**ABSTRACT**

To address sparse system identification problem in non-Gaussian impulsive noise environment, the recursive generalized maximum correntropy criterion (RGMCC) algorithm with sparse penalty constraints is proposed to combat impulsive-inducing instability. Specifically, a recursive algorithm based on the generalized correntropy with a forgetting factor of error is developed to improve the performance of the sparsity aware MCC algorithms via achieving robust steady-state error. Considering an unknown sparse system, the \(l_1\)-norm and correntropy induced metric (CIM) are employed in the RGMCC algorithm to exploit sparsity as well as to mitigate the impulsive noise simultaneously. Numerical simulations are given to show that the proposed algorithm is robust while providing robust steady-state estimation performance.

**Key words:** Recursive generalized correnyropy criterion, correntropy induced metric, \(l_1\)-norm, sparse system identification, non-Gaussian noise.

I. INTRODUCTION

In recent years, many sparse adaptive filtering (SAF) algorithms have been proposed for multipath channel estimation, underwater acoustic (UWA) channel probing as well as sparse coding [1-3]. The sparsity aware least mean square (LMS) algorithms [4-5] are among the outstanding representatives of such SAF methods. Over the years, the family of sparse LMS [6-11] have been applied to address sparse system identification (SSI) problem. The recursive least square (RLS) is another well-known algorithm for adaptive filtering. However, the RLS-type SAF algorithms are relatively few. The sparse RLS algorithm was first proposed by Babadi [12], in which the RLS is regularized by adding a weighted \(l_1\)-norm sparsity constraint. Eksioglu developed a new sparse RLS algorithm using a general convex function of the system estimate as a regularizing term [13]. Furthermore, \(l_1\)-norm and reweighted \(l_1\)-norm regularized RLS were proposed for SSI in [14-15]. Under non-Gaussian noise (especially impulsive noise) environments, however, these methods usually result in poor performance due to the fact that the mean square error (MSE) criterion used in those algorithms is very sensitive to large errors. Indeed, physical experiments have confirmed that impulsive noises are often occurred in many systems such as man-made low frequency atmospheric noise systems and underwater acoustic systems [16-17].

In recent years, robust SAF methods have been studied intensely. One of the efforts is to search for robust cost function to take place of MSE and use sparse penalty constraints (SPC) to develop robust SAF algorithms for SSI under non-Gaussian environments. A number of \(p\)-norm based sparse channel parameter estimation methods (such as sparse LMP [18] and sparse RLP [19]) are proposed to alleviate the negative effect of impulsive noise. Furthermore, the minimum error entropy (MEE) [20] and maximum correntropy criterion (MCC) [21] in information theoretical learning (ITL) have been developed as robust and efficient cost functions for non-Gaussian signal processing [22-23]. Taking advantages of the ITL, MEE and MCC based SAF algorithms were developed in [24-25]. However, these methods still have flaws in convergence speed and steady-state performance due to adopting the gradient based methods to search the optimal solution. At present, a generalized MCC (GMCC) has been proposed in [26] for robust adaptive filtering. In this work, we will develop a recursive GMCC (RGMCC) with SPC in a manner alike to the approach as outlined in [13] to address SSI under non-Gaussian noise environment. Simulation results are given to show the excellent performance of the proposed algorithm.

The rest of the paper is organized as follows. In Section II, the MCC and GMCC are briefly reviewed. In Section III, the sparse RGMCC algorithms are derived. In Section IV, simulation results are provided to demonstrate the performance of the proposed algorithms. Finally, conclusion is given in Section V.

II. REVIEW OF CORRENTROPY

**Definition 1.** Correntropy between any two random variables \(X\) and \(Y\) is a generalized similarity measure defined by [21]

\[
\forall (X,Y) = E(\kappa_n(X - Y)) = \int \kappa_n(x-y)f_{XY}(x,y)dxdy \quad (1)
\]
where $E(\cdot)$ denotes the expectation operator, $\kappa_\sigma$ is a shift-invariant kernel function with bandwidth $\sigma$, and $f(x,y)$ is the joint pdf of $X$ and $Y$. The correntropy is a similarity measure of how similar two random variables are, within a small neighborhood determined by $\sigma$. In practice, the given sample data of the random variables are finite and the joint pdf is unknown in general, leading to the sample mean estimator of correntropy as in definition 2.

**Definition 2.** Given finite samples $(x_i,y_i)$, of the variables $X$ and $Y$, the sample mean estimator of correntropy is

$$\hat{V}(X,Y) = \frac{1}{N} \sum_{i=1}^{N} \kappa_\sigma(x_i - y_i) = \frac{1}{N} \sum_{i=1}^{N} \kappa_\sigma(e_i) \quad (2)$$

where $e_i = x_i - y_i$. Correntropy of the error can be used as a cost function for adaptive systems training, which is called the maximum correntropy criterion (MCC). It has been employed to develop adaptive filters [27-29]. The most popular kernel function used in correntropy is the Gaussian kernel

$$\kappa_\sigma(x - y) = \frac{1}{\sigma\sqrt{2\pi}} \exp \left( -\frac{(x - y)^2}{2\sigma^2} \right) \quad (3)$$

**Remark 1:** Compared with the MSE, MCC (with Gaussian kernel) has some excellent property [30-31]: 1) it is always bounded for any distribution; 2) it contains all even-order moments and is useful for non-Gaussian signal processing; 3) the weights of higher-order moments are controlled by kernel width; 4) it is a local similarity measure, and is very robust to outliers point. Fig. 1 shows the correntropy in terms of the residual error with different values of the kernel width. As the residual error increases, the correntropy is sensitive within a range of small residual errors, and this range is controlled by the kernel width. For large magnitudes of residual error, the correntropy falls to zero. Consequently, MCC is robust to large outliers.

![Illustration of the correntropy objective function](image)

Now, we introduce a weight factor or forgetting factor to define a new estimator of correntropy as [29].

**Definition 3:** The new correntropy estimator with forgetting factor

$$\hat{V}_w(X,Y) = \sum_{i=1}^{N} \lambda_i^{n} \kappa_\sigma(x_i - y_i), (0 < \lambda < 1) \quad (4)$$

where $0 < \lambda < 1$ is the forgetting factor.

**Definition 4:** The generalized correntropy with the generalized Gaussian density (GGD) is given by

$$V_{\gamma}(X,Y) = E\left( G_{\gamma}(X-Y) \right) \quad (5)$$

where $G_{\gamma}(e) = \frac{p}{2q} \exp \left( -\frac{|e|^q}{q} \right) E\left( -|e|^q \right)$, with $\Gamma(\cdot)$ being the gamma function, $e = x - y$, $p > 0$ the shape parameter, $q > 0$ the scale parameter, $T = 1/q^q$ the kernel parameter, and $x_{\gamma} = p/2qT(1/p)$ the normalization constant. The detailed property of the generalized correntropy can be viewed in [26].

Similar to (2), the sample mean estimator of (5) is

$$\hat{V}_w(X,Y) = \sum_{i=1}^{N} \lambda_i^{n} G_{\gamma}(x_i - y_i) \quad (6)$$

By definition (4), we give a weighted mean estimator of the generalized correntropy as

$$\hat{V}_w(X,Y) = \sum_{i=1}^{N} \lambda_i^{n} G_{\gamma}(x_i - y_i) \quad (7)$$

Just like MCC, the generalized correntropy of error can also be utilized as a cost function for adaptation (called generalized MCC, GMCC). Clearly, the correntropy with Gaussian kernel corresponds to the generalized correntropy with $p = 2$. In this work, we focus mainly on developing a new recursive SAF algorithm using GMCC in (7).

## III. Sparse Recursive Generalized Maximum Correntropy Criterion Algorithms

### A. System description

Sparse system model is considered as follows. The input vector $u(n) = [u_1, u_2, \ldots, u_M]^T$ is sent over an FIR system with parameter vector $w = [w_1, w_2, \ldots, w_M]^T$ with sparse form, where $M$ is the size of the memory. It is assumed that the system parameters are real-valued, and most of them are zero. The received signal $d(n)$ is modeled as

$$d(n) = w^T u(n) + \nu(n) \quad (8)$$

where $\nu(n)$ is an additive noise that is independent of $u(n)$.

### B. Recursive GMCC algorithm with SPC

To derive the recursive GMCC (RGMCC) algorithm with SPC for SSI problem, we define a new cost function by regularizing the GMCC by a sparsity penalty term as

$$J(n) = \frac{1}{T_p} \sum_{t}^{T_p} \exp \left( -\tau |d - w^T u(n)\|^2 \right) - \rho s(n) \quad (9)$$

where $w(n) = [w_1(n), \ldots, w_M(n)]^T$ is the estimated weight vector at the $n$-th time instant, $s(n)$ denotes a sparsity penalty term, and $\rho > 0$ is a regularization parameter that balance the tradeoff between GMCC and sparsity penalty. The optimal parameter $w_n$ can be obtained by maximizing the cost function of (9). That is

$$w_n = \arg \max_w J(n) \quad (10)$$

Then, the gradient of (9) with respect to the weight vector $w(n)$ can be computed and setting it to zero yields

$$\sum_{i=1}^{M} \exp \left( -\tau |e(i)|^2 \right) |e(i)|^{p-2} e(i) w(i) = \rho s'(n) \quad (11)$$

where $e(i) = d(i) - w(n)^T u(i)$, $T$ is transpose operator, and $s'(n)$ is the derivative of $s(n)$ . From (11), we have
\[
\sum_{i=0}^{\infty} \lambda^i \exp(-\lambda |\epsilon(i)|^p) |\epsilon(i)|^{p-2} u(i)u'(i) w(n) \\
\sum_{i=0}^{\infty} \lambda^i \exp(-\lambda |\epsilon(i)|^p) |\epsilon(i)|^{p-2} u(i)u'(i) \rho s'(n)
\]  
\tag{12}
\]
Both sides of the equation (12) have similar structure, we define \( \Psi(n) \) and \( \Phi(n) \) as (13), respectively.
\[
\Psi(n) = \sum_{i=0}^{\infty} \lambda^i \exp(-\lambda |\epsilon(i)|^p) |\epsilon(i)|^{p-2} u(i) u'(i) \\
\Phi(n) = \sum_{i=0}^{\infty} \lambda^i \exp(-\lambda |\epsilon(i)|^p) |\epsilon(i)|^{p-2} u(i) u'(i) \rho s'(n)
\]
\tag{13}
One can see that an extra exponential factor is included in \( \Psi(n) \) and \( \Phi(n) \), which is introduced by the GMCC. According to (13), Eq. (12) can be reformulated in matrix form as
\[
\Psi(n)w(n) = \Phi(n) - \rho s'(n)
\]
\tag{14}
Then, we have
\[
w(n) = \Psi'(n) \Phi(n) - \rho s'(n) = \Psi'(n) r(n)
\]
\tag{15}
where \( r(n) = \Phi(n) - \rho s'(n) \). To avoid the difficulty of computing the inverse of \( \Psi(n) \), we represent the \( \Psi(n) \) recursively as
\[
\Psi(n) = \sum_{i=0}^{\infty} \lambda^i \exp(-\lambda |\epsilon(i)|^p) |\epsilon(i)|^{p-2} u(i) u'(i) + \exp(-\lambda |\epsilon(n)|^p) |\epsilon(n)|^{p-2} u(n) u'(n)
\]
\tag{16}
As the iterations increasing, the \( (n-1)^{\text{st}} \) weight coefficient closes to the \( n^{\text{th}} \), i.e., \( w(n-1) \rightarrow w(n), n \rightarrow \infty \). Under this condition, we have the recursive form of \( \Psi(n) \) as
\[
\Psi(n) = \lambda \Psi(n-1) + f(\epsilon(n))w(n)u'(n)
\]
\tag{17}
Similarly, the recursive form of \( \Phi(n) \) is derived as
\[
\Phi(n) = \lambda \Phi(n-1) + f(\epsilon(n))w(n)d(n)
\]
\tag{19}
Combining (12) and (16) yields
\[
r(n) = \Phi(n) - \rho s'(n)
\]
\tag{20}
\[
\lambda \Phi(n-1) - \lambda \Phi s'(n-1)
\]
\[= \lambda r(n-1) + f(\epsilon(n))w(n)d(n) - \rho \lambda s'(n-1)
\]
\tag{21}
To this end, we assume that the sign of the weight values do not change significantly in a single time step, i.e. \( s(n) \) approach to \( s(n-1) \). Hence, we approximate (20) by
\[
r(n) = \lambda r(n-1) + f(\epsilon(n))w(n)d(n) - \rho \lambda s'(n-1)
\]
\tag{21}
Now, we give a matrix form of equation (17). First, some new notations are defined as
\[
A = \Psi(n), B = \lambda \Psi(n-1), C = \sqrt{f(\epsilon(n))} \sqrt{w(n)}D = I
\]
where \( I \) is a unit matrix. So, the matrix form of the equation (17) becomes
\[
A = B + CD'C
\]
\tag{22}
Using the matrix inversion theorem and considering the equation (17), we obtain the inverse of \( \Psi(n) \) as
\[
\Psi^{-1}(n) = \lambda \Psi^{-1}(n-1) + \frac{f(\epsilon(n))\Psi^{-1}(n-1)w(n) - \lambda \Psi^{-1}(n-1)w(n)}{\lambda + f(\epsilon(n))w(n)}
\]
\tag{23}
For a simple description of (22), we introduce \( \Omega(n) \) and \( K(n) \) as
\[
\Omega(n) = \Psi^{-1}(n), K(n) = \frac{\Omega(n-1)w(n)}{\lambda + f(\epsilon(n))w(n) \Omega(n-1)w(n)}
\]
\tag{24}
where \( K(n) \) and \( \Omega(n) \) are the extended kalman gain vectors similar to those in RLS. After some tedious computation, we have
\[
w(n) = \Psi^{-1}(n) r(n)
\]
\tag{25}
The developed recursive GMCC with SPC is summarized in Table 1.

**Table 1.** RGMCC with SPC

| Initialization: | \( w(0) = 0 \), \( \Omega(0) = \epsilon^{-1} \), \( \rho, \lambda, \tau, p \) |

For \( n = 1, 2, \ldots \):

- Do:

  1. \( y(n) = w'(n-1)u(n) \)
  2. \( e(n) = d(n) - y(n) \)
  3. \( K(n) = \frac{\Omega(n-1)w(n)}{\lambda + f(\epsilon(n))w(n)} \)

- \( w(n) = w(n-1) + f(\epsilon(n))w(n-1)u(n) \)

- \( \rho \lambda^{n-1} - 1 \) when \( \rho = 0 \) and \( \rho = 2 \), one can get a solution arbitrarily close to that of the \( l_0 \)-norm, which is a small positive number. The gradient in vector form is

\[
\mathbf{s}'(n) = \frac{1}{M \sigma_0 \sqrt{2\pi}} \mathbf{w}(n) \mathbf{\epsilon} \mathbf{\epsilon}' e^{-\frac{w(n) \mathbf{\epsilon} \mathbf{\epsilon}' w(n)}{2\sigma_0^2}}
\]
\tag{26}
In this case, the proposed algorithm is called CIMRGMCC.

**Remark 2:** From the Table 1, we know that the ZARGMCC and CIMRGMCC will reduce to ZARLS [14] and CIMRLS when \( f(\epsilon(n)) = 1 \) and \( p = 2 \). While they will reduce to RMCC [29] when \( \rho = 0 \) and \( p = 2 \). The computational complexity of the sparse RGMCC algorithm is almost the same as that of the sparse RLS algorithm, and the only extra computational effort
needed is to calculate the term $f(e(n))$, which is obviously not expensive.

**Remark 3:** Currently a lot of work on sparsity-promoting optimization problems, such as alternating direction, proximal splitting methods and so on, have been applied to robust exemplar extraction [3] and signal recovery [34]. The proposed cost function can be optimized by these methods in the future study.

**IV. NUMERICAL SIMULATIONS**

In this section, we evaluate the proposed ZARGMCC and CIMRGMCC, compared with the RLS, ZARLS, CIMRLS, MCC, ZAMCC, and CIMMCC algorithms. The SSI problems are considered under non-Gaussian noise environments. Simulation results are obtained by averaging over 100 independent Monte Carlo (MC) runs and 5000 iterations are run for each MC. The input signal is assumed to be a white Gaussian process with zero mean and unit variance. The impulsive response of the sparse system model with $M=30$ is illustrated in Fig. 2. In the following simulations, the parameters are experimentally chosen such that all the algorithms have almost the same convergence speed, and achieve performance as far as possible.

The convergence performance will be evaluated by the mean square deviation (MSD) given by

$$\text{MSD}(n) = E \left[ |w_n - w(n)|^2 \right]$$  \hspace{1cm} (27)

**Simulation 1:** Performance comparison under impulsive noise with $\alpha$-stable distribution

In this simulation, we consider the noise process with $\alpha$-stable distribution, which provides a good model for many impulsive noises [35]. Its characteristic function is given by

$$p(t) = \exp \left[ j\beta t - \gamma^\alpha \right] \left[ 1 + j\beta \text{sgn}(t)S(t, \alpha) \right]$$ \hspace{1cm} (28)

in which

$$S(t, \alpha) = \begin{cases} \tan \frac{\alpha \pi}{2} & \text{if } \alpha \neq 1 \\ \frac{2}{\alpha} \log |t| & \text{if } \alpha = 1 \end{cases}$$ \hspace{1cm} (29)

where $\alpha \in (0, 2]$ is the characteristic factor, $-\infty < \delta < +\infty$ is the location parameter, $\beta \in [-1, 1]$ is the symmetry parameter, and $\gamma > 0$ is the dispersion parameter. The characteristic factor $\alpha$ measures the tail heaviness of the distribution. The bigger $\gamma$ is, the heavier the tail is. We denote the parameters vector of the noise model as $V_{\alpha, \delta, \beta} = (\alpha, \beta, \gamma, \delta)$.

First, we show the comparison results among all the algorithms in terms of the MSD. The noise parameter vector is $V_{\alpha, \delta, \beta} = (1.4, 0.04, 4.0)$. For all recursive algorithms, $\epsilon$ is set to 1, and $\lambda$ is 0.995. The convergence curves are shown in Fig. 3. One can observe that the sparse type algorithms achieve faster convergence rate and better steady-state performance than the non-sparse algorithms (RLS, MCC and RGMCC). In particular, the proposed ZARGMCC and CIMRGMCC achieve better performance than other algorithms, and this confirms the advantage of the recursive GMCC. To our delight, we see that the CIMRGMCC outperforms ZARGMCC, and this result also reflects the superiority of the CIM approaching to $l_1$-norm. It is worth noting that the RLS, ZARLS and CIMRLS are unstable in this situation because the second order moment of error is sensitive to impulsive disturbance (actually infinite for $\alpha$-stable noise). In our simulation, we also evaluate the elapsed CPU time for a single MC run. The elapsed CPU times are 0.1092006, 0.1092008, 0.1404009, 0.0468002, 0.0468003, 0.1092006, 0.0624004, 0.1248008, and 0.1404009 seconds for RLS, ZARLS, CIMRLS, MCC, ZAMCC, CIMMCC, RGMCC, ZARGMCC, and CIMRGMCC, respectively. From the results, we know that the proposed algorithms are also efficient in terms of time complexity. In the subsequent simulations, the RLS-type algorithms will be omitted due to instability, and without mentioned otherwise the parameters are set to the same values in this simulation.

![Fig. 2. Impulsive response of the sparse system.](image)

![Fig. 3. Tracking and steady-state behaviours.](image)
parameter is important for the CIMRGMCC. How to select an optimal value is a topic for future study.

Simulation 2: Performance comparison under noise with mixture of different distributions

We further consider a noise model with form $e(n) = (1 - a(n))A(n) + a(n)B(n)$, where $a(n)$ is an independent and identically distributed binary process with an occurrence probability $0 \leq c \leq 1$, $A(n)$ is a noise process with smaller variance, and $B(n)$ is another noise process with substantially much larger variance to represent impulsive disturbances. The noise processes $A(n)$ and $B(n)$ are mutually independent and they are both independent of $a(n)$. In this simulation, $c$ is set at 0.06, and $B(n)$ is assumed to be a white Gaussian process with zero-mean and variance 15. For the noise $A(n)$, two distributions are considered: 1) binary distribution over $\{-1, 1\}$ with probability mass $p(x=-1) = p(x=1) = 0.5$; 2) Laplace distribution with zero-mean and unit variance. The convergence curves in terms of MSD are shown in Fig7-8. From simulation results one can observe: 1) The RGMCC family algorithms are much more stable (or robust) than the LMP family algorithms; 2) The GMCC with $p \neq 2$ may outperform significantly the original MCC algorithm. In particular, the GMCC with $p = 6$ achieves the best performance when $A(n)$ is of binary distribution; 3) The algorithms with the CIM penalty have lower steady-state MSD than algorithms with other sparse penalty terms.

**Fig. 4. Performance evaluation versus parameter $\gamma$**

**Fig. 5. Steady-state MSD of CIMRGMCC for variations in $\alpha$ and $\tau$**

Fourth, we evaluate the performance of the CIMRGMCC in view of different regularization parameters $\rho$ (1, 2, 3, 4, 5, 7, 10, 15 and 20). The convergence curves are shown in Fig.5. We observe that the lowest MSD can be obtained when $\rho$ is equal to 3 or 4. From the result, we know that $\rho$ controls the power of the penalty term $s(n)$, and suitable regularization parameter should be selected for CIMRGMCC to get the best performance. Too bigger or smaller value for $\rho$ will lead to poor performance, which means that $\rho$ should be selected within a certain range.

**Fig. 6. Performance evaluation versus parameter $\rho$**

**Fig. 7. Convergence curves when $A(n)$ is of Binary distribution**

**Fig. 8. Convergence curves when $A(n)$ is of Laplace distribution**
V. CONCLUSION

In this paper, we proposed the RGMCC algorithms with SPC (including ZARGMCC and CIMRGMMC), which can be applied in SSI under non-Gaussian noise environment. The new algorithms were developed by regularizing the combination of the GMCC (with forgetting factor) and the SPC. Simulation results show that the proposed methods can achieve robust steady-state error and outperform other SAF algorithms.

ACKNOWLEDGMENTS

This work was supported in part by National Natural Science Foundation of China (no. 61372152), and the Doctoral Scientific Research Foundation of Xi’an University of Technology (No.I03-256081611).

REFERENCES