Proceedings of the 37th Chinese Control Conference
July 25-27, 2018, Wuhan, China

# Multi-task Total Least-Squares Adaptation over Networks

Zhongfa Wang[1], Lijuan Jia[†1], Zijiang Yang[2]

1. School of Information and Electornics, Beijing Institute of Technology, Beijing 100081, P. R. China
E-mail: jlj@bit.edu.cn

2. Ibaraki University, Ibaraki, Japan

**Abstract:** Collaborative parameter estimation is a significant application of distributed multi-agent network. In practical scenarios, there are many multi-task oriented applications that the networks have multiple optimum parameter vectors to be estimated. Considering the condition that the input and output of agents are corrupted by additive noises, the network can be modeled as the multi-task errors-in-variables (M-EIV) problem. Total least-squares (TLS) method is a typical solution to the EIV problem for it can minimize the perturbation both in input and output data. In this paper, we study the problem of unbiased parameter estimation over multi-task networks whose nodes' inputs are corrupted by white noises. We propose a novel multi-task TLS (M-TLS) algorithm which can reach consistent unbiased estimation. Simulation results show that the proposed algorithms can achieve consistent and unbiased estimation.

**Key Words:** Multi-agent network, Collaborative Parameter Estimation, Total Least-Squares, Multi-Task Learning

## 1 Introduction

The study of distributed multi-agent networks is a hot topic recently[1–4] and the networks are widely used in wireless, environmental monitoring and medical applications, etc[5, 6]. Their merits such as scalability, robustness and communication reduction make them suitable for performing decentralized tasks such as parameter estimation. Compared with single-task network, multi-task network is more suitable to the complex practical application scenarios. In this paper, we study the problem of multi-task parameter estimation over distributed adaptive networks.

Several algorithms over distributed networks have been intensively studied to solve the single-task parameter estimation problems[7–9]. However, many practical applications are multi-task oriented inherently[10], which means that the agents have to estimate multiple parameters. A diffusion least mean squares(LMS) algorithm over multi-task networks has been proposed in [11]. The research analyzes the stochastic behavior of diffusion LMS algorithm under the condition that it takes multiple tasks. The diffusion LMS network has been developed in an unsupervised way, which means that the nodes are unaware of the relations with theirs neighbors. Each node is allowed to choose its neighbors by adjusting the combination weights adaptively. As the recursive least-squares(RLS) has better convergence rate and estimation accuracy than LMS, a distributed sparse multi-task RLS algorithm has been put forth in [12]. The paper has pointed out that sparsity exists widely among nature signals and big data. A decentralized on-line alternating direction method of multipliers technique and an on-line sub-gradient method have been applied to the distributed RLS to construct the multi-task algorithm. Besides, there are also several researches on the topic of multi-task adaptive networks[13–15].

Though the multi-task algorithms have been well developed, most of them do not consider the condition that the input data is interrupted by additive noise, which is also called multi-task errors-in-variables (M-EIV) problem. In this situation, the total least-squares (TLS) algorithm is considered as an efficient solution to the EIV problem. Recently, a distributed TLS algorithm with adaptive intertask cooperation has been studied in [16]. It has the ability to undertake the multiple tasks and it is used to obtain unbiased estimation. But it needs to take the derivative of the TLS cost function directly, which leads to large algorithm complexity. To obtain unbiased estimation and take multiple tasks simultaneously, we propose a novel multi-task TLS (M-TLS) algorithm.

In this paper, the parameter estimation is considered as the process of solving the overdetermined equation. The error emendation vector and matrix are added to the both sides of the equation to correct the perturbations in output and input data. The M-TLS tries to minimize both the error emendation vector and matrix. But unlike the classic criterion, our proposed algorithm does not require the transmission or inversion of matrices. It is developed in a stochastic gradient descent fashion. The M-TLS is designed under diffusion criterion because the diffusion strategy is more robust to the link failure. Simulation results show that the suggested algorithm can obtain unbiased estimates in the presence of noisy inputs. It also illustrates that our clustering model meets the requirements of multi-task parameter estimation problems.

Our contributions are summarized as follows. Firstly, we analyze the M-EIV problem and illustrate that M-TLS can reach a consistent estimation and propose an M-TLS algorithm. We use the proximal gradient descent method to fulfill the computation instead of singular value decomposition (SVD). This makes M-TLS require less computational resource and have fast convergence rate. Secondly, we analyze the topology of multi-task distributed networks and construct the proposed algorithms in diffusion fashion. Through the clustered model, each cluster is assigned a task and the whole network can undertake the multiple parameter estimation objectives. At last, we make numerical simulations and evaluate the performances of algorithms.

[†]Jia Lijuan is the correspondence author.

This work is supported by National Natural Science Foundation (NNSF) of China under Grant No. 61671063 and No. 61771046.

*Notation:*

| | |
|---|---|
| $\boldsymbol{x}$ | Boldface lower case letters denote vectors. |
| $\boldsymbol{P}$ | Bold face upper case letters denote matrices. |
| $\mathbb{R}$ | Field of real valued numbers. |
| $\hat{\cdot}$ | Estimator. |
| $\cdot^o$ | True value of parameter. |
| $(\cdot)^{\mathrm{T}}$ | Vector or matrix transpose operator. |
| $\boldsymbol{J}(\cdot)$ | Cost function. |
| $\nabla(\cdot)$ | Differential operator. |
| $E[\cdot]$ | Statistical expectation. |
| $\|\cdot\|_2$ | Euclidean norm operator. |
| $N_k$ | Adjacent nodes of node $k$. |
| $C_p$ | Cluster $p$. |
| $C(k)$ | Cluster where node $k$ belongs. |

## 2 Problem Statement

We distinguish three types of networks depending on the relationship of parameter vectors $\boldsymbol{w}^o$ across the nodes[15]:

1) Single-task networks: All agents collaborate with each other to estimate the same parameter $\boldsymbol{w}^o$.
2) Multi-task networks: Each agent $k$ has its own optimum parameter $\boldsymbol{w}_k^o$.
3) Clustered multi-task networks: The networks are divided into several clusters. The optimum parameter vector $\boldsymbol{w}_{C_p}^o$ of cluster $C_p$ is different from other clusters yet all nodes of cluster $C_p$ have same optimum parameter vector.

The single-task networks and multi-task networks are particular cases of the clustered multi-task networks. When a cluster only contains one node, the clustered multi-task networks degenerate into multi-task networks. And when the nodes are all clustered together, the clustered multi-task networks degenerate into single-task networks. In this paper, we study the problem of multi-task parameter estimation over clustered multi-task networks.

Consider a multi-agent network shown as Figure 1. Each node is an $L$ order FIR system and the network consists $N$ nodes. The straight line between two neighboring nodes represents the bidirectional channel where information flows. Agents of cluster $C_p$ have the optimum parameter vector $\boldsymbol{w}_{C_p}^o$, which is different from the optimum parameter vector of other clusters. In this paper, the parameter estimation is modeled as an M-EIV problem, which emphasizes that both the input and output of each node are corrupted by noise.

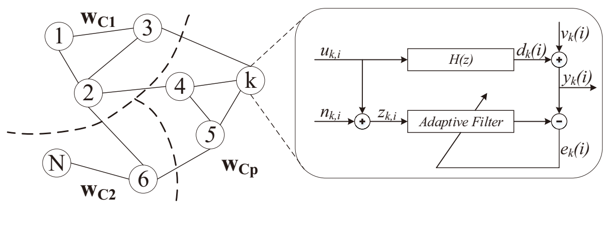


Fig. 1: EIV system identification model over network.

At time $i$, the node $k$ of cluster $C_p$ can be modeled as

$$\begin{cases} d_k(i) = \boldsymbol{u}_{k,i}^{\mathrm{T}}\boldsymbol{w}_{C_p}^o \\ \boldsymbol{z}_{k,i} = \boldsymbol{u}_{k,i} + \boldsymbol{n}_{k,i} \\ y_k(i) = d_k(i) + v_k(i) \end{cases}, \tag{1}$$

where $\boldsymbol{w}_{C_p}^o$ is an $L \times 1$ unknown parameter vector. All the vectors are column vectors in this paper. The $d_k(i)$ is a scalar and $\boldsymbol{u}_{k,i}$ is an $L \times 1$ vector and they are the measurements of the output and input processes $d_k$ and $\boldsymbol{u}_k$ respectively. The $\boldsymbol{n}_{k,i}$ and $v_k(i)$ denote the input noise vector and output noise scalar respectively.

The FIR system is modeled under several assumptions[4]:

- Both input and output noises are stationary ergodic zero mean white with unknown variance $\sigma_{\boldsymbol{n}}^2$ and $\sigma_v^2$ respectively.
- The $\boldsymbol{u}_k$ and $\boldsymbol{d}_k$ are correlation-ergodic processes.
- $\boldsymbol{u}_{k,i}$, $\boldsymbol{n}_{k,i}$, $v_k(i)$ are independent with each other.

Considering the cluster $C_p$ of the multi-task network, the least-squares (LS) algorithm is considered as solving overdetermined equations under LS criterion. Consider an overdetermined equation

$$\boldsymbol{Z}_{C_p}^{\mathrm{T}}\boldsymbol{W}_{C_p} = \boldsymbol{Y}_{C_p}, \tag{2}$$

where $\boldsymbol{Z}_{C_p} \in \mathbb{R}^{m\times n}$ is an error-perturbed data matrix, $\boldsymbol{W}_{C_p} \in \mathbb{R}^{n\times d}$ is a deviation-perturbed parameter matrix and $\boldsymbol{Y}_{C_p} \in \mathbb{R}^{m\times d}$ is an error-perturbed measurement. The solution of this equation does not exist because $m > n$. Thus an approximate solution is needed to be found out. The LS method aims to compensate the perturbation in right hand side of the equation by adding a correction matrix $\Delta\boldsymbol{Y}_{C_p}$ to get an exact solution:

$$\begin{aligned} &\widehat{\boldsymbol{W}}_{C_p}^{LS} = arg \min_{\boldsymbol{W}_{C_p},\Delta\boldsymbol{Y}_{C_p}} \|\Delta\boldsymbol{Y}_{C_p}\|_2^2 \\ &subject\ to\ \boldsymbol{Z}\mathrm{T}_{C_p}\boldsymbol{W}_{C_p} = \boldsymbol{Y}_{C_p} + \Delta\boldsymbol{Y}_{C_p}. \end{aligned} \tag{3}$$

Apparently the smaller the $\|\Delta\boldsymbol{Y}_{C_p}\|_2^2$ is, the less perturbation exists in $\boldsymbol{Y}_{C_p}$.

The LS method only focuses on minimizing the residual $\|\Delta\boldsymbol{Y}_{C_p}\|_2^2 = \|\boldsymbol{Z}_{C_p}^{\mathrm{T}}\boldsymbol{W}_{C_p} - \boldsymbol{Y}_{C_p}\|_2^2$, thus it cannot obtain a consistent estimation in EIV problems. The TLS algorithm is motivated by balancing the asymmetry that $\boldsymbol{Y}_{C_p}$ is corrected while $\boldsymbol{Z}_{C_p}$ is not[17] in LS. When both $\boldsymbol{Z}_{C_p}$ and $\boldsymbol{Y}_{C_p}$ are perturbed, the TLS method aims to compensate the perturbation in both sides of the equation:

$$\begin{aligned} &\widehat{\boldsymbol{W}}_{C_p}^{TLS} = arg \min_{\boldsymbol{W}_{C_p},\Delta\boldsymbol{Z}_{C_p},\Delta\boldsymbol{Y}_{C_p}} \|[\Delta\boldsymbol{Z}_{C_p}, \Delta\boldsymbol{Y}_{C_p}]\|_2^2 \\ &subject\ to\ (\boldsymbol{Z}_{C_p} + \Delta\boldsymbol{Z}_{C_p})\mathrm{T}\boldsymbol{W}_{C_p} = \boldsymbol{Y}_{C_p} + \Delta\boldsymbol{Y}_{C_p} \end{aligned} \tag{4}$$

Figure 2 shows the differences of geometrical meaning between the LS and TLS methods. When the dimension of parameter matrix $\boldsymbol{W}_{C_p}$ is 1, which means $n = 1$, $d = 1$, the relationship between $\boldsymbol{Y}_{C_p}$ and $\boldsymbol{Z}_{C_p}$ can be shown in rectangular coordinate system, where the slope of straight line is the estimated parameter. The target of TLS method is to minimize $\sum \alpha_i$ while that of LS is minimizing $\sum \beta_i$. That is, the LS method tries to minimize the vertical distance but the TLS tries to minimize the perpendicular distance. And we can draw the conclusion that TLS-type algorithm can gain a consistent estimation.

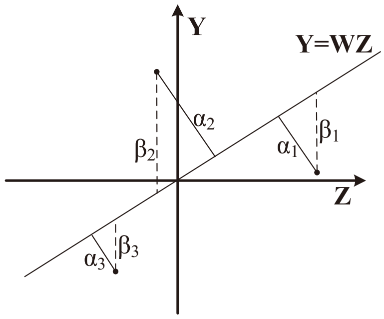


Fig. 2: Geometrical meaning of the LS and the TLS methods for $n = 1$, $d = 1$.

When the dimension of parameter matrix $\boldsymbol{W}_{C_p}$ is bigger than 1, finding a TLS solution is equal to seeking the closest approach between $\begin{bmatrix} \boldsymbol{z}_i \\ \boldsymbol{y}_i \end{bmatrix}$ and subspace $\boldsymbol{P}$, where $\boldsymbol{z}_i$ is the $i$-th row of matrix $\boldsymbol{Z}_{C_p}$ and $y_i$ is the $i$-th element of $\boldsymbol{Y}$ and $\boldsymbol{P} = \left\{ \begin{pmatrix} \boldsymbol{z} \\ y \end{pmatrix} | \boldsymbol{z} \in \mathbb{R}^{n\times 1}, y \in \mathbb{R}, y = \boldsymbol{z}^{\mathrm{T}}\boldsymbol{w} \right\}$.

## 3 Multi-task TLS algorithm

In many practical situations, the similarities and relationships do not exist among the parameters of neighboring nodes, which means that the cluster's objective parameters are often completely different from others. Thus in this paper, we study the condition that the information interaction does not exist in nodes which belongs to different clusters. Each cluster has its particular parameter vector. The nodes belong to same cluster collaborate with each other to estimate parameter while the nodes belong to different do not exchange informations with others. Considering a clustered multi-task network whose $N$ nodes are grouped into several clusters and using distributed FIR model, the M-EIV problem can be described as minimizing a global cost

$$\begin{aligned} &\min_{\boldsymbol{w}_k, \Delta\boldsymbol{Z}_k, \Delta\boldsymbol{y}_k} \sum_{k=1}^{N} \|[\Delta\boldsymbol{Z}_k, \Delta\boldsymbol{y}_k]\|_2^2 \\ &subject\ to\ (\boldsymbol{Z}_k + \Delta\boldsymbol{Z}_k)\boldsymbol{w} = \boldsymbol{y}_k + \Delta\boldsymbol{y}_k, \end{aligned} \tag{5}$$

where $\boldsymbol{Z}_k = [\boldsymbol{z}_{k,1}, \boldsymbol{z}_{k,1}, \cdots, \boldsymbol{z}_{k,i}]^{\mathrm{T}}$, $\Delta\boldsymbol{Z}_k = -[\boldsymbol{n}_{k,1}, \boldsymbol{n}_{k,1}, \cdots, \boldsymbol{n}_{k,i}]^{\mathrm{T}}$, $\boldsymbol{y}_k = [y_k(1), y_k(2), \cdots, y_k(i)]^{\mathrm{T}}$ and $\Delta\boldsymbol{y}_k = -[v_k(1), v_k(2), \cdots, v_k(i)]^{\mathrm{T}}$. Considering each cluster has different optimum parameters, the global cost degenerates into local cost within cluster $C(k)$:

$$\begin{aligned} &\min_{\boldsymbol{w}_k, \Delta\boldsymbol{Z}_k, \Delta\boldsymbol{y}_k} \sum_{k \in N_k \bigcap C(k)} \|[\Delta\boldsymbol{Z}_k, \Delta\boldsymbol{y}_k]\|_2^2 \\ &subject\ to\ (\boldsymbol{Z}_k + \Delta\boldsymbol{Z}_k)\boldsymbol{w} = \boldsymbol{y}_k + \Delta\boldsymbol{y}_k, \end{aligned} \tag{6}$$

For each node $k$, equation (6) can be converted into a local cost function

$$\begin{aligned} &\min_{\boldsymbol{w}_k, \Delta\boldsymbol{Z}_k, \Delta\boldsymbol{y}_k} \|[\Delta\boldsymbol{Z}_k, \Delta\boldsymbol{y}_k]\|_2^2 \\ &subject\ to\ (\boldsymbol{Z}_k + \Delta\boldsymbol{Z}_k)\boldsymbol{w} = \boldsymbol{y}_k + \Delta\boldsymbol{y}_k. \end{aligned} \tag{7}$$

Define the augmented data matrix $\boldsymbol{A}_k = [\boldsymbol{Z}_k, \boldsymbol{y}_k]$, $\Delta\boldsymbol{A}_k = [\Delta\boldsymbol{Z}_k, \Delta\boldsymbol{y}_k]$ and vector $\boldsymbol{b}_k = \begin{bmatrix} \boldsymbol{w} \\ -1 \end{bmatrix}$. The equation (7) can be described as

$$\begin{aligned} &\min_{\boldsymbol{w}_k, \Delta\boldsymbol{Z}_k, \Delta\boldsymbol{y}_k} \|\Delta\boldsymbol{A}_k\|_2^2 \\ &subject\ to\ (\boldsymbol{A}_k + \Delta\boldsymbol{A}_k)\boldsymbol{b}_k = 0. \end{aligned} \tag{8}$$

If $\|\boldsymbol{b}_k\|_2^2 = 1$ and rewrite equation (8) as

$$\boldsymbol{A}_k\boldsymbol{b}_k = \boldsymbol{c} = -\Delta\boldsymbol{A}_k\boldsymbol{b}_k, \tag{9}$$

the TLS problem can be converted into a constrained standard least-squares problem, which can be described as

$$\min \|\boldsymbol{A}_k\boldsymbol{b}_k\|_2^2 = \min \|\boldsymbol{c}\|_2^2 \tag{10}$$

$$subject\ to\ \boldsymbol{b}_k^{\mathrm{T}}\boldsymbol{b}_k = 1. \tag{11}$$

The vector $\boldsymbol{c}$ can be seen as the error vector of TLS solution to $\boldsymbol{A}_k\boldsymbol{b}_k = 0$. It is known that the solution of equation (10) and (11) is the right vector corresponding to the minimum singular value of the matrix $\boldsymbol{A}_k$[18]. If the singular value decomposition of $\boldsymbol{A}_{k,i}$ is

$$\boldsymbol{A}_k = U_k\Sigma_k V_k, \tag{12}$$

and the singular value of the matrix $\boldsymbol{A}_{k,i}$ is sorted as decreasing order and $\sigma_{k,L} > \sigma_{k,L+1}$, which means the smallest singular value is strictly smaller than the second smallest one, the total least square will be unique as

$$\hat{\boldsymbol{w}}_k^{TLS} = -\frac{v_{k,L+1}(1:L)}{v_{k,L+1}(L+1)}, \tag{13}$$

where $v_{k,i,L+1}$ is the $(L+1)$-th column of matrix $V_{k,i}$.

As the SVD consumes lots of computational resource[17], we introduce a proximal gradient descent method to obtain the singular vector for the TLS solution. Consider the problem

$$\min_{x} \quad f(\boldsymbol{x}_{k,i}) + g(\boldsymbol{x}_{k,i}), \tag{14}$$

where $f : \boldsymbol{R}^n \to \boldsymbol{R}$ is a differentiable function while $g : \boldsymbol{R}^n \to \boldsymbol{R}$ is no need to be differentiable. Both $f$ and $g$ are closed proper convex functions. The proximal gradient descent method can be implemented as

$$\boldsymbol{x}_i = \mathbf{prox}_{\mu g}(\boldsymbol{x}_{i-1} - \frac{\mu}{2}\nabla f(\boldsymbol{x}_{i-1})), \tag{15}$$

where the prox-operator is defined as

$$\mathbf{prox}_{\mu g}(\boldsymbol{u}) = \mathrm{argmin}_{\boldsymbol{x}} \left( g(\boldsymbol{x}) + \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{u}\|_2^2 \right). \tag{16}$$

Combining the proximal gradient descent method with the TLS problem described by equation (10) and (11), it is obvious that

$$f(\boldsymbol{b}_k) = \boldsymbol{b}_k^{\mathrm{T}}\boldsymbol{A}_k^{\mathrm{T}}\boldsymbol{A}_k\boldsymbol{b}_k \tag{17}$$

and $g(\cdot)$ is the indicator function

$$g(\boldsymbol{b}_k) = \begin{cases} 0 & \|\boldsymbol{b}_k\|_2^2 = 1 \\ +\infty & \|\boldsymbol{b}_k\|_2^2 \neq 1. \end{cases} \tag{18}$$

As the function $g$ is exactly the projection into the hyper-sphere, it is obvious that

$$\mathbf{prox}_{\mu g}(\boldsymbol{x}) = \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_2} \qquad \text{when } \boldsymbol{x} \neq 0. \tag{19}$$

The differential of function $f$ can be described as

$$\nabla f(\boldsymbol{b}_k) = 2\boldsymbol{A}_k^{\mathrm{T}}\boldsymbol{A}_k\boldsymbol{b}_k. \tag{20}$$

To make it suitable for on-line analysis, we use the instantaneous data $[\boldsymbol{z}_{k,i}^{\mathrm{T}}, y_k(i)]$ to replace $\boldsymbol{A}_k$, hence

$$\nabla f(\boldsymbol{b}_k) = 2\left[\boldsymbol{z}_{k,i}^{\mathrm{T}}, y_k(i)\right]^{\mathrm{T}}\left[\boldsymbol{z}_{k,i}^{\mathrm{T}}, y_k(i)\right]\boldsymbol{b}_k. \tag{21}$$

Substituting (21) and (19) into (15), we derive the updating function

$$\begin{cases} \boldsymbol{\varphi}_{k,i} = \boldsymbol{b}_{k-1,i} - \mu \begin{bmatrix} \boldsymbol{z}_{k,i} \\ y_k(i) \end{bmatrix} \left[\boldsymbol{z}_{k,i}^{\mathrm{T}}, y_k(i)\right]\boldsymbol{b}_{k-1,i} \\ \boldsymbol{b}_{k,i} = \dfrac{\boldsymbol{\varphi}_{k,i}}{\|\boldsymbol{\varphi}_{k,i}\|_2} \end{cases}. \tag{22}$$

and the estimate of the objective parameter

$$\hat{\boldsymbol{w}}_{k,i}^{TLS} = -\frac{\boldsymbol{b}_k(1:L)}{\boldsymbol{b}_k(L+1)}. \tag{23}$$

The M-TLS is designed in a diffusion strategy fashion. Compared with other distributed strategies, the diffusion strategy is more robust to link failure, which makes it more suitable for the complex application scenarios, such as multi-task networks. The diffusion includes two schemes: adapt-then-combine (ATC) and combine-then-adapt (CTA). It has been proved that the ATC usually outperforms CTA [10] and we construct our algorithm with ATC scheme.

---

**Multi-task TLS (M-TLS)**

**Initialization:**

$\boldsymbol{b}_{k,0} = \delta[1, 1, \cdots, 1]^{\mathrm{T}}$, $\delta$ is a small constant positive number.

**for** $i$ = 1:T

**for each node** $k$:

$\boldsymbol{\varphi}_{k,i} = \boldsymbol{b}_{k,i-1} - \mu \begin{bmatrix} \boldsymbol{z}_{k,i} \\ y_k(i) \end{bmatrix} \left[\boldsymbol{z}_{k,i}^{\mathrm{T}}, y_k(i)\right]\boldsymbol{b}_{k,i-1};$

$\boldsymbol{\psi}_{k,i} = \boldsymbol{\varphi}_{k,i}/\|\boldsymbol{\varphi}_{k,i}\|_2;$

$\boldsymbol{b}_{k,i} = \sum_{l \in N_k \bigcap C(k)} a_{lk}\boldsymbol{\psi}_{k,i};$

$\hat{\boldsymbol{w}}_{k,i}^{TLS} = -\frac{\boldsymbol{b}_{k,i}(1:L)}{\boldsymbol{b}_{k,i}(L+1)};$

**end**

---

## 4 Simulation results

In this section, we implement the proposed algorithms to obtain numerical simulation results. We compare the proposed algorithm with the TLS with adaptive cooperation (AC-dTLS), TLS with fixed intertask combiners (FC-dTLS) and independent TLS (IND-dTLS)[16] to evaluate its performance and verify its effectiveness.

The performances are evaluated by taking the measurements of the transient network mean square deviation (MSD). The MSD is defined as:

$$MSD = \frac{1}{N}\sum_{k=1}^{N}\|\boldsymbol{w}_{C(k)}^{o} - \hat{\boldsymbol{w}}_k\|_2. \tag{24}$$

Consider a multi-agent network that contains 18 agents which is divided into 6 clusters: $C_1 = 1,2,3, C_2 = 4,5,6, C_3 = 7,8,9, C_4 = 10,11,12, C_5 = 13,14,15, C_6 = 16,17,18$. The FIR filter order is set as 5 taps. To evaluate the performance of our proposed algorithm under different conditions, the simulations are under three cases, where the differences between the optimum vector of each clusters are large, medium and small. For the first case, the optimum parameters are set as:

$$\boldsymbol{w}_{C_p}^{o} = \begin{cases} [0.5, -0.4, 0.3, -0.2, 0.1]^{\mathrm{T}}, & p = 1 \\ [0.4, -0.2, 0.25, -0.1, 0.1]^{\mathrm{T}}, & p = 2 \\ [0.3, -0.3, -0.15, 0.15, -0.2]^{\mathrm{T}}, & p = 3 \\ [-0.2, 0.6, 0.4, 0.3, -0.5]^{\mathrm{T}}, & p = 4 \\ [-0.8, -0.1, 0.1, 0.2, -0.3]^{\mathrm{T}}, & p = 5 \\ [-0.6, 0.6, 0.4, 0.3, -0.5]^{\mathrm{T}}, & p = 6 \end{cases}.$$

For the other two cases, the optimum parameter vectors of each cluster are set as $\boldsymbol{w}_{C_p}^{o} = \boldsymbol{w}^{o} + \Delta\boldsymbol{w}_{C_p}^{o}$. The $\boldsymbol{w}^{o}$ is set as $[-0.2; 0.3; -0.4; 0.5; -0.7]$. The $\Delta\boldsymbol{w}_{C_p}^{o}$ is set randomly and varies within $[-0.01\ 0.01]$ for the second case. For the third case, it varies in $[-0.06\ 0.1]$. The iteration number is set as 1500. For the M-TLS algorithms, the step size is chosen as $\mu = 0.01$. All the results are averaged over 100 independent trials. The input data is white process whose variance is set as 1. The additive noises are also white processes whose variances are set randomly at each node and vary in $[0.02\ 0.12]$.

The transient noise variance estimation mechanism performance of the algorithms is shown in Figure 3.

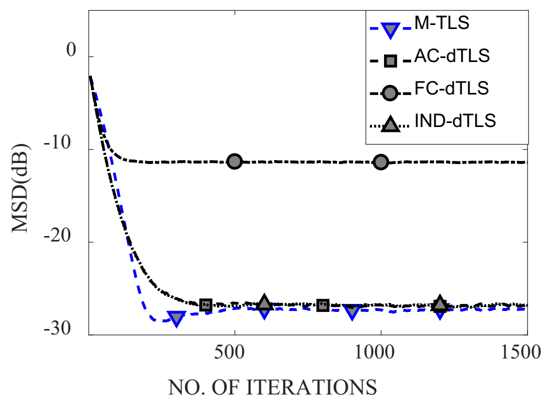


Fig. 3: Transient network MSD of the TLS-type algorithms for case 1.

Fromk Figure 3, we can drop the conclusion that our proposed algorithm has the best transient MSD performance when the optimum parameter vectors of each cluster are completely different.

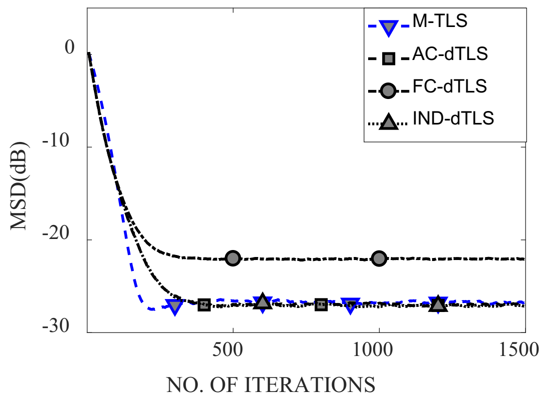


Fig. 4: Transient network MSD of the TLS-type algorithms for case 2.

For case 2, where the differences between each optimum parameter vectors are medium, the performance of M-TLS is also as good as the AC-dTLS and the IND-dTLS.

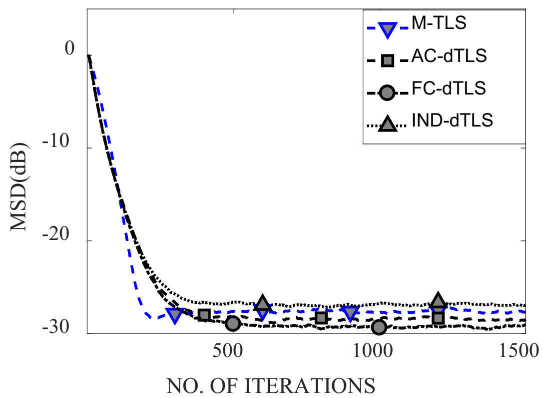


Fig. 5: Transient network MSD of the TLS-type algorithms for case 3.

For case 3, we can drop the conclusion that all of the TLS-type algorithms has nearly the same performances.

## 5 Conclusion

In this work, we study the problem of collaborative unbiased estimation for M-EIV model over multi-agent network. We propose an M-LTS algorithm that has an excellent performance under the multi-task conditions. Our proposed method minimizes the perturbation in both input and output at each agent while the latter algorithm calculates and compensates the bias. For the M-TLS, we use the proximal gradient method to fulfill the calculation instead of SVD, which reduces the computational cost greatly. The proposed algorithms are implemented and tested under the diffusion distributed strategy of network cooperation for the reason that the diffusion strategy is more robust to link failure though it costs more communication resource. Simulation results show that the M-TLS attains the good convergence rates and its MSD performance is no worse than the other TLS-type algorithms. In this research, we assume that the nodes between different clusters do not have information interactions. In future work, we will focus on the condition that the nodes diffused in different clusters can communicate with each other and maintain the estimation collaboratively.

## References


[1] Ali H. Sayed, Tu S Y, Chen J, et al, Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior, *IEEE Signal Processing Magazine*, 30(3):155-171, 2013.

[2] Towfic Z J, Chen J, Ali H. Sayed, Collaborative learning of mixture models using diffusion adaptation, *IEEE International Workshop on Machine Learning for Signal Processing*, 2011:1-6.

[3] Chen J, Ali H. Sayed, Diffusion Adaptation Strategies for Distributed Optimization and Learning Over Networks, *IEEE Transactions on Signal Processing*, 60(8):4289-4305,2012.

[4] Lou J, Jia L, Tao R, et al, Distributed incremental bias-compensated RLS estimation over multi-agent networks, *SCIENCE CHINA: Information Sciences*, 60(3):032204, 2017.

[5] Tu S Y, Ali H. Sayed, Mobile Adaptive Networks, *IEEE Journal of Selected Topics in Signal Processing*, 5(4): 649-664, 2011.

[6] Estrin D, Girod L, Pottie G, et al, Instrumenting the world with wireless sensor networks, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2001: 2033-2036 vol.4.

[7] Arablouei R, Werner S, Dogancay K. Diffusion-based distributed adaptive estimation utilizing gradient-descent total least-squares. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013: 5308-5312.

[8] Lopes C G, Ali H. Sayed Diffusion Least-Mean Squares Over Adaptive Networks, *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007: 3122-3136.

[9] Cattivelli F S, Lopes C G, Ali H. Sayed, Diffusion recursive least-squares for distributed estimation over adaptive networks, *IEEE Transactions on Signal Processing*, 56(5): 1865-1877, 2008.

[10] Huang W, Yang X, Shen G, Communication-reducing diffusion LMS algorithm over multitask networks, *Information Sciences*, s 382-383: 115-134, 2017.

[11] Jie Chen, Cdric Richard, Ali H. Sayed, Diffusion LMS Over Multitask Networks, *IEEE Transactions on Signal Processing*, 63(11): 2733-2748, 2015.

[12] Cao X, Liu K J R. Decentralized Sparse Multitask RLS Over Networks. *IEEE Transactions on Signal Processing*, 65(23): 6217-6232, 2017.

[13] Chen J, Richard C, Ali H. Sayed, Diffusion LMS for clustered multitask networks, *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014: 5487-5491.

[14] Hajiabadi M, Hodtani G A, Khoshbin H, Adaptive multitask network based on maximum correntropy learning algorithm, *International Journal of Adaptive Control and Signal Processing*, 31.8(2017).

[15] Chen J, Richard C, Ali H. Sayed, Multitask Diffusion Adaptation Over Networks, *IEEE Journal of Selected Topics in Signal Processing*, PP(99): 1-1, 2014.

[16] Li C, Huang S, Liu Y, et al. Distributed TLS over multitask networks with adaptive intertask cooperation. *IEEE Transactions on Aerospace and Electronic Systems*, 52(6): 3036-3052, 2017.

[17] Markovsky I, Huffel S V, Overview of total least-squares methods, *Signal Processing*, 87(10): 2283-2302, 2007.

[18] Golub G H, Loan C F V, An Analysis of the Total Least Squares Problem, *Siam Journal on Numerical Analysis*, 17(6): 883-893, 1980.