Recursive Maximum Correntropy Algorithms for Second-Order Volterra Filtering

Ji Zhao, J. Andrew Zhang, *Senior Member, IEEE*, Qiang Li,
Hongbin Zhang, *Senior Member, IEEE*, Xueyuan Wang

***Abstract*—As a special case of the Volterra system, the second-order Volterra (SOV) filter is very efficient for nonlinear system identification. The improved correntorpy based on the generalized Gaussian density function has been proven robust against impulsive noise. In this brief, we propose several SOV filters based on a recursive maximum correntropy (RMC) algorithm for nonlinear system identification. We first introduce a basic RMC algorithm, which faces a trade-off between filtering accuracy and tracking capability due to the use of a fixed forgetting factor (FFF). Two RMCs with variable FF (VFF) are further proposed to enhance the tracking ability. Simulation results demonstrate that our proposed algorithms outperform existing ones in impulsive noise environments and/or in time-varying systems.**

***Index Terms*—Correntropy, adaptive filtering, Volterra filter, variable forgetting factor, system identification, impulsive noise.**

I. Introduction

SYSTEM identification plays a key role in establishing mathematical model for an unknown system from the input-output signals. In the adaptive filtering field, linear system identification has been widely studied and applied [1]. However, it is not very efficient in scenarios with intrinsic complexity and nonlinearity [2], where nonlinear systems identification can be a better solution. As a polynomial system, the Volterra system has been widely studied and used in nonlinear system identification. In practice, the second-order Volterra (SOV) filter is often adopted, since it can realize acceptable modelling accuracy and is computationally efficient [3], [4]. Generally, the SOV filter can be adjusted by adaptive filtering algorithms, which are derived typically from the second-order statistics of the estimation error in the presence of Gaussian noise.

However, in practice, impulsive noise which is sparse and random with hign peak energy in the time domain, can significantly degrade the performance of algorithms based on the second-order statistics. Such impulsive noise can be modelled well by a standard symmetric α-stable distribution, which can be represented by a characteristic function [5], i.e., $\phi(t) = \exp\left(-\gamma|t|^{\alpha}\right)$, where $\alpha \in (0, 2]$ stands for the characteristic exponent describing the intensity of impulsive noise, and $\gamma > 0$ is the dispersion of the noise and plays a role similar to the variance of a Gaussian distribution. Therefore, researchers have proposed and studied various optimization criteria for robust learning. Examples include the least mean p-power error (LMP) criterion [6], [7], logarithmic LMP (LLMP) [8], kernel risk sensitive loss [9] and information theoretic criteria [10], [11]. The improved correntorpy based on the generalized Gaussian density function is a nonlinear and local similarity measure, quantifying how similar two random variables are in a neighborhood of the joint space controlled by the kernel parameters [12]. In addition, the improved entropy can exploit higher-order statistics of the estimation error to enhance the robustness against non-Gaussian noise. Therefore, it has been widely applied to develop robust learning algorithms [12]–[16].

In this brief, we propose several recursive maximum correntropy (RMC) algorithms for nonlinear system identification with impulsive noise: 1) We propose a basic RMC algorithm by combining the SOV filter with the improved correntropy. This is based on the fact that the improved correntropy can provide robustness against outliers, and the recursive method can achieve a smaller steady-state misalignment compared with the gradient method and the affine projection method. However, the basic RMC algorithm needs to overcome a contradiction between the filtering accuracy and the tracking capability, due to the use of a fixed forgetting factor (FFF); 2) Inspired by the work on variable forgetting factor (VFF) [8], [17]–[19], we further propose two variable forgetting factor strategies, which are derived by using the squared a *posteriori* error and the autocorrelation of a *priori* and a *posteriori* errors, respectively. We then apply them to the basic RMC algorithm, and leading to VFF-RMC-I and VFF-RMC-II, respectively. Simulation results are provided and demonstrate that our proposed VFF-RMC algorithms can not only maintain excellent filtering accuracy of RMC, but also enhance its tracking ability. Furthermore, in comparison with the adaptive convex combination of two existing RMC algorithms [15], VFF-RMC-II demonstrates better tracking behaviour and is more computationally efficient.

II. Preparations

A. Second-order Volterra Filter

We consider a nonlinear system identified by a SOV filter shown in Fig. 1, in which n is the time instance, $v(n)$ is

This work was supported in part by the Southwest University of Science and Technology Doctor Fund (Grant no. 20zx7119), in part by the National Natural Science Foundation of China (Grant nos. 61971100, 61771411), in part by the Sichuan Science and Technology Program (Grant no. 2019JDTD0019).

J. Zhao, Q. Li and X. Wang are with the School of Information Engineering, Southwest University of Science and Technology, Mianyang, 621010, PR China. (e-mail: zhaoji@swust.edu.cn, liqiangsir@swust.edu.cn, wangxueyuan@swust.edu.cn).

J. Andrew.Zhang is with the Global Big Data Technologies Centre, University of Technology Sydney, Sydney, NSW 2007 Australia. (e-mail: Andrew.Zhang@uts.edu.au).

H. Zhang is with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, PR China. (e-mail: zhanghb@uestc.edu.cn).

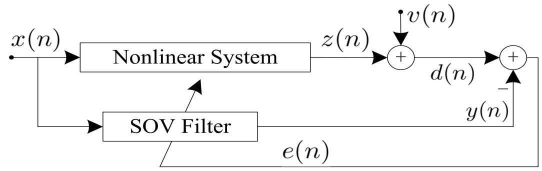

Fig. 1: Diagram of nonlinear system identified by a SOV filter.

the noise signal, $z(n)$ is the desired output of the unknown nonlinear system, $d(n)=z(n)+v(n)$ is the observed output, $y(n)$ is the output of SOV filter, and $e(n)=d(n)-y(n)$ denotes the prediction error. Assume the unknown nonlinear system can be represented by the following second Volterra series

$$d(n)=z(n)+v(n)=\boldsymbol{w}_o^T\boldsymbol{u}(n)+v(n), \tag{1}$$

where $\boldsymbol{w}_o \in \mathbb{R}^{L\times 1}$ is a Volterra kernel of the SOV filter, and $\boldsymbol{u}(n)$ is the expanded input vector. The variables $\boldsymbol{w}_o$ and $\boldsymbol{u}(n)$ are defined as follows

$$\begin{cases} \boldsymbol{u}(n)=\left[\boldsymbol{x}_1(n)^T,\boldsymbol{x}_2(n)^T\right]^T \\ \boldsymbol{x}_1(n)=[x(n),\dots,x(n-N+1)]^T \\ \boldsymbol{x}_2(n)=[x^2(n),x(n)x(n-1),\dots,x^2(n-N+2) \\ \qquad x(n-N+2)x(n-N+1),x^2(n-N+1)]^T \\ \boldsymbol{w}_o=\left[\boldsymbol{w}_{o1}^T,\boldsymbol{w}_{o2}^T\right]^T \\ \boldsymbol{w}_{o1}=[w_{o1}(0),w_{o1}(1),\dots,w_{o1}(N-1)]^T \\ \boldsymbol{w}_{o2}=[w_{o2}(0,0),w_{o2}(0,1),\dots,w_{o2}(N-2,N-2) \\ \qquad w_{o2}(N-2,N-1),w_{o2}(N-1,N-1)]^T, \end{cases} \tag{2}$$

where $\boldsymbol{w}_{o1}$ is the linear kernel with length N, $\boldsymbol{w}_{o2}$ is the second kernel with length $0.5N(N+1)$, $\boldsymbol{x}_1(n)$ is the N most recent inputs, and $\boldsymbol{x}_2(n)$ is the second nonlinear combination of $\boldsymbol{x}_1(n)$. Hence, the total length of SOV is $L=0.5N(N+3)$.

B. Recursive Maximum Correntropy Algorithm for SOV Filter

In practice, non-Gaussian noise with high amplitude is usually contained in $v(n)$. To effectively approximate the kernels in (1), we use the improved correntropy criterion[1] to derive a recursive algorithm to update the estimation $\boldsymbol{w}(n)$ of $\boldsymbol{w}_o$, since the improved correntropy has higher-order absolute moments of the error and is robust against the impulsive noise [12].

For the sake of simplicity, we ignore the normalization constant $s/(2t\Gamma(s^{-1}))$ in (23) and consider an exponentially-weighted cost function as follows

$$J(\boldsymbol{w}(n))=\sum_{i=1}^{n}\beta^{n-i}\exp\left(-\tau|d(i)-y(i)|^s\right), \tag{3}$$

where $y(i)=\boldsymbol{w}(n)^T\boldsymbol{u}(i)$ is the output of the SOV filter, $\beta\in(0,1]$ is a forgetting factor. Setting the gradient of (3) with respect to $\boldsymbol{w}(n)$ to a null vector, we get

$$\begin{cases} \boldsymbol{w}(n)=[\boldsymbol{R}(n)]^{-1}\,\boldsymbol{p}(n) \\ \boldsymbol{R}(n)=\displaystyle\sum_{i=1}^{n}\beta^{n-i}f(e(i))\boldsymbol{u}(i)\boldsymbol{u}(i)^T \\ \boldsymbol{p}(n)=\displaystyle\sum_{i=1}^{n}\beta^{n-i}f(e(i))y(i)\boldsymbol{u}(i) \\ f(e(i))=\exp\left(-\tau|e(i)|^s\right)|e(i)|^{s-2} \\ e(i)=d(i)-y(i)=d(i)-\boldsymbol{w}(n)^T\boldsymbol{u}(i), \end{cases} \tag{4}$$

where $\boldsymbol{R}(n)$ and $\boldsymbol{p}(n)$ stand for a weighted autocorrelation matrix of the input and a weighted cross correlation vector between the noisy output and the input, respectively. Under fixed values of τ and s, the weight factor $f(e(i))$ suppresses effectively the negative influence of impulsive noise and enhances the robustness of algorithms derived from the improved correntropy.

We can then approximate $\boldsymbol{R}(n)$ in a recursive form as $\boldsymbol{R}(n)\approx\beta\boldsymbol{R}(n-1)+f(e(n))\boldsymbol{u}(n)\boldsymbol{u}(n)^T$. Based on the matrix inversion Lemma (5.4) in [1], the inverse of $\boldsymbol{R}(n)$, i.e., $\boldsymbol{Q}(n)=[\boldsymbol{R}(n)]^{-1}$, can be estimated as

$$\begin{cases} \boldsymbol{Q}(n)=\beta^{-1}\boldsymbol{Q}(n-1)-\beta^{-1}\boldsymbol{g}(n)\boldsymbol{u}(n)^T\boldsymbol{Q}(n-1) \\ \boldsymbol{g}(n)=\dfrac{f(e(n))\boldsymbol{Q}(n-1)\boldsymbol{u}(n)}{\beta+f(e(n))\boldsymbol{u}(n)^T\boldsymbol{Q}(n-1)\boldsymbol{u}(n)}. \end{cases} \tag{5}$$

Similarly, we get $\boldsymbol{p}(n)\approx\beta\boldsymbol{p}(n-1)+f(e(n))d(n)\boldsymbol{u}(n)$. Combining $\boldsymbol{p}(n)$, (4) and (5), after some calculations, we get the updated formulation of $\boldsymbol{w}(n)$ as

$$\boldsymbol{w}(n)=\boldsymbol{w}(n-1)+\boldsymbol{g}(n)e(n), \tag{6}$$

where $e(n)=d(n)-\boldsymbol{w}(n-1)^T\boldsymbol{u}(n)$ denotes a *priori* error. Algorithm 1 summarizes the RMC algorithm[2], which can achieve excellent filtering accuracy as can be seen from the simulation results. It is noted that numerical simulations were conducted due to the complexity of an analytical analysis. However, simulation results also reveal that RMC need to be balanced between the steady-state misalignment and the tracking ability due to the use of the FFF. In the following section, we present two VFF strategies to overcome this problem.

III. RMC WITH VARIABLE FORGETTING FACTOR

Although, in RMC, the contradiction between filtering accuracy and tracking capability can be resolved by the convex combination method, the computational complexity is at least doubled [15]. Alternatively, we propose two enhanced RMCs with two VFF strategies derived from the squared a *posteriori* error and the autocorrelation of a *priori* and a *posteriori* errors, respectively.

A. VFF-RMC-I Based on a Posteriori Error

In this situation, (6) is rewritten as

$$\begin{cases} \boldsymbol{w}(n)=\boldsymbol{w}(n-1)+\bar{\boldsymbol{g}}(n)e(n) \\ \bar{\boldsymbol{g}}(n)=\dfrac{f(e(n))\boldsymbol{Q}(n-1)\boldsymbol{u}(n)}{\beta(n)+f(e(n))\|\boldsymbol{u}(n)\|_{\boldsymbol{Q}}^2}, \end{cases} \tag{7}$$

[1] Defined in the Appendix A.

[2] RMC: step 1 to step 2 and step 6 with $\beta(n)=\beta_f$; VFF-RMC: step 1 to step 5 and step 7, with $\beta(n)=\beta^\star$.

Algorithm 1: Pseudocode of Proposed Algorithms

Initialization:
parameters: $N > 0 \Rightarrow L = 0.5N(N+3)$, $\tau > 0$, $s > 0$, $\beta_f \in (0, 1]$, $a > b = c = d \in (0, 1)$, $\theta \in (1, 2]$, $N_w \in [5, 9]$,
$\hat{\sigma}_v^2(0) = \hat{\sigma}_e^2(0) = \hat{\sigma}_q^2(0) = \hat{q}(0) = 0$, $\boldsymbol{w}(0) = \boldsymbol{0}$, $\boldsymbol{Q}(0) = \boldsymbol{I}$
Computation:
while $\{\boldsymbol{u}(n), d(n)\}$ $(n \geq 1)$ available **do**
1) $y(n) = \boldsymbol{w}(n-1)^T \boldsymbol{u}(n)$, $\quad e(n) = d(n) - y(n)$
2) $f(e(n)) = \exp\left(-\tau|e(n)|^s\right)|e(n)|^{s-2}$, $\; q(n) = f(e(n))\|\boldsymbol{u}(n)\|_{\boldsymbol{Q}}^2$
3) Update $\hat{\sigma}_v^2(n)$ by (12), $\quad \hat{\sigma}_e^2(n)$ by (13)
4) the variable forgetting factors
$\beta^\star = \begin{cases} \beta_f, & \hat{\sigma}_e(n) \leq \theta\hat{\sigma}_v(n) \\ \text{VFF-I: (12)} - \text{(15) or VFF-II: (12), (13), (18) and (19)} \end{cases}$
5) Based on VFF or FFF used, one can choose $\beta(n) \in \{\beta_f, \beta^\star\}$
6) For FFF, update $\boldsymbol{w}(n)$ by (6)
7) For VFF, update $\boldsymbol{w}(n)$ by (7)
end while

where $\|\boldsymbol{u}(n)\|_{\boldsymbol{Q}}^2 = \boldsymbol{u}(n)^T \boldsymbol{Q}(n-1)\boldsymbol{u}(n)$. Defining a *posteriori* error of RMC as $e_p(n) = d(n) - \boldsymbol{w}(n)^T \boldsymbol{u}(n)$ and, considering (7), we get

$$e_p(n) = e(n)\left(1 - \boldsymbol{g}(n)^T \boldsymbol{u}(n)\right). \tag{8}$$

Squaring both sides of (8), and then taking the expectations, we obtain

$$\begin{aligned} E\left[e_p^2(n)\right] &= E\left[e^2(n)\left(1 - \frac{q(n)}{\beta(n) + q(n)}\right)^2\right] \\ &\overset{(a)}{\approx} E\left[e^2(n)\right] E\left[\left(1 - \frac{q(n)}{\beta(n) + q(n)}\right)^2\right], \end{aligned} \tag{9}$$

where $q(n) = f(e(n))\|\boldsymbol{u}(n)\|_{\boldsymbol{Q}}^2$, $E[\cdot]$ is mathematical expectation, and the inequality (a) is obtained based on two reasons: 1) $f(e(n))$ is bounded, and under impulsive noise, $f(e(n)) \to 0$, thus the effect of $f(e(n))$ can be ignored; 2) we can assume that the input and error signals are uncorrelated. In addition, $E\left[e_p^2(n)\right]$ can be approximated by the variance of system noise $v(n)$, i.e., $E\left[e_p^2(n)\right] \approx E\left[v^2(n)\right] = \sigma_v^2$ [19]. And, then we can get the following result from (9)

$$E\left[\left(1 - \frac{q(n)}{\beta(n) + q(n)}\right)^2\right] = \frac{\sigma_v^2}{\sigma_e^2}, \tag{10}$$

where $\sigma_e^2 = E\left[e^2(n)\right]$ denotes the power of $e(n)$. Based on (10), we can obtain a variable forgetting factor as

$$\beta(n) = \frac{\sigma_q \sigma_v}{\sigma_e - \sigma_v}, \tag{11}$$

where $\sigma_q^2 = E\left[q^2(n)\right]$. Theoretically, the impulsive noise modelled by the α-stable distribution has $\sigma_v^2 = \infty$ with $\alpha \in (0, 2)$. However, in practice, σ_v^2 can be estimated by the following robust median operation [8]

$$\begin{cases} \hat{\sigma}_v^2(n) = a\hat{\sigma}_v^2(n-1) + (1-a)C(e(n)) \\ C(e(n)) = \dfrac{1.483(N_w - 1)}{N_w + 4}\text{med}\left(\boldsymbol{\epsilon}_e(n)\right) \\ \boldsymbol{\epsilon}_e(n) = \left[e^2(n), \ldots, e^2(n - N_w + 1)\right]^T, \end{cases} \tag{12}$$

where $a \in (0, 1)$ is a weighting factor, med$(\cdot)$ denotes the median operation, and N_w stands for the length of the estimation window and is usually in $[5, 9]$. Similarly, the power of $e(n)$ can be estimated as follows

$$\hat{\sigma}_e^2(n) = b\hat{\sigma}_e^2(n-1) + (1-b)C(e(n)), \tag{13}$$

where $b \in (0, 1)$ is a weighting factor. Since $q(n)$ is related to $f(e(n))$ and the weighted norm of input, in practice, σ_q^2 can be estimated by an exponential window as

$$\hat{\sigma}_q^2(n) = c\hat{\sigma}_q^2(n-1) + (1-c)q^2(n), \tag{14}$$

where $c \in (0, 1)$ is also a weighting factor. Generally, we can set $a > b = c$ to realize a good filtering performance [19].

Observing the VFF $\beta(n)$ in (11), we can anticipate that $\hat{\sigma}_e(n) \leq \hat{\sigma}_v(n)$ may happen during the evolution of our proposed algorithm. Therefore, when $\hat{\sigma}_e(n) \leq \theta\hat{\sigma}_v(n)$ with $\theta \in (1, 2]$, $\beta(n)$ can be replaced by a fixed value $\beta_f \in \{0, 1\}$. In addition, when the system changes, $\hat{\sigma}_e(n) \gg \hat{\sigma}_v(n)$, and thus (11) holds. Therefore, (11) can be modified as

$$\beta(n) = \begin{cases} \beta_f, & \hat{\sigma}_e(n) \leq \theta\hat{\sigma}_v(n) \\ \min\left\{\dfrac{\hat{\sigma}_q(n)\hat{\sigma}_v(n)}{\hat{\sigma}_e(n) - \hat{\sigma}_v(n)}, \beta_f\right\}, & \text{otherwise,} \end{cases} \tag{15}$$

where $\min\{\cdot\}$ is the minimum operation to set $\beta(n) \in (0, 1)$.

Remark 1 : From (15) we can find that: 1) when $\hat{\sigma}_e(n) \gg \hat{\sigma}_v(n)$, a small $\beta(n)$ is obtained, which enhances the tracking ability of proposed algorithm when system changes; 2) when $\hat{\sigma}_e(n)$ is very close to $\hat{\sigma}_v(n)$, a large fixed β_f results in better filtering accuracy of the proposed algorithm. We denote the RMC with a variable forgetting factor in (15) as VFF-RMC-I, which is summarized in Algorithm 1.

B. VFF-RMC-II Based on a Priori and a Posteriori Errors

Observing the related derivations of (15), we can find that (9) is the key step, which relates $\beta(n)$ to the powers of the *priori* error and noise. Therefore, we can anticipate that there exist other methods to obtain such relations, for example, the autocorrelation of a *priori* and a *posteriori* errors. In this situation, the goal is to obtain a forgetting factor $\beta(n)$ that can lead to the following equation

$$\begin{aligned} E\left[e_p(n)e(n)\right] &= E\left[e^2(n)\left(1 - \frac{q(n)}{\beta(n) + q(n)}\right)\right] \\ &\overset{(b)}{\approx} E\left[e^2(n)\right] E\left[1 - \frac{q(n)}{\beta(n) + q(n)}\right] = \sigma_v^2, \end{aligned} \tag{16}$$

where the (b) is obtained in the same way as (a) in (9). Therefore, after some calculations, (16) becomes

$$\beta(n) = \frac{E\left[q(n)\right]\sigma_v^2}{\sigma_e^2 - \sigma_v^2}. \tag{17}$$

Based on the estimations in (12) and (13), and the remarks on (15), we can rewrite (17) as

$$\beta(n) = \begin{cases} \beta_f, & \hat{\sigma}_e(n) \leq \theta\hat{\sigma}_v(n) \\ \min\left\{\dfrac{\hat{q}(n)\hat{\sigma}_v^2(n)}{\hat{\sigma}_e^2(n) - \hat{\sigma}_v^2(n)}, \beta_f\right\}, & \text{otherwise} \end{cases} \tag{18}$$

where $\hat{q}(n)$ is the estimation of $E\left[q(n)\right]$ and is defined by

$$\hat{q}(n) = d\hat{q}(n-1) + (1-d)q(n). \tag{19}$$

where $d \in (0, 1)$ is a weighting factor. In practice, we can set $a > b = d$ to achieve acceptable filtering performance [8].

Remark 2: Compared with (15), (18) has almost the same function. However, (18) does not require the squared operation. We call the RMC with a variable forgetting factor in (18) as VFF-RMC-II, which is also summarized in Algorithm 1.

C. Computational Complexity

For the basic RMC algorithm, the main required computation is to update $\boldsymbol{Q}(n)$ in (5), and the complexity is $\mathcal{O}(L^2)$. Compared with RMC, the two VFF-RMCs require some additional computations as given in (12)-(15) and (12), (13), (18) and (19), respectively. These computations are two comparisons, seven multiplications, and $\mathcal{O}(N_w \log N_w)$ in the median operation. However, when a convex combination method is used, e.g., the adaptive convex combination of two RMC algorithms with control (AC-RMC-C) [15], the additional complexity is at least $\mathcal{O}(L^2)$. Therefore, in comparison with AC-RMC-C, our proposed VFF-RMCs are computationally more efficient. Furthermore, as will be seen from Section IV, VFF-RMCs can achieve faster tracking than AC-RMC-C when system changes. In addition, the recursive LLMP (RLLMP) and improved VFF-RLLMP (IVFF-RLLMP) [8] are also able to identify the SOV system. The computational complexity of both RLLMP algorithms is also $\mathcal{O}(L^2)$. However, our proposed algorithms outperform them in terms of filtering accuracy.

IV. Simulation Results

In this part, we implement some examples of nonlinear system identification to test the performance of our proposed schemes in terms of filtering accuracy and tracking ability. A unknown SOV system with $N = 4$ is defined by

$$\begin{cases} \boldsymbol{w}_o = \left[\boldsymbol{w}_{o1}^T, \boldsymbol{w}_{o2}^T\right]^T, \ \boldsymbol{w}_{o1} = [1, -0.8, 0, 1.9]^T \\ \boldsymbol{w}_{o2} = [0.95, 0, 1.1, 0, 0, 0, 0, -0.63, 0, 0]^T \\ \boldsymbol{u}(n) = \left[\boldsymbol{x}_1(n)^T, \boldsymbol{x}_2(n)^T\right]^T, z(n) = \boldsymbol{w}_o^T \boldsymbol{u}(n). \end{cases} \tag{20}$$

which was used in [2], [8]. In the following experiments, unless noted otherwise, the input $x(n)$ is produced by a Gaussian distribution with zero-mean and unit-variance, and we use the normalized mean square deviation (NMSD) defined as $20 \log_{10} \frac{\|\boldsymbol{w}_o - \boldsymbol{w}(n)\|}{\|\boldsymbol{w}_o\|}$ to measure the filtering performance. All simulation results are obtained by averaging over 100 independent runs.

First, we compare the filtering performance of RMC, VFF-RMC-I and VFF-RMC-II under impulsive noise with $\alpha = 1.25$ and $\gamma = 1/15$. In this experiment, the SOV changes $\boldsymbol{w}_o$ to $-\boldsymbol{w}_o$ at $n = 2000$. For RMC, the parameters are $\beta_f = 0.99$, $\tau = 0.001$, $s = 1.4$; For VFF-RMC-I and VFF-RMC-II, $a = 0.98$, $b = 0.95$, $N_w = 8$, $\theta = 1.5$, and other parameters are the same with those for RMC. Fig. 2 plots the NMSD filtering curves, and shows that : 1) before SOV changes, three RMCs realize the same filtering performance, since two VFF-RMCs have the same forgetting factor $\beta(n) = 0.99$; 2) after SOV changes, the two VFF-RMCs can not only hold the filtering accuracy in the steady-state, but also enhance the tracking ability. This is particularly obvious for VFF-RMC-II, as VFF-II achieves smaller forgetting factor than VFF-I after system changes; 3) although the adaptation of $\beta(n)$ lasts only some iterations, it is very efficient for enhancing the convergence rate of RMC after system changes. Furthermore, we have conducted many experiments to test our proposed algorithms, which can achieve convergence and realize acceptable filtering accuracy and convergence rate[3].

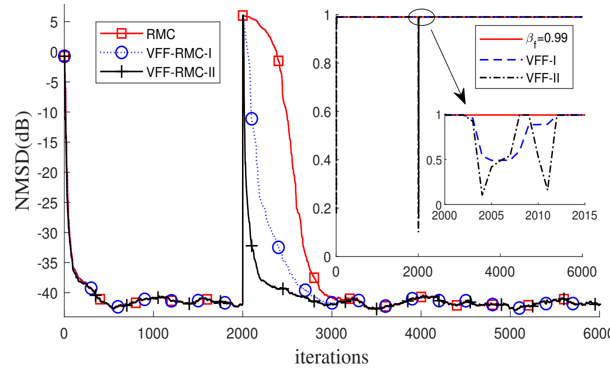

Fig. 2: The NMSD curves of proposed RMCs with Gaussian input and impulsive noise with $\alpha = 1.25$ and $\gamma = 1/15$.

Then, we compare our proposed RMCs with AC-RMC-C, RLLMP and IVFF-RLLMP. In this experiment, each algorithm trains 12000 iterations. The white Gaussian input signal is used for $n \in (0, 6000]$, and then a coloured input signal is used for $n \in [6001, 12000]$, and it is generated by filtering a zero-mean Gaussian signal with unity-variance through a second-order system $H(z) = (1 + 0.6z^{-1})/(1 + z^{-1} + 0.21z^{-2})$. For the three RMC algorithms, they have the same parameters as those in the previous. For AC-RMC-C, the large FFF is 0.99, the small FFF is 0.9, the rate for adaption parameter is $\mu_\chi = 0.9$, the selection parameter is $\theta = 0.9$, and the control factor is $\epsilon_2 = 0.003$; For RLLMP, $p = 1.2$, and the FFF is 0.99; For IVFF-RLLMP, the parameters are the same with those for RLLMP, except that $\varsigma = 0.95$, $\beta = 0.98$, $\theta = 1.5$, and $N_w = 8$. Fig. 3 plots the NMSD filtering curves. From this figure, we can observe that: 1) From the perspective of filtering accuracy, all RMC algorithms outperform two RLLMP algorithms; 2) VFF-RMC-II demonstrates the best tracking behaviour among the four RMC algorithms; 3) In comparison with AC-RMC-C, VFF-RMC-I can achieve better tracking ability, at a lower complexity.

V. Conclusion

In this brief, the RMC algorithm is developed for SOV system identification with impulsive noise. We also further propose two VFF-RMCs to enhance the tracking ability of RMC. Simulation results demonstrate that: 1) our proposed algorithms achieve better filtering accuracy than LLMP based algorithms; 2) based on the autocorrelation of *a priori* and *a posteriori* errors, the variable forgetting factor introduced in our algorithms can lead to better tracking ability for recursive-type algorithms. Our algorithms may be further improved by exploiting other relevance between the priori error and noise.

[3]Due to the page limit, some simulation results are plotted in a supplementary file.

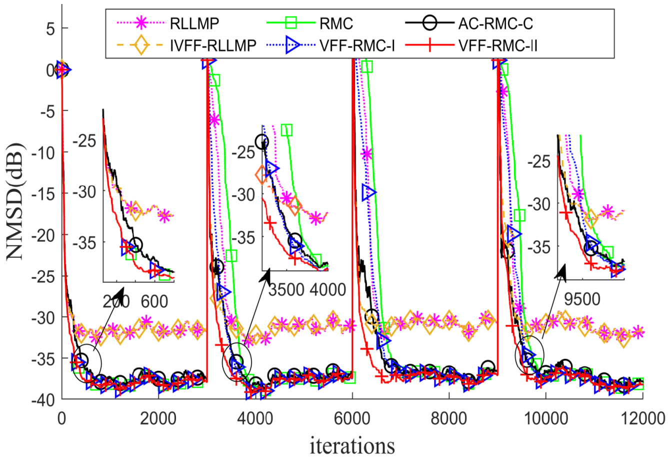

Fig. 3: The NMSD curves of compared algorithms with Gaussian input signal for $n \in (0, 6000]$ and with coloured input signal $H(z)$ for $n \in [6001, 12000]$, and impulsive noise with $\alpha = 1.25$ and $\gamma = 0.1$. The SOV changes $\left[\boldsymbol{w}_{o1}^T, \boldsymbol{w}_{o2}^T\right]^T$ to $\left[\boldsymbol{w}_{o1}^T, -\boldsymbol{w}_{o2}^T\right]^T$, $\left[\boldsymbol{w}_{o1}^T, \boldsymbol{w}_{o2}^T\right]^T$ to $\left[-\boldsymbol{w}_{o1}^T, \boldsymbol{w}_{o2}^T\right]^T$ and $\left[\boldsymbol{w}_{o1}^T, \boldsymbol{w}_{o2}^T\right]^T$ to $\left[-\boldsymbol{w}_{o1}^T, -\boldsymbol{w}_{o2}^T\right]^T$, at $n = 3000$, $n = 6000$ and $n = 9000$, respectively.

Appendix A
The Improved Correntropy

As a local similarity measure between two random variables X and Y, the correntropy has been widely used in adaptive system identification [11], [20]–[22]. It is defined as

$$V(X, Y) = E\left[\kappa(x - y)\right] = \int \kappa(x - y) dF_{x,y}(x, y), \qquad (21)$$

where $E\left[\cdot\right]$ is the mathematical expectation, $\kappa(\cdot)$ is a Mercer kernel, and $F_{x,y}(x, y)$ is the joint distribution function of X and Y. Generally, the $\kappa(\cdot)$ used in (21) is a Gaussian kernel defined by

$$\kappa_\sigma(x - y) = (\sqrt{2\pi})^{-1} \exp\left(-0.5\sigma^{-2}|x - y|^2\right), \qquad (22)$$

where $\sigma > 0$ denotes the kernel size and $(\sqrt{2\pi})^{-1}$ is the normalization parameter. However, Gaussian kernel is not always the best choice, and hence the feature of $V(X, Y)$ is expanded by using the following generalized Gaussian density function to replace the Gaussian kernel [12]

$$\kappa_{s,t}(x - y) = \frac{s}{2t\Gamma(s^{-1})} \exp\left(-\tau|x - y|^s\right), \qquad (23)$$

where $s > 0$ is the shape parameter, $t > 0$ is the scale parameter, $\tau = t^{-s}$ is the kernel parameter, $\Gamma(\cdot)$ is the gamma function, and $s/(2t\Gamma(s^{-1}))$ denotes the normalization constant. Equation (23) means that the correntropy with Gaussian kernel is a special case of $\kappa_{s,t}(x-y)$ with $s = 2$. Adapting to various situations, the correntropy with (23) can realize satisfactory filtering performance by adjusting the values of s and τ. Therefore, the new kernel can effectively extend the properties and application scenarios of the correntropy. We call equation (21) with the kernel (23) as the improved correntropy. In practice, the joint distribution $F_{x,y}(x, y)$ is unknown, and only a finite number of data $\{x_i, y_i\}_{i=1}^N$ are available, resulting in the sample estimator of the improved correntropy, i.e.,

$$\hat{V}_N^{s,t}(X, Y) = \frac{1}{N} \sum_{i=1}^{N} \kappa_{s,t}(x_i - y_i). \qquad (24)$$

As proposed in [12], (24) can be used as a cost function for adaptation algorithms with robustness against impulsive noises [12]–[16].

References

[1] A. H. Sayed, *Adaptive Filters*, John Wiley & Sons, Inc. Hoboken, New Jersey, 2008.

[2] B. Weng and K. E. Barner, "Nonlinear system identification in impulsive environments," *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2588–2594, Jul. 2005.

[3] T. Ogunfunmi, *Adaptive Nonlinear System Identification: The Volterra and Wiener Based Approachs*, New York, NY, USA: Springer-Verlag, 2007.

[4] L. Lu, W. Wang, X. Yang, W. Wu, and G. Zhu, "Recursive Geman-McClure estimator for implementing second-order Volterra filter," *IEEE Trans.Circuits Syst. II, Exp. Briefs*, vol. 66, no. 7, pp. 1272–1276, Jul. 2019.

[5] M. Shao and C. L. Nikias, "Signal processing with fractional lower order moments: stable processes and their applications," *Proc. IEEE,* vol. 81, no. 7, pp. 986–1010, Jul. 1993.

[6] R. Pogula, T. K. Kumar, and F. Albu, "Robust sparse normalized LMAT algorithms for adaptive system identification under impulsive noise environments," *Circuits Syst. Signal Process.,* vol. 38, no. 11, pp. 5103–5134, Nov. 2019.

[7] J. Zhao, H. Zhang, G. Wang, and J. A. Zhang, "Projected kernel least mean p-power algorithm: convergence analyses and modifications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 10, pp. 3498–3511, Oct. 2020.

[8] L. Lu, H. Zhao, and B. Chen, 'Improved-variable-forgetting-factor recursive algorithm based on the logarithmic cost for Volterra system identification," *IEEE Trans. Circuits Syst. II Express Briefs,* vol. 63, no. 6, pp. 588–592, Jun. 2016.

[9] B. Chen, L. Xing, B. Xu, H. Zhao, N. Zheng, and J. C. Príncipe, "Kernel risk-sensitive loss: definition, properties and application to robust adaptive filtering," *IEEE Trans. Signal Process.*, vol. 65, no. 11, pp. 2888–2901, Jun. 2017.

[10] J. C. Príncipe, *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*, Springer New York Dordrecht Heidelberg London, 2010.

[11] W. Liu, P. P. Pokharel, and J. C. Príncipe, "Correntropy: properties and applications in non-Gaussian signal processing," *IEEE Trans. Signal Process*., vol. 55, no. 11, pp. 5286–5298, Nov. 2007.

[12] B. Chen, L. Xing, Z. Zheng, and J. C. Príncipe, "Generalized correntropy for robust adaptive filtering," *IEEE Trans. Signal Process.,* vol. 64, no. 13, pp. 3376–3387, Jul. 2016.

[13] J. Zhao, H. Zhang, and J. A. Zhang, "Generalized maximum correntropy algorithm with affine projection for robust filtering under impulsive-noise environments," *Signal Process.,* vol. 172, 107524, 2020.

[14] L. Chen, H. Qu, and J. Zhao, "Generalized correntropy based deep learning in presence of non-Gaussian noises," *Neurocomput.,* vol. 278, pp. 41–50, 2018.

[15] J. Zhao, H. Zhang, and G. Wang, "Fixed-point generalized maximum correntropy: convergence analysis and convex combination algorithms," *Signal Process.,* vol. 154, pp. 64–73, 2019.

[16] Y. Zhu, H. Zhao, X. Zeng, and B. Chen, "Robust generalized maximum correntropy criterion algorithms for active noise control," *IEEE Trans. Audip, Speech, Lang. Process.,* vol. 28, pp. 1282–1292, 2020.

[17] F. Albu, "Improved variable forgetting factor recursive least square algorithm," *12th Int. Conf. Control, Autom., Robot. Vis.,* Guangzhou, China, 2012, pp.1789–1793.

[18] Z. Banjac, B. Kovačević, and I. K. Kovačević, "Variable forgetting factor estimation in impulsive noise environment," *22nd Telecommun. Forum Telfor,* Belgrade, 2014, pp.449–452.

[19] C. Paleologu, J. Benesty and S. Ciochina, "A robust variable forgetting factor recursive least-squares algorithm for system identification," *IEEE Signal Process. Lett.,* vol. 15, 2008, pp. 597–600.

[20] Y. Li, Y. Wang, R. Yang, and F. Albu, "A soft parameter function penalized normalized maximum correntropy criterion algorithm for sparse system identification," *Entropy*, vol. 19, no. 1, pp. 45–60, 2017.

[21] Y. Li, Y. Wang, F. Albu, and J. Jiang, "A general zero attraction proportionate normalized maximum correntropy criterion algorithm for sparse system identification," *Symmetry*, vol. 9, no. 10, pp. 229–241, 2017.

[22] Z. Jiang, Y. Li, and Y. Zakharov, "A robust group-sparse proportionate affine projection algorithm with maximum correntropy criterion for channel estimation," *27th European Signal Process. Conf.,* A Coruna, Spain, 2019, pp.1–5.