Maximum Total Correntropy Adaptive Filtering against Heavy-Tailed Noises

Fei Wang\textsuperscript{a}, Yicong He\textsuperscript{a,∗}, Shiyuan Wang\textsuperscript{b}, Badong Chen\textsuperscript{a}

\textsuperscript{a}Institute of Artificial Intelligence and Robotics, Xi’an Jiaotong University, Xi’an, China, 710049
\textsuperscript{b}The School of Electronic and Information Engineering, Southwest University, Chongqing, China, 400715

Abstract

Total least squares (TLS) method has been widely used in errors-in-variables (EIV) modeling tasks in which both input and output data are disturbed by noises, and adaptive filtering algorithm using TLS has shown significantly superior performance to classical least squares (LS) method in EIV system. The TLS essentially extends the minimum mean square error (MMSE) criterion to EIV model, which, however, may work poorly when noise is non-Gaussian (especially heavy-tailed distribution). Recently, an information theoretic learning (ITL) based minimum total error entropy (MTEE) adaptive filtering algorithm has been proposed, which extends the minimum error entropy (MEE) criterion to EIV model and shows desirable performance in non-Gaussian noise environments. However, due to complex mathematical expression, MTEE is computationally expensive and difficult to carry out the theoretical analysis. In this paper, we propose a new ITL-based criterion called maximum total correntropy (MTC) and develop a gradient-based MTC adaptive filtering algorithm. We analyze theoretically the local stability and steady-state performance of the proposed algorithm. Simulation results confirm the theoretical analysis and show the superior performance of MTC in heavy-tailed noises. Further, simulation comparisons between MTC and MTEE are presented. Compared with the MTEE, the MTC is mathematically more tractable and computationally much simpler while achieving similar or even better performance.

Keywords: adaptive filtering, maximum total correntropy, robust method, parameter estimation

\footnotesize{∗}Corresponding author

Email addresses: wfx@xjtu.edu.cn (Fei Wang), heyicong@stu.xjtu.edu.cn (Yicong He), wangsy.swu@gmail.com (Shiyuan Wang), chenbd@xjtu.edu.cn (Badong Chen)
1. Introduction

Adaptive filtering algorithms are powerful modeling tools for estimating the unknown parameters of a model from noisy observed data such that the filter (the model) output gets as close to the desired response as possible. For linear systems, the least mean square (LMS) algorithm has been widely used due to its simplicity and easy of implementation. LMS is based on the well-known minimum mean square error (MMSE) criterion and can achieve desirable performance when dealing with standard regression model, where the input data have been observed without noises [1]. However, in practice, due to sampling errors, human errors, or modeling errors, the observed input data can also be disturbed by some noises. Thus the assumption of standard regression model does no longer hold and the performance of LMS will degrade seriously. To solve this problem, the errors-in-variables (EIV) model is proposed to describe the unknown system in which all the variables (both input and output) are perturbed by noises [2]. A well-known approach for EIV model is the total least square (TLS) method [3], which obtains the solution by minimizing the perturbation of errors in all variables. By applying the TLS method to adaptive filtering, several algorithms have been proposed such as total LMS (TLMS) [4, 5], gradient-descent TLS (GD-TLS) [6, 7] and recursive total least squares (RTLS) [8].

Similar to the MMSE criterion, the TLS also only considers the second-order statistics of error, which is rational under Gaussian assumption. However, in many real-world situations, due to impulsive disturbances or large outliers, the noise distribution will be no longer Gaussian and TLS based algorithms may suffer from poor accuracy in estimation. To improve the robustness in non-Gaussian noise environments, some methods beyond TLS should be proposed. Recently, an information theoretic learning (ITL) [9, 10] based adaptive filtering algorithm called minimum total error entropy (MTEE) has been proposed [11]. As an extension of the minimum error entropy (MEE) criterion [12, 13, 14, 15, 16, 17], MTEE takes entropies of both input and output errors into consideration, which can outperform the original TLS in heavy-tailed noises. However, since MTEE adopts the parzen window method to estimate the error distributions at each iteration, it is computationally very expensive especially when the sliding window length is large. Moreover, due to complex mathematical expression, rigorous analysis for MTEE is very difficult and complicated [11].

In recent years, an another ITL-based criterion called maximum correntropy criterion (MCC) has been gained increasing attention [18, 19, 20, 21, 22, 23]. Correntropy is a generalized correlation measure between two random variables induced by a kernel function [24]. With a Gaussian kernel, correntropy involves all the even moments of the error and is insensitive to outliers. Compared with the error entropy, correntropy based algorithms are computationally much simpler and easier to implement. Correntropy has been successfully used in many applications such as classification [25], shape matching [26], and robust principal component analysis [27]. In adaptive filtering domain, various MCC
based algorithms have been developed, which can achieve robust performance for standard regression model in non-Gaussian noises [28]. However, without consideration of the input noises, the existing MCC based algorithms still can’t work well for EIV model.

In this paper we propose a new criterion called maximum total correntropy (MTC), and develop a gradient-based MTC adaptive filtering algorithm. Taking advantages of both TLS method and MCC criterion, the MTC algorithm can deal well with non-Gaussian noises in both input and output data. The local stability and steady-state performance of the proposed algorithm are also analyzed. Further, the steady state performance of MTC algorithm under small step-size and uncorrelated input signal is investigated, showing an important conclusion that given a small step size, the change of output noise variance has little effect on the steady state mean square derivation (MSD). Finally, simulations are carried out to verify the theoretical results and demonstrate the desirable performance of the new algorithm. In general, our main contributions are summarized as follows:

i) An efficient and robust adaptive algorithm for EIV model with low computational cost is developed.

ii) Comprehensive mathematical analysis on local stability and steady-state performance is given.

iii) Theoretical analysis and desirable performance are confirmed by simulation results. In particular, our MTC can achieve similar or even better performance than the higher complexity MTEE algorithm.

The organization of this paper is as follows. In Section 2 we briefly review the TLS method and GD-TLS algorithm. MTC criterion and the corresponding gradient based adaptive MTC algorithm are proposed in Section 3. In Section 4, we analyze the local stability and steady-state performance. Simulation results are then presented in Section 5, and finally the conclusion is given in Section 6.

2. Brief review of Total Least Square Adaptive filtering

Suppose we are given a sequence \( \{x(i), y(i)\}_{i=1}^{N} \), where \( x(i) \in \mathbb{R}^{L \times 1} \) is the input vector at time \( i \), and \( y(i) \in \mathbb{R} \) is the corresponding scalar-valued output. The relationship between input and output can be described as a mapping function \( y(i) = f(x(i)) \). For a linear system, the mapping function can be represented as a vector \( h \in \mathbb{R}^{L \times 1} \) so that

\[
y(i) = h^T x(i)
\]

(1)

In an EIV model, both input and output signals are assumed to be observed in noises, described as

\[
\tilde{x}(i) = x(i) + u(i) \\
\tilde{y}(i) = y(i) + v(i)
\]

(2)

where \( u(i) \) and \( v(i) \) stand for the input noise and output noise with covariance matrix \( \sigma_u^2 I_{L \times L} \) and variance \( \sigma_v^2 \), respectively. The \( u(i) \) and \( v(i) \) are supposed
to be wide-sense stationary and mutually uncorrelated. The goal of an adaptive filtering algorithm is to estimate the desired mapping \( h \) by iteratively updating the weight vector \( w(i) \) based on the observed sequence \( \{\tilde{x}(j), \tilde{y}(j)\}_{j=1}^{N} \).

When \( \sigma_o^2 = \sigma_i^2 \), in a geometric view, by defining the augmented data
\[
\tilde{d}(i) = [\tilde{y}(i) \; \tilde{x}(i)^T]^T
\]
the TLS solution can be obtained by minimizing the sum of squared perpendicular distances from the point \( \tilde{d}(i) \) to the hyperplane [8]
\[
w^T x^o = 0
\]
where \( x^o \in \mathbb{R}^{(L+1) \times 1} \) and the augmented weight vector \( w^a = [1 \; -w^T]^T \). Equivalently, the TLS solution can be solved by minimizing the following cost function over \( w \in \mathbb{R}^L \times 1 \)
\[
J_{gd-tls}(w) = \frac{1}{N} \sum_{i=1}^{N} \frac{\|w^o^T \tilde{d}(i)\|^2}{\sqrt{w^a^T w^a}} = \frac{1}{N} \sum_{i=1}^{N} \frac{w^o^T \tilde{d}(i) \tilde{d}^T(i) w^a}{w^a^T D w^a}
\]
(3)
where \( \| \cdot \| \) denotes the Euclidean norm. When \( \sigma_o^2 \neq \sigma_i^2 \), Eq.(3) can be rewritten as
\[
J_{gd-tls}(w) = \frac{1}{N} \sum_{i=1}^{N} \frac{w^o^T \tilde{d}(i) \tilde{d}^T(i) w^a}{w^a^T D w^a}
\]
(4)
where
\[
D = \text{diag} \{\beta, 1, \ldots, 1\}
\]
with \( \beta = \sigma_o^2 / \sigma_i^2 \) is the weighting matrix that normalizes the noise variances. Although the distance measure of Eq.(4) is not exactly the distance from \( \tilde{d}(i) \) to the hyperplane, the obtained solution is still referred to as the TLS solution [8].

In the cost function of the GD-TLS adaptive algorithm, the sample mean operation in Eq.(4) is replaced with the ensemble average (expectation) operation over both input and output noises[5, 6, 7]
\[
J_{gd-tls}(w) = E \left[ \frac{w^o^T \tilde{d}(i) \tilde{d}^T(i) w^a}{w^a^T D w^a} \right]
\]
(5)

In order to learn the relationship between GD-TLS and LMS in the view of residual which is defined as the deviation between the observed value and predicted value, we rewrite the cost function in Eq.(5) as
\[
J_{gd-tls}(w) = E \left[ \frac{(\tilde{y}(i) - w^T \tilde{x}(i))^2}{w^a^T D w^a} \right] = E \left[ \frac{e^2(i)}{||\tilde{w}||^2} \right]
\]
(6)
where \( \tilde{w} = \sqrt{D} w^a = [\sqrt{\beta} - w^T]^T \) is the modified augmented weight vector and \( e(i) = \tilde{y}(i) - w^T \tilde{x}(i) \) is the computed deviation (residual) at iteration \( i \). Compared with cost function of the LMS
\[
J_{lims}(w) = E \left[ e^2(i) \right]
\]
(7)
one can observe that LMS minimizes the expectation of the squared residuals, while GD-TLS in fact minimizes the expectation of weighted squared residuals by multiplying a weighting factor \( \frac{1}{\|\bar{w}\|^2} \) \[2, 29]\.

### 3. Maximum Total Correntropy Algorithm

Correntropy is a local and nonlinear similarity measure between two random variables within a "window" in the joint space determined by the kernel width. Given two random variables \( X \) and \( Y \), the correntropy is defined by \[24\]

\[
V(X, Y) = E[\kappa(X, Y)] = \int \kappa(x, y)dF_{XY}(x, y) \tag{8}
\]

where \( \kappa_\sigma \) is a shift-invariant Mercer kernel, and \( F_{XY}(x, y) \) denotes the joint distribution function of \( (X, Y) \). Given a finite number of samples \( \{x_i, y_i\}_{i=1}^N \), the correntropy can be approximated as

\[
\hat{V}(X, Y) = \frac{1}{N} \sum_{i=1}^{N} \kappa(x_i, y_i) \tag{9}
\]

In general, the kernel function of correntropy \( \kappa(x, y) \) is the Gaussian kernel

\[
\kappa(x, y) = G_\sigma(e) = \exp\left(-\frac{e^2}{2\sigma^2}\right) \tag{10}
\]

where \( e = x - y \) and \( \sigma \) is the kernel width.

Compared with the LMS (or MSE) which is based on the second-order statistics of the error, correntropy involves all the even moments of the difference between \( X \) and \( Y \). Replacing the second-order measure in Eq.(7) with correntropy measure leads to the maximum correntropy criterion (MCC) \[28\]. The MCC solution is obtained by maximizing the following utility function

\[
J_{mcc}(w) = E\left[G_\sigma(e(i))\right] \tag{11}
\]

We remark that Eq.(11) has similar formulation with Welsch’s function which was originally introduced in \[30\]. Maximizing the MCC utility function in Eq.(11) is equivalently minimizing the Welsch’s cost function.

Inspired by TLS method and MCC, we propose in this work a new criterion called maximum total correntropy (MTC), which obtains the optimal solution by maximizing the following utility function

\[
J_{mtc}(w) = E\left[G_\sigma\left(\frac{e(i)}{\sqrt{\bar{w}^T \bar{w}}}\right)\right] = E\left[\exp\left(-\frac{e^2(i)}{2\sigma_{mtc}^2 \|\bar{w}\|^2}\right)\right] \tag{12}
\]

where \( \sigma_{mtc} \) is the kernel width. The relationship between four cost (utility) functions are illustrated in Fig.1. One can observe that MTC solution actually minimizes the mean of nonlinear weighted squared residuals.
Taking the gradient of \( J_{\text{mtc}}(w) \) leads to

\[
\hat{g}_{\text{mtc}}(w) = \frac{\partial J_{\text{mtc}}(w)}{\partial w} = \frac{1}{\sigma_{\text{mtc}}^2} E \left[ \frac{1}{\|\tilde{w}\|} \exp \left( -\frac{e^2(i)}{2\sigma_{\text{mtc}}^2 \|\tilde{w}\|^2} \right) \right] 
\times \left( \|\tilde{w}\|^2 e(i) \tilde{x}(i) + e^2(i) w \right)
\]

(13)

In a gradient-based adaptive filtering algorithm, the gradient is usually approximated by its instantaneous value. By dropping the expectation operator in Eq.(13), we obtain the following instantaneous gradient

\[
\hat{g}_{\text{mtc}}(w) = \frac{1}{\sigma_{\text{mtc}}^2} \exp \left( -\frac{e^2(i)}{2\sigma_{\text{mtc}}^2 \|\tilde{w}\|^2} \right) \frac{\|\tilde{w}\|^2 e(i) \tilde{x}(i) + e^2(i) w}{\|\tilde{w}\|^4} 
\]

(14)

Thus the weight update of gradient-based MTC adaptive algorithm at time \( i \) can be derived as

\[
w(i + 1) = w(i) + \eta \hat{g}_{\text{mtc}}(w(i)) = w(i) + \mu \exp \left( -\frac{e^2(i)}{2\sigma_{\text{mtc}}^2 \|\tilde{w}(i)\|^2} \right) \frac{\|\tilde{w}(i)\|^2 e(i) \tilde{x}(i) + e^2(i) w(i)}{\|\tilde{w}(i)\|^4} 
\]

(15)

where \( \mu = \frac{\eta}{\sigma_{\text{mtc}}^2} \) is the step-size parameter and \( \eta > 0 \). The pseudo code of the MTC algorithm is summarized in Algorithm 1. Note that since correntropy is not concave, to ensure the convergence, we follow the method described in [9, 23] and train the filter with GD-TLS first and then switch to MTC.
Algorithm 1 MTC Algorithm

Initialization
choose step-size $\eta > 0$, Gaussian kernel width $\sigma_{mtc}$
GD-TLS iteration number $C_T$
initial weight vector $\mathbf{w}(1)$

Computation
for $i = 1, 2, ...$ do
  %compute the output
  $d(i) = \mathbf{w}^T(i)\hat{\mathbf{x}}(i)$
  %compute the error
  $e(i) = \hat{y}(i) - d(i)$
  if $i \leq C_T$ then
    %update the weight vector using GD-TLS
    $a(i) = \|\bar{\mathbf{w}}(i)\|^2 e(i)\hat{\mathbf{x}}(i) + e^2(i)\mathbf{w}(i)$
    $\|\bar{\mathbf{w}}(i)\|^4$
    $\mathbf{w}(i + 1) = \mathbf{w}(i) + \mu a(i)$
  else
    %update the weight vector using MTC
    $a(i) = \exp\left(-\frac{e^2(i)}{2\sigma_{mtc}^2\|\bar{\mathbf{w}}(i)\|^2}\right)\|\bar{\mathbf{w}}(i)\|^2 e(i)\hat{\mathbf{x}}(i) + e^2(i)\mathbf{w}(i)$
    $\|\bar{\mathbf{w}}(i)\|^4$
    $\mathbf{w}(i + 1) = \mathbf{w}(i) + \mu a(i)$
  end if
end for

From Eq.(15) one can observe that if $e(i)$ is relatively small compared with kernel width $\sigma_{mtc}$, the approximation

$$\exp\left(-\frac{e^2(i)}{2\sigma_{mtc}^2\|\bar{\mathbf{w}}(i)\|^2}\right) \approx 1$$

will hold. In this case, the weight update can be approximated as

$$\mathbf{w}(i + 1) = \mathbf{w}(i) + \mu \frac{\|\bar{\mathbf{w}}(i)\|^2 e(i)\hat{\mathbf{x}}(i) + e^2(i)\mathbf{w}(i)}{\|\bar{\mathbf{w}}(i)\|^4}$$

which is the weight update of typical GD-TLS algorithm [5, 6, 7]. Additionally, let $\mu(i) = \mu \exp\left(-\frac{e^2(i)}{2\sigma_{mtc}^2\|\bar{\mathbf{w}}(i)\|^2}\right)$. Eq.(15) can be rewritten as

$$\mathbf{w}(i + 1) = \mathbf{w}(i) + \mu(i) \frac{\|\bar{\mathbf{w}}(i)\|^2 e(i)\hat{\mathbf{x}}(i) + e^2(i)\mathbf{w}(i)}{\|\bar{\mathbf{w}}(i)\|^4}$$

Since $\exp\left(-\frac{e^2(i)}{2\sigma_{mtc}^2\|\bar{\mathbf{w}}(i)\|^2}\right)$ is bounded with $[0, 1]$, the step-size $\mu(i)$ is also bounded with $[0, \mu]$. Thus MTC can be viewed as a variable step-size GD-TLS (VSS-GD-TLS) algorithm. In particular, due to the property of Gaussian function, a large residual caused by outlier may lead to a small step size and slow
down the weight update process. Therefore, the bad influence of the outliers can be efficiently reduced. Since the variable step size depends on the weight vector $\mathbf{w}(i)$ at each iteration, the convergence behavior and steady state performance are very different from that of the GD-TLS.

**Remark 1.** The noise variance ratio $\beta = \sigma_o^2/\sigma_i^2$ in MTC is calculated without outliers. In the paper we assume that $\beta$ is already known. In practice, there are several ways to estimate the noise variance ratio $\beta$ from heavy-tailed noises.

1. One can manually eliminate the outliers and then compute $\beta$. This method is feasible in practice since most of the outliers can be easily identified.
2. If the occurrence probability of the input and output outliers are equal, one can reject a fixed percentage of the noise values with large amplitudes (for example top 5%) and then compute $\beta$.

### 4. Local Stability Analysis

In this section, we carry out the performance analysis of the proposed MTC algorithm.

Taking the Taylor expansion of the MCC utility function in Eq.(12), we can obtain

$$J_{mtc}(\mathbf{w}) = E \left[ \exp \left( -\frac{e^2(i)}{2\sigma_{mtc}^2 \| \tilde{\mathbf{w}} \|^2} \right) \right] = \sum_{n=0}^{\infty} \frac{(-1)^n}{2^n n!} E \left[ \left( \frac{e^2(i)}{\sigma_{mtc}^2 \| \tilde{\mathbf{w}} \|^2} \right)^n \right]$$

(18)

Therefore, if $\sigma_{mtc}$ is selected so that the value of $\frac{e^2(i)}{\sigma_{mtc}^2 \| \tilde{\mathbf{w}} \|^2}$ is relatively small for all $e(i)$, the high order terms ($n \geq 2$) can be omitted. In this case, the proposed MTC can be approximated as the GD-TLS, thus the local stability and steady state performance of MTC will be similar to that of GD-TLS. Further, when the kernel width $\sigma$ tends to infinity, the equation

$$\lim_{\sigma \to +\infty} -2\sigma^2 \exp \left( -\frac{x^2}{2\sigma^2} \right) + 2\sigma^2 = x^2$$

(19)

will hold. Thus maximizing the MTC utility function will be equal to minimizing GD-TLS cost function in Eq.(6).

When $\sigma_{mtc}$ is selected so that the high order terms ($n \geq 2$) cannot be omitted, the stability and convergence will no longer be the same as GD-TLS. Moreover, since $J_{mtc}(\mathbf{w})$ is not a convex or concave function, the global minimum or maximum is hard to analyze. In particular, given a certain $\mathbf{w}$, $J_{mtc}(\mathbf{w})$ can be viewed as a function of residual $e$

$$h(e) = E \left[ \exp \left( -\frac{e^2}{2\sigma_{mtc}^2 \| \tilde{\mathbf{w}} \|^2} \right) \right]$$

(20)

where $h(e)$ is strictly concave in the range of

$$e \in (-\sigma_{mtc} \sqrt{\| \tilde{\mathbf{w}} \|^2}, \sigma_{mtc} \sqrt{\| \tilde{\mathbf{w}} \|^2})$$
In this case, a compromised way is to analyze the local property of $J_{mte}(u)$ instead of global property. Below we show that a rigorous mathematical analysis under Gaussian noise is available, which guarantees the local stability and convergence of MTC algorithm in Gaussian noise case. Rigorous analysis is very hard for non-Gaussian cases because it is difficult to evaluate the involved expectations. In the following, we instead analyze the influence of outliers on local stability and show that local stability will not be highly affected in heavy-tailed noise environments. The superior performance of the MTC under heavy-tailed noise will be illustrated by the simulations in Section 5.

For tractability, we utilize the following assumptions during the local stability analysis.

A1 The elements of the input vector $x(i)$ are i.i.d. processes.

A2 The elements of the input noise $u(i)$ and the output noise $v(i)$ are i.i.d. processes and independent of the input vector $x(i)$.

A3 The covariance matrix $R = E[x(i)x^T(i)]$ is positive definite with full rank.

Remark 2. Assumption A1 and A2 are called the independence assumption [31] which is widely used in analyzing adaptive filtering algorithms [1] and can be justified in many practical situations. The assumption A3 ensures the uniqueness of the optimal solution. To get rigorous mathematical analysis, without mentioned, in the following we assume that $v(i)$ and the elements of $u(i)$ are zero-mean Gaussian.

4.1 Local maximum

From Eq.(13), the gradient at $h$ can be written as

$$g_{mte}(h) = \frac{1}{\sigma_{mte}^2 \|\tilde{h}\|^2} E \left( \exp \left( - \frac{e_h^2(i)}{2\sigma_{mte}^2 \|\tilde{h}\|^2} \right) (e_h(i)) \tilde{x}(i) \right)$$

$$+ \frac{h}{\sigma_{mte}^2 \|\tilde{h}\|^4} E \left( \exp \left( - \frac{e_h^2(i)}{2\sigma_{mte}^2 \|\tilde{h}\|^2} \right) e_h^2(i) \right) \tag{21}$$

where the expectations are taken over both input and output noises, $\|\tilde{h}\|^2 = ||h||^2 + \beta$ and

$$e_h(i) = \tilde{y}(i) - h^T \tilde{x}(i) = v(i) - h^T u(i) \tag{22}$$

In Gaussian noise environment, $e_h(i) = v(i) - h^T u(i)$ will also be Gaussian distributed with

$$E(e_h(i)) = E(v(i)) - E(h^T u(i)) = 0 \tag{23}$$

$$E(e_h^2(i)) = E(v^2(i)) + E\left(h^T u(i)\right)^2$$

$$= \sigma_o^2 + ||h||^2 \sigma_i^2$$

$$= \sigma_i^2 ||\tilde{h}\|^2 \tag{24}$$
that is $e_h \sim N\left(0, \sigma_i^2 \|\bar{h}\|^2\right)$. Using integral method to compute the expectation (the detailed derivations are provided in the Appendix A), we obtain

$$E\left(\exp\left(-\frac{e_h^2(i)}{2\sigma_{mtc}^2\|\bar{h}\|^2}\right) e_h^2(i)\right)$$

$$= \|\bar{h}\|^2 \left(\frac{\sigma_{mtc}^2}{\sigma_i^2 + \sigma_{mtc}^2}\right)^{\frac{3}{2}} \sigma_i^2$$

(25)

$$E\left(\exp\left(-\frac{e_h^2(i)}{2\sigma_{mtc}^2\|\bar{h}\|^2}\right) e_h(i)\tilde{x}(i)\right)$$

$$= -\left(\frac{\sigma_{mtc}^2}{\sigma_i^2 + \sigma_{mtc}^2}\right)^{\frac{3}{2}} \sigma_i^2 \lambda$$

(26)

where $\lambda = \frac{\sigma_{mtc}^2}{\sigma_i^2 + \sigma_{mtc}^2}$. Substituting Eq.(25) and Eq.(26) to Eq.(21) can we obtain $g_{mtc}(h) = 0$. That is, $h$ is a critical point of $J_{mtc}(w)$.

To verify $h$ is the local maximum point of $J_{mtc}(w)$, one need further to prove the Hessian of $J_{mtc}(w)$ is negative-definite. To simplify the analysis process, here we use some results from analysis of GD-TLS in [7]. From Eq.(14) we can obtain the relation between instantaneous gradient of GD-TLS and MTC

$$\hat{g}_{mtc}(w) = -\frac{1}{2\sigma_{mtc}^2} \exp\left(-\frac{e^2(i)}{2\|\bar{w}\|^2}\right) \hat{g}_{gd-tls}(w)$$

(27)

where

$$\hat{g}_{gd-tls}(w) = -\frac{2}{\|\bar{w}\|^2} \left(\tilde{x}(i) + e^2(i)w\right)$$

(28)

is the instantaneous gradient of $J_{gd-tls}(w)$ in Eq.(5). Then the instantaneous Hessian matrix $\hat{H}_{mtc}(w)$ can be calculated as

$$\hat{H}_{mtc}(w) = \frac{\partial \hat{g}_{mtc}(w)}{\partial w^T}$$

$$= \frac{1}{4\sigma_{mtc}^4} \exp\left(-\frac{e^2(i)}{2\sigma_{mtc}^2\|\bar{w}\|^2}\right) \hat{g}_{gd-tls}(w)\hat{g}_{gd-tls}(w)^T$$

(29)

$$- \frac{1}{2\sigma_{mtc}^2} \exp\left(-\frac{e^2(i)}{2\sigma_{mtc}^2\|\bar{w}\|^2}\right) \hat{H}_{gd-tls}(w)$$

where $\hat{H}_{gd-tls}$ denotes the instantaneous Hessian matrix in GD-TLS algorithm.
given by
\[
H_{\text{gd-tls}}(w) = \frac{\partial \hat{g}_{\text{gd-tls}}(w)}{\partial w^T} = \frac{2}{||w||^2} \left( \hat{z}(i)\hat{z}^T(i) - \hat{g}_{\text{gd-tls}}(w)w^T - \hat{g}_{\text{gd-tls}}(w^T)w - \frac{e^2(i)}{||w||^2} \right) \tag{30}
\]

Taking the expectation of both sides of Eq.(30) results in the Hessian matrix at \( h \)
\[
H_{\text{mtc}}(h) = E \left( \hat{H}_{\text{mtc}}(h) \right) = \frac{1}{4\sigma_{\text{mtc}}^4} E \left( \exp \left(-\frac{e_h^2(i)}{2\sigma_{\text{mtc}}^2 ||\hat{h}||^2} \right) \hat{g}_{\text{gd-tls}}(h)\hat{g}_{\text{gd-tls}}^T(h) \right) - \frac{1}{2\sigma_{\text{mtc}}^2} E \left( \exp \left(-\frac{(e_h(i))^2}{2\sigma_{\text{mtc}}^2 ||\hat{h}||^2} \right) \hat{H}_{\text{gd-tls}}(h) \right) \tag{31}
\]

Again, using integral method to compute the expectation gives
\[
E \left( \exp \left(-\frac{e_h^2(i)}{2\sigma_{\text{mtc}}^2 ||\hat{h}||^2} \right) \hat{g}_{\text{gd-tls}}(h)\hat{g}_{\text{gd-tls}}^T(h) \right) = \frac{4}{||\hat{h}||^4} \left( ||\hat{h}||^2 \lambda^2 \sigma_i^2 (R + \sigma_i^2 I) - \lambda^2 \sigma_i^4 hh^T \right) \tag{32}
\]
\[
E \left( \exp \left(-\frac{(e_h(i))^2}{2\sigma_{\text{mtc}}^2 ||\hat{h}||^2} \right) \hat{H}_{\text{gd-tls}}(h) \right) = \frac{2}{||\hat{h}||^2} \left( \lambda^2 (R + \sigma_i^2 I) - \lambda^2 \frac{\sigma_i^4 hh^T}{\sigma_{\text{mtc}}^2 ||\hat{h}||^2} - \lambda^2 \sigma_i^2 I \right) \tag{33}
\]

By substituting Eq.(32) and Eq.(33) to Eq.(31), the Hessian of \( J_{\text{mtc}}(w) \) at critical point \( h \) can be derived as
\[
H_{\text{mtc}}(h) = \frac{1}{\sigma_{\text{mtc}}^4 ||\hat{h}||^2} \left( ||\hat{h}||^2 \lambda^2 \sigma_i^2 (R + \sigma_i^2 I) - \lambda^2 \sigma_i^4 hh^T \right) \tag{34}
\]

Based on assumption A2 that the covariance matrix \( R \) is positive definite and has all positive eigenvalues, \( H_{\text{mtc}}(h) \) will have all negative eigenvalues, hence is negative-definite. Thus, since \( h \) is the critical point of \( J_{\text{mtc}}(w) \) and the Hessian of \( J_{\text{mtc}}(w) \) is negative-definite, we can conclude that \( h \) is the local maximum of \( J_{\text{mtc}}(w) \).
The surfaces of $J_{mte}(w)$ with different kernel widths $\sigma_{mte}$ are plotted in Fig.2. We use the numerical integration to approximate the expectations. The parameters are set to $L = 2$, $\sigma_{o}^2 = \sigma_{i}^2 = 0.1$, $R = I$, $h = [-0.5, 0.7]^T$. One can observe that $J_{mte}(w)$ has a unique local maximum at point $w = h$ within a large range of $w$, which guarantees a unique solution in a wide range of $w$ near the desired $h$. Moreover, since MTC is a non-convex optimization problem, we can only prove that $h$ is a local maximum. In order to get a global optimal solution, the initial condition should be chosen carefully, or one can use some other criterion such as GD-TLS to train the adaptive filter first to make sure the solution is near the global optimal solution.

**Remark 3.** It should be noticed that in Eq.(25) and Eq.(26), $\exp\left(-\frac{e_h^2(i)}{2\sigma_{3cc}^2\|h\|^2}\right)$, $e_h(i)$ and $u(i)$ are not pairwise independent, thus the expectations cannot be simply multiplied with each other. Moreover, we rewrite $\lambda$ as

$$\lambda = \frac{1}{\frac{\sigma_{o}^2}{\sigma_{mte}^2} + 1} \quad (35)$$
When $\sigma_{mtc}^2$ is much larger than input noise variance $\sigma_i^2$, $\lambda$ will be close to 1 and
\[
E \left( \exp \left( -\frac{e_h^2(i)}{2\sigma_{mtc}^2\|\hat{h}\|^2} \right) e_h^2(i) \right) \\
\approx \|\hat{h}\|^2 \sigma_i^2 \\
= E (e_h^2(i))
\]
will hold. In this case the approximation $\exp \left( -\frac{e_h^2(i)}{2\sigma_{mtc}^2\|\hat{h}\|^2} \right) \approx 1$ in [11] can be applied. However, when the kernel width $\sigma_{mtc}$ is small, $\sigma_{mtc}^2$ cannot be neglected, thus the local behaviour will be highly affected by different $\sigma_{mtc}$.

**Remark 4.** In Gaussian noise, since $h$ is the local maximum point, the MTC can offer the unbiased solution. In non-Gaussian noise environments, however, the unbiased estimation may not be satisfied due to impulsive disturbances or large outliers. Nevertheless, compared with GD-TLS, the proposed MTC can efficiently reduce the effects of outliers. Consider a heavy-tailed non-Gaussian case where the original Gaussian noises $v(i)$ and $u(i)$ are contaminated with outliers with probability $p \ll 1$. We rewrite the $k$-th element of the gradient vector in Eq.(21) as
\[
g_{mtc}(h) = \frac{1}{\sigma_{mtc}^2\|\hat{h}\|^2} E[G_1(e_h(i))\tilde{x}(i)] \\
+ \frac{h}{\sigma_{mtc}^2\|\hat{h}\|^2} E[G_2(e_h(i))]
\]
where
\[
G_1(x) = \exp(-\frac{x^2}{2\sigma_{mtc}^2\|\hat{h}\|^2})x \\
G_2(x) = \exp(-\frac{x^2}{2\sigma_{mtc}^2\|\hat{h}\|^2})x^2
\]
It is easy to prove that $G_1(x)$ and $G_2(x)$ are bounded. Moreover, given a relatively small $\sigma_{mtc}$, $G_1(x)$ and $G_2(x)$ will nearly be zero when $x$ is large. Thus a large value of $e_h(i)$ caused by outliers will lead to small values of $G_1(e_h(i))$ and $G_2(e_h(i))$. So the effects of large outliers will be efficiently reduced, resulting a small degree of bias, i.e. the estimated vector is near the desired solution.

The analysis of $\hat{H}_{mtc}(h)$ is similar to the above analysis. Thus in general one can see that large outliers will have little influence on the local maximum property of $J_{mtc}(w)$, showing robustness in local stability under heavy-tailed noise environments.
4.2. Local mean behaviour

To analyse the local stability, we assume that the MTC algorithm has converged to the vicinity of local maximum after sufficient iteration times, thus the estimated weight vector $\mathbf{w}(i)$ is close to desired weight vector $\mathbf{h}$. In Eq.(14) the the expectation of gradient $\mathbf{g}_{\text{mtc}}(\mathbf{w})$ is approximated by instantaneous gradient $\hat{\mathbf{g}}_{\text{mtc}}(\mathbf{w})$, leading to the gradient error defined by

$$e_g(\mathbf{w}(i)) = \hat{\mathbf{g}}_{\text{mtc}}(\mathbf{w}(i)) - \mathbf{g}_{\text{mtc}}(\mathbf{w}(i)) \tag{38}$$

Substituting Eq.(38) to Eq.(15) yields

$$\mathbf{w}(i+1) = \mathbf{w}(i) + \eta \mathbf{g}_{\text{mtc}}(\mathbf{w}(i)) + \eta e_g(\mathbf{w}(i)) \tag{39}$$

Subtracting both side of Eq.(39) from $\mathbf{h}$ gives

$$\tilde{\mathbf{w}}(i+1) = \tilde{\mathbf{w}}(i) - \eta \mathbf{g}_{\text{mtc}}(\mathbf{w}(i)) - \eta e_g(\mathbf{w}(i)) \tag{40}$$

where $\tilde{\mathbf{w}}(i) = \mathbf{h} - \mathbf{w}(i)$ is the weight error vector at iteration time $i$. Since $J_{\text{mtc}}(\mathbf{w})$ is twice continuously differentiable in a neighborhood of a line segment between points $\mathbf{w}(i)$ and $\mathbf{h}$, we can use Theorem 1.2.1 in [32] and obtain

$$\mathbf{g}_{\text{mtc}}(\mathbf{w}(i)) = \mathbf{g}_{\text{mtc}}(\mathbf{h}) - \mathbf{H}_{\text{mtc}}(\mathbf{h} - t\tilde{\mathbf{w}}(i)) \mathbf{dt} \tilde{\mathbf{w}}(i) \tag{41}$$

where $(a)$ is obtained by approximating $h - t\tilde{w}(i) \approx h$ since $t\tilde{w}(i)$ is relatively small compared with $h$ for $t \in [0,1]$ near the local maximum. Substituting Eq.(41) into Eq.(40) yields

$$\tilde{\mathbf{w}}(i+1) \approx \tilde{\mathbf{w}}(i) + \eta \mathbf{H}_{\text{mtc}}(\mathbf{h}) \tilde{\mathbf{w}}(i) - \eta e_g(\mathbf{w}(i)) \tag{42}$$

Under assumption A1, $\tilde{\mathbf{w}}(i)$ is independent of $\mathbf{x}(i)$ since $\tilde{\mathbf{w}}(i)$ is a function of past input data and noise. Thus taking the expectations of both sides of Eq. (42) gives the local mean weight error recursion

$$\mathbb{E}[	ilde{\mathbf{w}}(i+1)] \approx \mathbb{E}[(\mathbf{I} + \eta \mathbf{H}_{\text{mtc}}(\mathbf{h})) \tilde{\mathbf{w}}(i) - \eta e_g(\mathbf{w}(i))] \tag{43}$$

To guarantee the local convergence to steady state in the mean sense, the magnitudes of all the eigenvalues of matrix $\mathbf{I} + \eta \mathbf{H}_{\text{mtc}}(\mathbf{h})$ must be less than unity, leading to the following condition

$$|1 + \eta \frac{\lambda^2 \rho(k)}{\sigma^2_{\text{mtc}} \| \tilde{\mathbf{h}} \|^2}| < 1 \tag{44}$$
where \( \{\rho(k)\}_{k=1}^L \) are the eigenvalues of \( R \). Since \( R \) is positive-definite, \( \rho(k) \) will be positive for all \( k \), thus the step-size \( \mu = \frac{2}{\sigma_{\text{mtc}}^2} \) in Eq. (44) should be selected as
\[
0 < \mu < \max \left\{ \frac{2\|\bar{h}\|^2}{\lambda^2 \rho(k)} \right\}
\]
and the local convergence will be guaranteed in mean sense.

4.3. Mean square stability and steady state performance

At the vicinity of local maximum, the gradient error \( e_g(w(i)) \) in Eq. (38) can be approximated as
\[
e_g(w(i)) \approx e_g(h) = \hat{g}_{\text{mtc}}(h)
\]
To obtain the energy conservation, we follow the method of [7] and first multiply both sides of Eq. (42) with arbitrary matrix \( Q \in \mathbb{R}^{L \times L} \)
\[
Q \tilde{w}(i+1) \approx Q(I + \eta H_{\text{mtc}}(h)) \tilde{w}(i) - \eta Q \hat{g}_{\text{mtc}}(h)
\]
Assumption A1 guarantees the independence between \( \tilde{w}(i) \) and \( \tilde{x}(i) \). Thus squaring both sides of Eq. (47) gives the following mean energy conservation
\[
E\left[\|\tilde{w}(i+1)\|_V^2\right] \approx E\left[\|\tilde{w}(i)\|_\Sigma^2\right] + \eta^2 E\left[\|\hat{g}_{\text{mtc}}(h)\|_V^2\right]
\]
where
\[
\|m\|_A^2 = m^T A m
\]
\[
V = Q^T Q
\]
\[
\Sigma = (I + \eta H_{\text{mtc}}(h)) V (I + \eta H_{\text{mtc}}(h))
\]
After some algebra, one can obtain the following mean-square relation
\[
E\left[\|\tilde{w}(i+1)\|_F^2\right] \approx E\left[\|\tilde{w}(i)\|_F^2\right] + \eta^2 vec\{M\} \theta
\]
where
\[
F = [I + \eta H_{\text{mtc}}(h)] \otimes [I + \eta H_{\text{mtc}}(h)]
\]
\[
\theta = vec\{V\}
\]
and
\[
M = E \left( \hat{g}_{\text{mtc}}(h) \hat{g}_{\text{mtc}}^T(h) \right)
\]
\[
= \frac{1}{4\sigma_{\text{mtc}}^4} E \left( \exp \left( -\frac{e_h^2(i)}{\sigma_{\text{mtc}}^2 \|\bar{h}\|^2} \right) \hat{g}_{\text{mtc}}(h) \hat{g}_{\text{mtc}}^T(h) \right)
\]
\[
= \frac{\sigma_i^2}{\sigma_{\text{mtc}}^4 \|\bar{h}\|^4} \left( \frac{\sigma_{\text{mtc}}^2}{2\sigma_i^2 + \sigma_{\text{mtc}}^2} \right)^3 \left( \|\bar{h}\|^2 (R + \sigma_i^2 I) - \sigma_i^2 hh^T \right)
\]
where \( vec\{\cdot\} \) is the matrix vectorization operator.
For simplicity, we define the eigenvalues vector \( \zeta = [\zeta(1), \ldots, \zeta(L^2)]^T \in R^{L^2 \times 1} \) formed by all eigenvalues of \( F \). Thus, according to the property of Kronecker product, \( \zeta \) can be computed as

\[
\zeta = \text{vec}\{(1_{L \times 1} + \eta \frac{\lambda^2 \rho}{\sigma^2_{\text{mte}} \|\bar{h}\|^2})(1_{L \times 1} + \eta \frac{\lambda^2 \rho}{\sigma^2_{\text{mte}} \|\bar{h}\|^2})^T\}
\]

where \( \rho = [\rho(1), \ldots, \rho(L)]^T \in R^{L \times 1} \) is formed by all eigenvalues of \( R \) and \( 1_{L \times 1} \) is the all-one vector with length \( L \). When Eq.(44) is satisfied, all eigenvalues of \( F \) will be also less than 1, and the mean square stability in Eq.(49) can be guaranteed. Thus the condition of step-size \( \mu \) in Eq.(45) will guarantee both the mean and mean-square stability.

When the iteration number is large enough, Eq.(49) can be written as

\[
E\left[ \|\hat{w}(\infty)\|_\theta^2 \right] \approx E\left[ \|\hat{w}(\infty)\|_{F\theta}^2 \right] + \eta^2 \text{vec}\{M\} \theta
\]

we can obtain the steady state MSD as

\[
E\left[ \|\hat{w}(\infty)\|_\theta^2 \right] \approx \eta^2 \text{vec}\{M\}^T(I - F)^{-1} \text{vec}\{I\}
\]

4.4. Steady state MSD under small step-size and uncorrelated input signal

Here we give a more explicit expression for Eq.(54) under assumptions that step size is small and all channels of input signal are pairwise uncorrelated. These assumptions are commonly used in practice. For uncorrelated input signal, the covariance matrix \( R \) is actually a diagonal matrix with diagonal entries \( \{\phi(k)\}_{k=1}^L \), and \( F \) is also a diagonal matrix with diagonal entries \( \{\phi(k)\}_{k=1}^L \).

In the following analysis, we use operator diag\{\cdot\} in two ways. For matrix \( A \), diag\{\( A \)\} extracts the diagonal entries of \( A \) into a vector. While for vector \( m \), diag\{\( m \)\} generates a diagonal matrix whose entries are elements of \( m \). Thus, by defining the vectors

\[
\phi = [\phi(1), \phi(2), \ldots, \phi(L)]^T
\]

\[
\phi^2 = [\phi^2(1), \phi^2(2), \ldots, \phi^2(L)]^T
\]

\[
\varphi = [\varphi(1), \varphi^2(2), \ldots, \phi^2(L)]^T
\]

the relation between \( \phi \) and \( \varphi \) can be derived as

\[
\varphi = \text{diag}\{\text{vec}\{M\}\}
\]

\[
= \text{diag}\{[I + \eta H_{\text{mte}}(h)] \otimes [I + \eta H_{\text{mte}}(h)]\}
\]

\[
= \text{vec}\{(1_{L \times 1} - \frac{\mu \lambda^2}{\|\bar{h}\|^2} \phi)(1_{L \times 1} - \frac{\mu \lambda^2}{\|\bar{h}\|^2} \phi)^T\}
\]

\[
= \text{vec}\{1_{L \times L} - \frac{\mu \lambda^2}{\|\bar{h}\|^2} (\phi_{L \times 1} + \phi^T_{L \times 1}) + \frac{\mu^2 \lambda^3}{\|\bar{h}\|^2} \Phi \phi^T\}
\]

16
where $\phi_{L \times 1} = \text{col}\{\phi^T, \ldots, \phi^T\} \in R^{L \times L}$. Thus

$$(I - F)^{-1} \text{vec}\{I\}$$

$$= \text{vec}\left\{ \text{diag}\left( \frac{\mu \lambda^2}{\| \bar{h} \|^2} (\phi_{L \times 1} + \phi_{L \times 1}^T) - \frac{\mu^2 \lambda^3}{\| \bar{h} \|^2} \phi \phi^T \right) \right\}^{-1}$$

$$= \text{vec}\left\{ \text{diag}\left( \frac{2 \mu \lambda^2}{\| \bar{h} \|^2} \phi - \frac{\mu^2 \lambda^3}{\| \bar{h} \|^2} \phi^2 \right) \right\}^{-1}$$

(57)

According to the relationship of the vectorization operator and the matrix trace

$$\text{vec}\{A\}^T \text{vec}\{B\} = \text{tr}\{B^T A\}$$

(58)

we can obtain

$$(\text{vec}\{M\})^T (I - F)^{-1} \text{vec}\{I\}$$

$$= \text{tr}\left\{ \text{diag}\left( \frac{2 \mu \lambda^2}{\| \bar{h} \|^2} \phi - \frac{\mu^2 \lambda^3}{\| \bar{h} \|^2} \phi^2 \right)^{-T} M \right\}$$

(59)

If the step size $\mu$ is sufficient small, $\mu^2$ can be ignored and Eq.(59) can be simplified as

$$(\text{vec}\{M\})^T (I - F)^{-1} \text{vec}\{I\}$$

$$\approx \text{tr}\left\{ \frac{\| \bar{h} \|^2}{2 \mu \lambda^2} R^{-1} M \right\}$$

$$= \text{tr}\left\{ \frac{\sigma_i^2 a^2}{2 \mu \sigma_m^2 \| \bar{h} \|^2} \left( \| \bar{h} \|^2 (I + \sigma_i^2 R^{-1} I) - \sigma_o^2 R^{-1} h h^T \right) \right\}$$

(60)

where $a = \frac{\sigma_i^2 + \sigma_o^2}{\sigma_m^2}$. Thus the steady state MSD in Eq.(54) can be simplified as

$$E\left[ \| \tilde{w}(\infty) \|^2 \right] \approx \frac{\mu \sigma_i^2 a^2}{2} \left( 4 + \sigma_i^2 \sum_{k=1}^{L} \frac{1}{\phi(k)} \right) - \frac{\mu \sigma_i^2 a^2 L \| \bar{h} \|^2}{2} \sum_{k=1}^{L} \phi(k)$$

(61)

From Eq.(61) one can observe that under small step-size and uncorrelated input signal, the steady-state MSD is related to four parameters: the step-size parameter $\eta$, the noise variance $\sigma_i^2$ and $\sigma_o^2$, and the Gaussian kernel width $\sigma_m$. Note that the noise ratio $\beta$ can be computed from $\sigma_i^2$ and $\sigma_o^2$. For further analysis, we rewrite Eq.(61) as

$$E\left[ \| \tilde{w}(\infty) \|^2 \right] \approx A - B$$

(62)
where $A$ and $B$ stands for each term in the right side of Eq.(61). We can see that the output noise $\sigma^2_o$ only affects term $B$. Moreover, we compute the subtraction

$$B = \frac{\sigma^2_o \sum_{k=1}^{L} \frac{h^2(k)}{\sigma(k)}}{\|h\|^2 \left(4+\sigma^2_i \sum_{k=1}^{L} \frac{1}{\sigma(k)}\right)}$$  \hspace{1cm} (63)

If the the variances of each channel of input signal are the same, i.e. $R = \sigma^2_x I$, Eq.(63) can be simplified as

$$B = \frac{1}{4 \left(1 + \frac{\beta}{\|h\|^2}\right) \left(\frac{\sigma^2_x}{\sigma^2_i} + 1\right)}$$  \hspace{1cm} (64)

By defining the input SNR as

$$SNR_{in} = 10 \log \frac{\sigma^2_x}{\sigma^2_i}$$  \hspace{1cm} (65)

one can see that when the input SNR is not too slow, we will have $B \ll A$ and $A - B \approx A$. Moreover, if variance of each channel of input signal are different, we can get the following upper bound

$$B \leq \frac{1}{4 \left(1 + \frac{\beta}{\|h\|^2}\right) \min \{\phi(k)\}}$$  \hspace{1cm} (66)

If variances of all input channels are relatively large compared with input noise, $B \ll A$ will also be guaranteed. Thus the influence of different output noise variances on steady state MSD will be very small, i.e. the steady state performance is mainly depended on variance of input noise.

When $\sigma_{mte} \to \infty$, the MTC equals to GD-TLS, thus the steady state MSD of GD-TLS can be derived by simply setting $a = 1$ in Eq.(61). Thus in Gaussian noise, given the step size $\mu_{gd-tls}$ for GD-TLS, one can set the step size of MTC as

$$\mu = \left(\frac{\sigma^2_i + \sigma^2_{mte}}{2\sigma^2_i + \sigma^2_{mte}}\right)^{\frac{3}{2}} \mu_{gd-tls}$$  \hspace{1cm} (67)

to achieve the same steady state MSD with GD-TLS.

5. Simulation results

In this section, we present simulation results to verify the theoretical analysis and performance of the proposed algorithm.

5.1. Simulation settings

Consider a linear EIV system identification problem with filter length $L = 4$. The weight vector of the unknown system is assumed to be $h = [0.4 \ 0.7 \ -0.3 \ 0.5]^T$. Each element of the input vector $x(i)$ is independently generated from zero mean
Figure 3: Theoretical and simulated MSD with different input noise variances $\sigma_i^2$ ($\sigma_{mte} = 0.5, \beta = 1$)

Gaussian with unit variance. In the simulations, we consider a Gaussian mixture model (GMM) as non-Gaussian heavy-tailed noise model [13, 35]. The model combines two independent Gaussian noise processes, given by

$$v(i) = (1 - \theta(i))A(i) + \theta(i)B(i)$$

(68)

where $A(i)$ and $B(i)$ are Gaussian noise processes with variance $\sigma_A^2$ and $\sigma_B^2$ respectively. $\theta(i)$ is of binary distribution over $\{0, 1\}$, with probability $p(\theta(i) = 1) = c, p(\theta(i) = 0) = 1 - c$, where $c$ controls the occurrence probability of noise processes $A(i)$ and $B(i)$. In practice, $A(i)$ represents general noise disturbance with small variance, while $B(i)$ stands for outliers that occur occasionally with larger variance.

In the simulations, the filtering performance at iteration $i$ is evaluated by MSD which is approximated as the ensemble average of squared deviations $\|\mathbf{w}(i) - \mathbf{h}\|^2$ over 1000 Monte Carlo runs with the same data but different noise realizations. For MTC, the GD-TLS iteration number $C_T$ is set to 100.

5.2. Theoretical and simulated steady-state performance

In this part, we show the steady-state performance with different parameters, along with comparison between theoretical and simulated results. In the simulations, we set $c$ to be zero in accordance with the noise assumptions of theoretical analysis, and in each simulation the iteration number is large enough to
ensure the convergence to steady-state. The initial weight vector $w(1)$ is set to a zero vector. The steady-state MSD is computed as the mean value over last 1000 MSD values. The corresponding results are shown in Fig.3-5, which plot the steady-state MSD as a function of the step-size for different noise variances, noise ratios and kernel widths, respectively.

From the results one can observe that all the simulated MSD values are very close to the theoretical values, which confirms the validity of the theoretical analysis. In particular, Fig.4 shows that given the same input noise variance, different variances of output noise will lead to similar performance, which has been analyzed in Section 4. Moreover, as Fig.5 shows, when the kernel width $\sigma_{mtc}$ increases, the steady-state MSD of MTC algorithm is closer with steady-state MSD of GD-TLS algorithm.

5.3. Convergence Performance Comparison

In this part, first we compare the convergence performance of MTC with other algorithms. Apart from comparison with existing MCC, GD-TLS and LMS, we also make comparison with three new robust adaptive algorithms for EIV model, named total Huber (THU), total least absolute deviations (TLAD) and total fair (TF), whose cost functions are extended from Huber, LAD and fair estimation[36] respectively. The cost (utility) functions are shown in Table 1.

We perform several simulations by choosing different $c$ and $\sigma^2_B$ to represent different non-Gaussian heavy-tailed noise distributions. The variance of input
noise $\sigma_i^2$ and output noise $\sigma_o^2$ are set to both 0.04. The kernel width of MTC and MCC are set to $0.5/\sqrt{2}$ and 0.5 respectively to achieve the similar measurement of residuals at steady state. Similar to MTC, for MCC we train the filters with LMS method during the first 100 iterations. Moreover, for fair comparison, the GD-TLS is also performed for THU, TLAD and TF during the first 100 iterations. The step-sizes are chosen so that all the algorithms have almost the same initial convergence speed. For each Monte Carlo run, the entries of the the initial weight vector $w(1)$ are independently generated from zero mean Gaussian with variance 0.01. The average learning curves in terms of MSD are shown in Fig.6. From the simulation results we observe that both GD-TLS and TF achieve best performance under Gaussian noise ($c = 0$). In GMM noise with $\sigma_B^2 = 2$, MTC and THU get comparable good performance. When $\sigma_B^2 = 9$, MTC outperforms other algorithms significantly. In addition, when $\sigma_B^2$ is large, MCC may outperform TLS although MCC is not an algorithm proposed for EIV model.

Second, we compare the steady-state MSD of MTC and MTEE under different noise distributions. To evaluate the performance consistently, we fixed all the parameters during simulations. The kernel width of MTC and MTEE are both set to 0.5 and the window length of MTEE is chosen as 5, 20, 100 respectively. The $\sigma_i^2$ and $\sigma_o^2$ are both set to 0.01. The step-size of MTC and MTEE are set to 0.009, 0.0055 respectively, so that the steady-state performance of MTC and MTEE are nearly the same under Gaussian noise ($c = 0$). We gradually increase $\sigma_B^2$ and $c$ and compute steady-state MSD computed by averaging...
Table 1: Cost (Utility) Functions of the Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Cost (Utility) function</th>
</tr>
</thead>
<tbody>
<tr>
<td>LMS</td>
<td>$E\left[e^2(i)\right]$</td>
</tr>
<tr>
<td>GD-TLS</td>
<td>$E\left[e^2(i)|\bar{w}|^2\right]$</td>
</tr>
<tr>
<td>MCC</td>
<td>$E\exp\left(-\frac{e^2(i)}{2\sigma_m^2}\right)$</td>
</tr>
<tr>
<td>MTCC</td>
<td>$E\exp\left(-\frac{e^2(i)}{2\sigma_m^2|\bar{w}|^2}\right)$</td>
</tr>
<tr>
<td>THU</td>
<td>$E\rho\sigma\left(e^{(i)}|\bar{w}|^2\right)$</td>
</tr>
<tr>
<td>TLAD</td>
<td>$E\left</td>
</tr>
<tr>
<td>TF</td>
<td>$E\left[A^2\left</td>
</tr>
</tbody>
</table>

last 1000 MSD values. The setting of initial weight vector $\bar{w}(1)$ is the same as previous simulation. Fig.7 depicts the steady-state MSD surfaces of MTC and MTEE and Fig.8 shows the average learning curves for some representative noise environment. One can observe that the steady-state MSD of MTC and MTEE are very similar. Furthermore, when noise is heavy, MTC converges faster than MTEE.

We also compare the running time of MTC and MTEE. The results of the average execution time for each simulation are summarized in Table 2. The running times are measured on a PC with a 2.8GHz processor and 8 GB memory. One can observe that: 1) the running time of MTC is much shorter than MTEE. 2) as window length increases, the running time of MTEE grows significantly. The results confirms that MTC is very computational efficient compared with MTEE, while the performance is comparable or even better. Actually, the complexity of MTEE is $O(LD)$ where $D$ is the window length, while the complexity of MTC is only $O(L)$.

Table 2: Steady-state MSD performance and average running time of MTC and MTEE

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Windows length</th>
<th>Average running time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTC</td>
<td>1</td>
<td>0.0629</td>
</tr>
<tr>
<td>MTEE</td>
<td>5</td>
<td>0.4163</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>1.1305</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>4.8901</td>
</tr>
</tbody>
</table>
5.4. Performance with different kernel widths $\sigma_{mtc}$

The kernel width $\sigma_{mtc}$ controls affects the rate of convergence and robustness to impulsive disturbances or large outliers. In this part we analysis the influence of $\sigma_{mtc}$. We choose four kernel widths 0.2, 0.3, 0.5, 1.0 and evaluate the steady-state performance in different heavy-tailed noise environments. The step sizes are selected so that the steady-state MSD values with different kernel widths are almost the same under Gaussian noise ($c = 0$). The initial weight vector in this simulation is set to zero vector. The steady-state MSD surfaces with different kernel widths $\sigma_{mtc}$ are depicted in Fig.9. We can see that MTC works well for all four kernel widths, and small kernel width performs better than large kernel width. In particular, Fig.10 shows the steady-state MSD with different $\sigma_{mtc}$ at $c = 0.05$, we can observe that the performance is more stable when the kernel width is small, i.e there are few changes in steady-state MSD values when $\sigma_{mtc}$ is 0.2 and 0.3, while when $\sigma_{mtc} = 1$ the steady-state MSD varies from about -33dB up to -20.4dB. In conclusion, MTC can work well in a wide range of kernel width when dealing with different heavy-tailed noises.

6. Conclusion

We propose in this work a new criterion, maximum total correntropy (MTC), combining the advantages of the TLS method and correntropy, and develop a robust adaptive filtering algorithm, called the MTC algorithm. The developed algorithm can deal with the EIV modeling when both input and output signals are disturbed by some non-Gaussian noises (especially heavy-tailed noises). Theoretical analysis guarantees the local convergence and stability in Gaussian noises, and simulation results confirm the robustness of MTC algorithm in heavy-tailed noises. Compared with the robust MTEE algorithm, the MTC is much simpler and easier to implement while achieving similar or even better performance. We also demonstrate how the kernel width of total correntropy affects the performance of MTC. Although the MTC can significantly outperform the TLS in a wide range of kernel width under heavy-tailed noise situations, one would expect a better choice with better performance. How to select an optimal kernel width is an interesting topic for future study.

Acknowledgements

This work was supported by National Natural Science Foundation of China under Grant No.91648208 and No.61231018, program of introducing talents of discipline to university of China under Grant B13043 and National Science and Technology support program of China under Grant 2015BAH31F01.
Appendix A. Derivation of expectation

Some calculations of integral are based on the following formulas

\[ \int_{-\infty}^{+\infty} x^{2n+1} e^{-ax^2} dx = 0 \]
\[ \int_{-\infty}^{+\infty} x^{2n} e^{-ax^2} dx = 2 \sqrt{\pi} (2n)! \left( \frac{1}{4a} \right)^{\frac{2n+1}{2}} \]  

(A.1)

Since elements of the input noise vector \( u(i) \) and the output noise \( v(i) \) are zero-mean i.i.d. zero-mean Gaussian, the linear combination \( \delta(i) = e_h(i) + h^T \Phi u(i) \) where

\[ \Phi = \text{diag}\{p(1), p(2), ..., p(L)\}, \{p(m)\}_{m=1}^L \in \{0, 1\} \]

is also Gaussian with distribution

\[ \delta \sim N \left( 0, \sigma_i^2 \left( \|\bar{h}\|_2^2 - h^T \Phi h \right) \right) \]

The integral results are summarized in Table A.3. For simplicity, we define

\( G_m = \exp \left( -\frac{e_h^2(i)}{2\sigma_{mtc}^2 \|\bar{h}\|^2} \right) \) and

\( \lambda = \frac{\sigma_{mtc}^2}{\sigma_i^2 + \sigma_{mtc}^2} \). Each integral result is represented as a index with capital letter. Hence, it is easy to calculate the following equations as

\[ E \left( \exp \left( -\frac{e_h^2(i)}{2\sigma_{mtc}^2 \|\bar{h}\|^2} \right) e_h(i) \tilde{x}(i) \right) = E \left( \exp \left( -\frac{e_h^2(i)}{2\sigma_{mtc}^2 \|\bar{h}\|^2} \right) e_h(i) (x(i) + u(i)) \right) \]
\[ = \lambda E(x(i)) + F \]
\[ = F \]
\[ = -\lambda^2 \sigma_i^2 h \]  

(A.2)
\[ E \left( \exp \left( -\frac{\epsilon_h^2(i)}{2\sigma_{mtc}^2 \| \bar{h} \|^2} \right) \hat{g}_{gd-tls}(h) \hat{g}^T_{gd-tls}(h) \right) \]

\[ = \frac{4}{\| \bar{h} \|^4} E \left( \exp \left( -\frac{\epsilon_h^2(i)}{2\sigma_{mtc}^2 \| \bar{h} \|^2} \right) \left( \epsilon_h^2(i) \bar{x}(i) \bar{x}^T(i) + 2\epsilon_h^2(i) \bar{x}(i) \bar{h}^T \| \bar{h} \|^2 + \epsilon_h^2(i) \bar{h} \| \bar{h} \|^4 \right) \right) \]

\[ = \frac{4}{\| \bar{h} \|^4} \left( DR + E (x(i)) G^T + GE (x^T(i)) + J \right) \]  
(A.3)

\[ = \frac{4}{\| \bar{h} \|^4} \left( DR + J + 2h^T \bar{h} (B + H) + \bar{h} h^T \bar{h} \| \bar{h} \|^4 \right) \]

\[ = \frac{4}{\| \bar{h} \|^4} \left( \| \bar{h} \|^2 \lambda^2 \sigma_i^2 \left( R + \sigma_i^2 I \right) - \lambda^2 \sigma_i^4 \bar{h} h^T \right) \]

\[ E \left( \exp \left( -\frac{\epsilon_h^2(i)}{2\sigma_{mtc}^2 \| \bar{h} \|^2} \right) H_{gd-tls} \right) \]

\[ = \frac{2}{\| \bar{h} \|^2} E \left( \exp \left( -\frac{\epsilon_h^2(i)}{2\sigma_{mtc}^2 \| \bar{h} \|^2} \right) \left( \bar{x}(i) \bar{x}^T(i) - \hat{g}_{gd-tls}(h) \bar{h}^T - \hat{g}_{gd-tls}^T(h) \bar{h} - \epsilon_h^2(i) \right) \right) \]

\[ = \frac{2}{\| \bar{h} \|^2} \left( CR + E (x(i)) K^T + KE (x^T(i)) + I \right) \]  
(A.4)

\[ + 2\sigma_{mtc}^2 g_{mtc}(h) h^T + 2\sigma_{mtc}^2 g_{mtc}^T(h) h - \frac{DF}{\| \bar{h} \|^2} \]

\[ = \frac{2}{\| \bar{h} \|^2} \left( CR + I - \frac{DF}{\| \bar{h} \|^2} \right) \]

\[ = \frac{2}{\| \bar{h} \|^2} \left( \lambda^2 \left( R + \sigma_i^2 I \right) - \lambda^2 \frac{\sigma_i^4 \bar{h} h^T}{\sigma_{mtc}^2 \| \bar{h} \|^2} - \lambda^2 \sigma_i^4 I \right) \]

References


Figure 6: Average learning curves under different noise distributions
Figure 7: Steady-state MSD surfaces of MTC and MTEE with different noise distributions

Figure 8: Average learning curves of MTC and MTEE with different noise distributions
Figure 9: Steady-state MSD surfaces with different kernel widths

Figure 10: Steady-state MSD with different kernel widths at $c = 0.05$
<table>
<thead>
<tr>
<th>Index</th>
<th>Expectation</th>
<th>Integral Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$E(G_m e_h(i))$</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>$E(G_m e_h^3(i))$</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>$E(G_m)$</td>
<td>$\lambda \frac{1}{2}$</td>
</tr>
<tr>
<td>D</td>
<td>$E(G_m e_h^2(i))$</td>
<td>$|h|^2 \lambda \frac{1}{2} \sigma_i^2$</td>
</tr>
<tr>
<td>E</td>
<td>$E(G_m e_h^4(i))$</td>
<td>$3\sigma_i^4 |h|^4 \lambda \frac{5}{2}$</td>
</tr>
<tr>
<td>F</td>
<td>$E(G_m e_h(i)u(i))$</td>
<td>$-\lambda \frac{3}{2} \sigma_i^2 h$</td>
</tr>
<tr>
<td>G</td>
<td>$E(G_m e_h^2(i)u(i))$</td>
<td>0</td>
</tr>
<tr>
<td>H</td>
<td>$E(G_m e_h^3(i)u(i))$</td>
<td>$-3 |h|^2 \sigma_i^4 \lambda \frac{5}{2} h$</td>
</tr>
<tr>
<td>I</td>
<td>$E(G_m u(i)u^T(i))$</td>
<td>$\lambda \frac{1}{2} \sigma_i^2 I - \lambda \frac{3}{2} \frac{\sigma_i^4 hh^T}{\sigma_{mtc}^2 |h|^2}$</td>
</tr>
<tr>
<td>J</td>
<td>$E(G_m e_h^2(i)u(i)u^T(i))$</td>
<td>$\sigma_i^4 \lambda \frac{3}{2} \left(|h|^2 I + \frac{(2\sigma_{min}^2 - \sigma_i^2)hh^T}{\sigma_i^2 + \sigma_{min}^2}\right)$</td>
</tr>
<tr>
<td>K</td>
<td>$E(G_m u(i))$</td>
<td>0</td>
</tr>
</tbody>
</table>