages over the last 100 iterations. The parameters are set as: p is set within $[1,6]$; risk-sensitive parameter λ in the KRP is set as 1; $\zeta = 0.1$ and $\rho = 1$; kernel size σ in the KRP is set as 1. As can be seen from Figure 2a, we have that the filtering accuracy of RMKRP is the highest when $p = 4$ and decreases gradually when p is either too small or too large. Then, the influence of σ on the filtering performance of RMKRP with $p = 4$ is shown in Figure 2b, where the steady-state MSEs are obtained as averages over the last 100 iterations. The parameters are set as: risk-sensitive parameter λ is fixed at 1; σ lies in $[0.17,5]$. From Figure 2b, we see that RMKRP can achieve the highest filtering accuracy when σ is about 1. It is reasonable to note that RMKRP are sensitive to outliers when the kernel width is large, and decreases

its ability of error-correction when the kernel width is small. Finally, the influence of λ on the filtering performance of RMKRP with $\sigma = 1$ and $p = 4$ is shown in Figure 2c, where the steady-state MSEs are obtained as averages over the last 100 iterations. The parameters are set as: the range of λ is selected as $\lambda \in \{0.001, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 1, 2, 3, 4\}$. From Figure 2c, we see that λ has a slight influence on the filtering accuracy when λ is small, and a large λ can increase the steady-state MSE obviously. Therefore, from Figure 2, the parameters of RMKRP can be chosen by trials to obtain the best performance in practice. Similarly, the parameters of QRMKRP can be chosen by the same method as that in RMKRP.

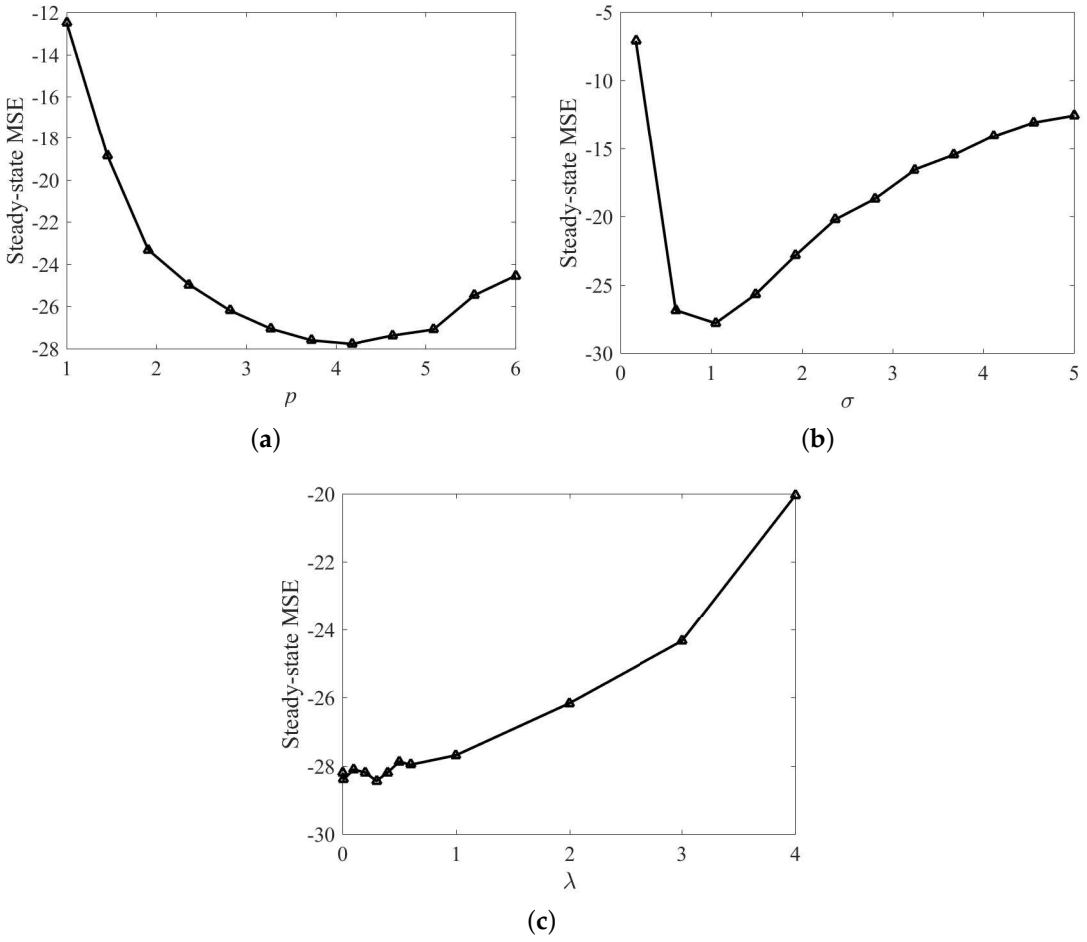

Figure 2. Steady-state MSE of RMKRP with different p in MG time series prediction (**a**); steady-state MSE of RMKRP with different σ in MG time series prediction (**b**); steady-state MSE of RMKRP with different λ in MG time series prediction (**c**).

The performance comparison of QKLMS, QKMCC, QMKRL, KLMS, KMCC, MKRL, KRLS, KRMC, KRMC-NC, and QKRLS is conducted in the same environments as in (39). The parameters of the proposed algorithms are selected by trials to achieve desirable performance, and the parameters of compared algorithms are chosen such that they have almost the same convergence rate. $\lambda = 1$, $p = 4$, and $\sigma = 1$ are set for RMKRP; $\lambda = 1$, $p = 4$, $\sigma = 1$, and $\epsilon = 0.2$ for QRMKRP; $\eta = 0.1$ for KLMS; $\eta = 0.09$ and $\sigma = 3.5$ for KMCC; $\eta = 0.09$, $\sigma = 1$, and $\lambda = 1$ for MKRL; $\eta = 0.1$ and $\epsilon = 0.2$ for QKLMS; $\eta = 0.09$, $\epsilon = 0.2$, and $\sigma = 3.5$ for QKMCC; $\eta = 0.09$, $\epsilon = 0.2$, $\sigma = 1$, and $\lambda = 1$ for QMKRL; $\zeta = 0.1$, $\rho = 1$, and $\sigma = 3.5$ for KRMC; the novelty criterion thresholds $\delta_1 = 0.15$, $\delta_2 = 0.1$, $\zeta = 0.1$, $\rho = 1$, and $\sigma = 3.5$ for KRMC-NC; $\zeta = 0.1$ for KRLS; $\zeta = 0.1$ and $\epsilon = 0.2$ for QKRLS. Figure 3 shows the compared MSEs of RMKRP, QRMKRP, and the compared algorithms. As can be seen from Figure 3, RMKRP achieves a better filtering accuracy than KRLS, KRMC, KLMS, KMCC, and MKRL. QRMKRP achieves

a better steady-state testing MSE than the sparsification algorithms including QKRLS, KRMC-NC, QKLMS, QKMCC, and QMKRL. We also see from Figure 3 that the proposed algorithms provide good robustness to impulsive noises. For detailed comparison, the dictionary size, consumed time, and steady-state MSEs in Figure 3 are shown in Table 1. Note that the steady-state MSEs of KLMS, QKLMS, KRLS, and QKRLS are not shown in Table 1 since they cannot converge in such impulsive noise environment. From Table 1, we see that RMKRP has similar consumed time to KRLS and KRMC but provides better filtering accuracy. In addition, QRMKRP provides the highest filtering accuracy in all the compared sparsification algorithms and approaches the filtering accuracy of RMKRP with a significantly lower network size.

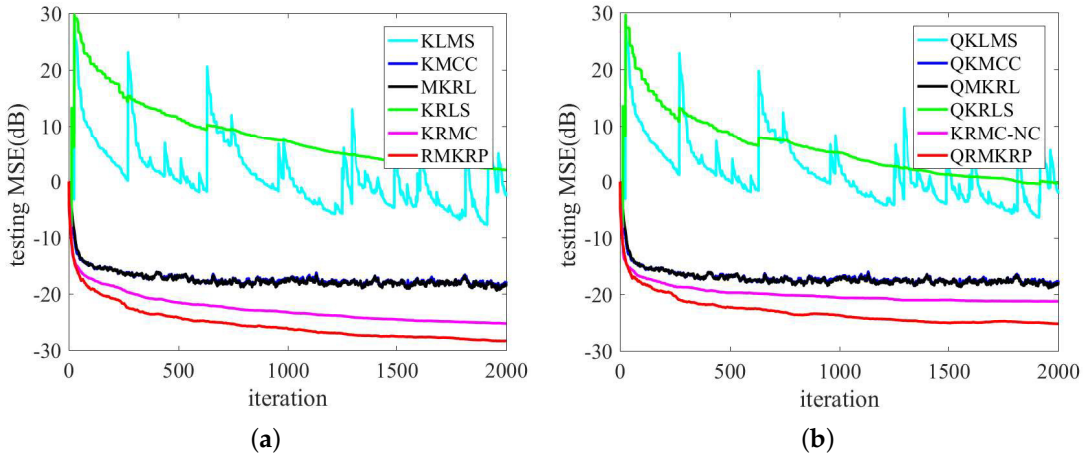

Figure 3. Comparison of the MSEs of KLMS, KMCC, MKRL, KRLS, KRMC, and RMKRP in MG time series prediction (**a**); comparison of the MSEs of QKLMS, QKMCC, QMKRL, QKRLS, KRMC-NC, and QRMKRP in MG time series prediction (**b**).

Table 1. Simulation results of QKLMS, QKMCC, QMKRL, QKRLS, KRMC-NC, KLMS, KMCC, MKRL, KRLS, KRMC, RMKRP, and QRMKRP in MG time series prediction.

Algorithms	Size	Time (s)	MSE (dB)
KLMS [6]	2000	30.9501 s	N/A
QKLMS [12]	28	2.1011 s	N/A
KRLS [8]	2000	58.5358 s	N/A
QKRLS [13]	28	2.3374 s	N/A
KMCC [22]	2000	30.8285 s	−18.5063
QKMCC [30]	28	2.0995 s	−17.8707
MKRL [26]	2000	30.9117 s	−18.7312
QMKRL [26]	28	2.1063 s	−18.1037
KRMC [21]	2000	58.1229 s	−25.1618
KRMC-NC [21]	462	2.8045 s	−21.5183
QRMKRP	28	2.3443 s	−24.9326
RMKRP	2000	58.2196 s	−28.1802

4.2. Nonlinear System Identification

To further validate the performance superiorities of the proposed RMKRP and QRMKRP algorithms, the nonlinear system identification is considered. Here, the nonlinear system is of the following form [31].

$$\mathrm{s}(t) = s(t-1)(0.8 - 0.5\exp(-s^2(t-1))) - (0.3 + 0.9\exp(-s^2(t-1)))s(t-2) + 0.1\sin(s(t-1)\pi), \quad (44)$$

where $\mathrm{s}(t)$ denotes the output at discrete time t with the initial $s(-1) = 0.1$ and $s(-2) = 0.1$. The two previous outputs $\mathbf{u}(k) = [s(t-1), s(t-2)]^{\mathrm{T}}$ are utilized as the input to estimate the current output $\mathrm{s}(t)$.

The training set includes a segment of 2000 samples corrupted by the additive noises shown in (39), and another 200 samples without noise are used as the testing set. The kernel width σ_1 is set to 1 for the Gaussian function. All simulation results are averaged over 50 independent Monte Carlo runs.

Similar to MG chaotic time series prediction, the influence of power parameter p, risk-sensitive parameter λ, and kernel width σ on the performance of RMKRP is also discussed in nonlinear system identification. The influence of p on the steady-state performance of RMKRP is shown in Figure 4a, where the steady-state MSEs are obtained as averages over the last 100 iterations. The parameters are set as: p is set within $[1, 6]$; λ is set as 0.1; $\zeta = 0.1$ and $\rho = 1$; kernel size σ in the KRP is set as 1. The influence of σ on the filtering performance of RMKRP is shown in Figure 4b, where risk-sensitive parameter λ is fixed at 0.1; σ lies in $[0.17, 5]$; p is set as 4. The influence of λ on the filtering performance of RMKRP is shown in Figure 4c, where the range of λ is selected as $\lambda \in \{0.001, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 1, 2, 3, 4\}$; σ is set as 1; p is set as 4. As can be seen from Figure 4, we can obtain the same conclusions as those in Figure 2.

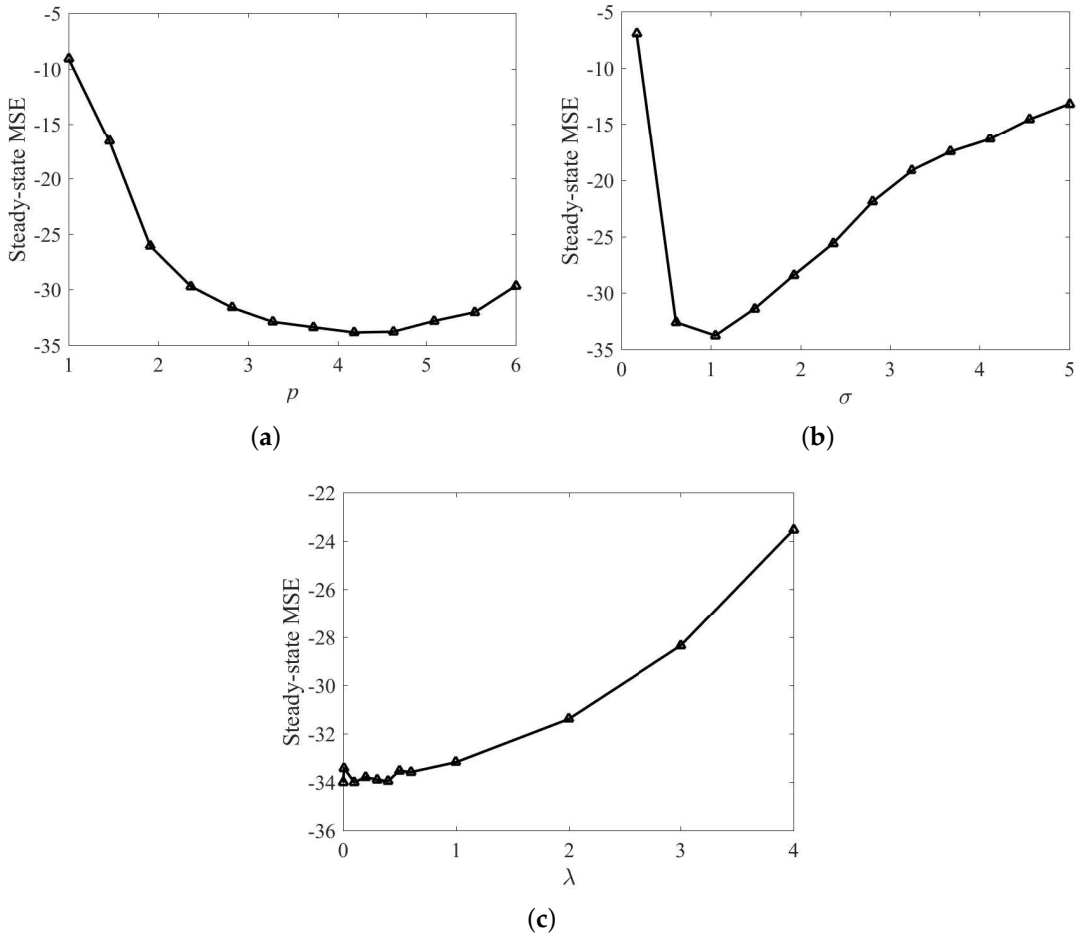

Figure 4. Steady-state MSE of RMKRP with different p in nonlinear system identification (**a**); steady-state MSE of RMKRP with different σ in nonlinear system identification (**b**); steady-state MSE of RMKRP with different λ in nonlinear system identification (**c**).

We compare the filtering performance of QKLMS, QKMCC, QMKRL, KLMS, KMCC, MKRL, KRLS, KRMC, KRMC-NC, and QKRLS in the same environments as in (39). The parameters of the proposed algorithms are selected by trials to achieve desirable performance, and the parameters of compared algorithms are chosen such that they have almost the same convergence rate. $\lambda = 0.1$, $p = 4$, and $\sigma = 1$ are set for RMKRP; $\lambda = 0.1$, $p = 4$, $\sigma = 1$, and $\epsilon = 0.2$ for QRMKRP; $\eta = 0.1$ for KLMS; $\eta = 0.09$ and $\sigma = 3.5$ for KMCC; $\eta = 0.09$, $\sigma = 1$, and $\lambda = 2$ for MKRL; $\eta = 0.1$ and $\epsilon = 0.2$ for QKLMS; $\eta = 0.09$, $\epsilon = 0.2$, and $\sigma = 3.5$ for QKMCC; $\eta = 0.09$, $\epsilon = 0.2$, $\sigma = 1$, and $\lambda = 2$ for QMKRL; $\zeta = 0.1$, $\rho = 1$, and $\sigma = 3.5$

for KRMC; the novelty criterion thresholds $\delta_1 = 0.01$, $\delta_2 = 0.1$, $\zeta = 0.1$, $\rho = 1$, and $\sigma = 3.5$ for KRMC-NC; $\zeta = 0.1$ for KRLS; $\zeta = 0.1$ and $\epsilon = 0.2$ for QKRLS. Figure 5 shows the compared MSEs of RMKRP, QRMKRP, and the compared algorithms. For detailed comparison, the dictionary size, consumed time, and steady-state MSEs in Figure 5 are also shown in Table 2, where the steady-state MSEs of KLMS, QKLMS, KRLS, and QKRLS are not shown since they cannot converge in such impulsive noise environments. From Figure 5 and Table 2, we can obtain the same conclusions as those in Figure 3 and Table 1.

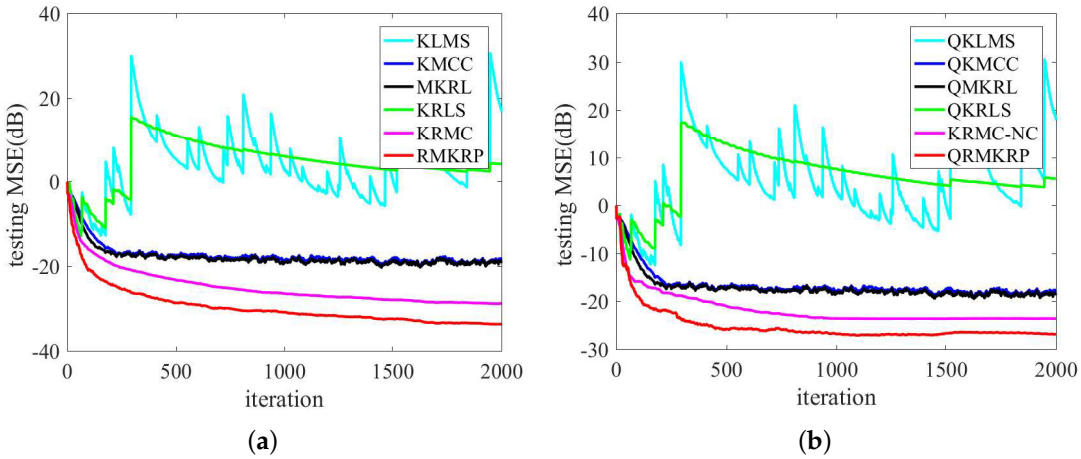

Figure 5. Comparison of the MSEs of KLMS, KMCC, MKRL, KRLS, KRMC, and RMKRP in nonlinear system identification (**a**); comparison of the MSEs of QKLMS, QKMCC, QMKRL, QKRLS, KRMC-NC, and QRMKRP nonlinear system identification (**b**).

Table 2. Simulation results of QKLMS, QKMCC, QMKRL, QKRLS, KRMC-NC, KLMS, KMCC, MKRL, KRLS, KRMC, RMKRP, and QRMKRP in nonlinear system identification.

Algorithms	Size	Time (s)	MSE (dB)
KLMS [6]	2000	21.2447 s	N/A
QKLMS [12]	14	1.7284 s	N/A
KRLS [8]	2000	48.6055 s	N/A
QKRLS [13]	14	1.9643 s	N/A
KMCC [22]	2000	21.1328 s	−19.233
QKMCC [30]	14	1.763 s	−17.9723
MKRL [26]	2000	21.0313 s	−19.5390
QMKRL [26]	14	1.7243 s	−18.5748
KRMC [21]	2000	48.7601 s	−28.7583
KRMC-NC [21]	496	2.6874 s	−23.671
QRMKRP	14	1.9681 s	−27.3128
RMKRP	2000	48.6101 s	−34.0790

5. Conclusions

In this paper, the kernel risk-sensitive mean p-power error (KRP) criterion is proposed by constructing mean p-power error (MPE) into kernel risk-sensitive loss (KRL) in RKHS, and some basic properties are presented. The KRP criterion with power parameter p is more flexible than KRL to handle the signal corrupted by impulsive noises. Two kernel recursive adaptive algorithms are derived to obtain desirable filtering accuracy under the minimum KRP (MKRP) criterion, i.e., the recursive minimum KRP (RMKRP) and quantized RMKRP (QRMKRP) algorithms. The RMKRP can achieve higher accuracy but with almost identical computational complexity as that of the KRLS and KRMC. The vector quantization method is introduced into RMKRP, thus generating QRMKRP, and QRMKRP can effectively reduce network size while maintaining the filtering accuracy. Simulations conducted in Mackey–Glass (MG) chaotic time series prediction and nonlinear system identification

under impulsive noises illustrate the superiorities of RMKRP and QRMKRP from the aspects of robustness and filtering accuracy.

Author Contributions: Conceptualization, T.Z. and S.W.; methodology, T.Z. and H.Z.; software, T.Z. and L.W.; validation, S.W. and T.Z.; formal analysis, T.Z. and H.Z.; investigation, T.Z. and K.X.; resources, S.W.; data curation, S.W. and T.Z.; writing—original draft preparation, T.Z.; writing—review and editing, S.W. and H.Z.; visualization, T.Z.; supervision, S.W.; project administration, S.W. and L.W.; funding acquisition, S.W. and K.X.

Funding: This work was supported by the National Natural Science Foundation of China (61671389), Fundamental Research Funds for the Central Universities (XDJK2019B011), and the Research Fund for Science and Technology Commission Foundation of Chongqing (cstc2017rgzn-zdyfX0002).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kivinen, J.; Smola, A.J.; Williamson, R.C. Online learning with kernels. *IEEE Trans. Signal Process.* **2004**, *52*, 1540–1547. [CrossRef]
2. Chen, B.; Li, L.; Liu, W.; Príncipe, J.C. Nonlinear adaptive filtering in kernel spaces. In *Springer Handbook of Bio-/Neuroinformatics;* Springer: Berlin, Germany, 2014; pp. 715–734.
3. Nakajima, Y.; Yukawa, M. Nonlinear channel equalization by multi-kernel adaptive filter. In Proceedings of the IEEE 13th International Workshop on Signal Processing Advances in Wireless Communications, Cesme, Turkey, 17–20 June 2012; pp. 384–388.
4. Jiang, S.; Gu, Y. Block-sparsity-induced adaptive filter for multi-clustering system identification. *IEEE Trans. Signal Process.* **2015**, *63*, 5318–5330. [CrossRef]
5. Zheng, Y.; Wang, S.; Feng, J.; Tse, C.K. A modified quantized kernel least mean square algorithm for prediction of chaotic time series. *Digital Signal Process.* **2016**, *48*, 130–136. [CrossRef]
6. Liu, W.; Príncipe, P.P.; Príncipe, J.C. The kernel least mean square algorithm. *IEEE Trans. Signal Process.* **2008**, *56*, 543–554. [CrossRef]
7. Liu, W.; Príncipe, J.C. Kernel affine projection algorithms. *IEEE Trans. Signal Process.* **2004**, *52*, 2275–2285.
8. Engel, Y.; Mannor, S.; Meir, R. The kernel recursive least-squares algorithm. *IEEE Trans. Signal Process.* **2004**, *52*, 2275–2285. [CrossRef]
9. Liu, W.; Park, I.; Príncipe, J.C. An information theoretic approach of designing sparse kernel adaptive filters. *IEEE Trans. Neural Netw.* **2009**, *20*, 1950–1961. [CrossRef]
10. Platt, J. A resource-allocating network for function interpolation. *Neural Comput.* **1991**, *3*, 213–225. [CrossRef]
11. Richard, C.; Bermudez, J.C.M.; Honeine, P. Online prediction of time series data with kernels. *IEEE Trans. Signal Process.* **2009**, *57*, 1058–1067. [CrossRef]
12. Chen, B.; Zhao, S.; Zhu, P.; Príncipe, J.C. Quantized kernel least mean square algorithm. *IEEE Trans. Neural Netw. Learn. Syst.* **2012**, *23*, 22–32. [CrossRef]
13. Chen, B.; Zhao, S.; Zhu, P.; Príncipe, J.C. Quantized kernel recursive least squares algorithm. *IEEE Trans. Neural Netw. Learn. Syst.* **2013**, *24*, 1484–1491. [CrossRef] [PubMed]
14. Príncipe, J.C. *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*; Springer: New York, NY, USA, 2010.
15. Pei, S.-C.; Tseng, C.-C. Least mean *p*-power error criterion for adaptive FIR filter. *IEEE J. Sel. Areas Commun.* **1994**, *12*, 1540–1547.
16. Boel, R.K.; James, M.R.; Petersen, I.R. Robustness and risk sensitive filtering. *IEEE Trans. Autom. Control* **2002**, *47*, 451–461. [CrossRef]
17. Chen, B.; Xing, L.; Xu, B.; Zhao, H.; Zheng, N.; Príncipe, J.C. Kernel risk-sensitive loss: Definition, properties and application to robust adaptive filtering. *IEEE Trans. Signal Process.* **2017**, *65*, 2888–2901. [CrossRef]
18. Ma, W.; Duan, J.; Man, W.; Zhao, H.; Chen, B. Robust kernel adaptive filters based on mean p-power error for noisy chaotic time series prediction. *Eng. Appl. Artif. Intell.* **2017**, *58*, 101–110. [CrossRef]
19. Liu, W.; Pokharel, P.P.; Príncipe, J.C. Correntropy: Properties and applications in non-gaussian signal processing. *IEEE Trans. Signal Process.* **2007**, *55*, 5286–5298. [CrossRef]
20. Chen, B.; Xing, L.; Liang, J.; Zheng, N.; Príncipe, J.C. Steady-state mean-square error analysis for adaptive filtering under the maximum correntropy criterion. *IEEE Signal Process. Lett.* **2014**, *21*, 880–884.

21. Wu, Z.; Shi, J.; Zhang, X.; Ma, W.; Chen, B. Kernel recursive maximum correntropy. *Signal Process.* **2015**, *117*, 11–16. [CrossRef]
22. Zhao, S.; Chen, B.; Príncipe, J.C. Kernel adaptive filtering with maximum correntropy criterion. In Proceedings of the International Joint Conference on Neural Network, San Jose, CA, USA, 31 July–5 August 2011; Volume 31, pp. 2012–2017.
23. He, R.; Hu, B.; Zheng, W.; Kong, X. Robust principal component analysis based on maximum correntropy criterion. *IEEE Trans. Image Process.* **2011**, *20*, 1485–1494.
24. He, R.; Zheng, W.; Hu, B. Maximum correntropy criterion for robust face recognition. *IEEE Trans. Patt. Anal. Mach. Intell.* **2011**, *33*, 1561–1576.
25. Santamaría, I.; Pokharel, P.P.; Príncipe, J.C. Generalized correlation function: Definition, properties, and application to blind equalization. *IEEE Trans. Signal Process.* **2006**, *54*, 2187–2197. [CrossRef]
26. Luo, X.; Deng, J.; Wang, W.; Wang, J.-H.; Zhao, W. A Quantized Kernel Learning Algorithm Using a Minimum Kernel Risk-Sensitive Loss Criterion and Bilateral Gradient Technique. *Entropy* **2017**, *19*, 365. [CrossRef]
27. Chen, B.; Xing, L.; Wang, X.; Qin, J.; Zheng, N. Robust learning with kernel mean p-power error loss. *IEEE Trans. Cybern.* **2017**, *48*, 2101–2113. [CrossRef] [PubMed]
28. Liu, W.; Príncipe, J.C.; Haykin, S. *Kernel Adaptive Filtering: A Comprehensive Introduction*; Wiley: New York, NY, USA, 2010.
29. Weng, B.; Barner, K.E. Nonlinear system identification in impulsive environments. *IEEE Trans. Signal Process.* **2005**, *53*, 2588–2594. [CrossRef]
30. Wang, S.; Zheng, Y.; Duan, S.; Wang, L.; Tan, H. Quantized kernel maximum correntropy and its mean square convergence analysis. *Dig. Signal Process.* **2017**, *63*, 164–176. [CrossRef]
31. Fan, H.; Song, Q. A linear recurrent kernel online learning algorithm with sparse updates. *Neural Netw.* **2014**, *50*, 142–153. [CrossRef] [PubMed]