

# CONSENSUS-BASED DISTRIBUTED TOTAL LEAST-SQUARES ESTIMATION USING PARAMETRIC SEMIDEFINITE PROGRAMMING

*Cristiano Gratton<sup>\*</sup>, Naveen K. D. Venkategowda<sup>\*</sup>, Reza Arablouei<sup>†</sup>, Stefan Werner<sup>\*</sup>*

<sup>\*</sup> Department of Electronic Systems, NTNU - Norwegian University of Science and Technology

<sup>†</sup> CSIRO's Data61, Pullenvale QLD 4069, Australia

## ABSTRACT

We propose a new distributed algorithm to solve the total least-squares (TLS) problem when data are distributed over a multi-agent network. To develop the proposed algorithm, named distributed ADMM TLS (DA-TLS), we reformulate the TLS problem as a parametric semidefinite program and solve it using the alternating direction method of multipliers (ADMM). Unlike the existing consensus-based approaches to distributed TLS estimation, DA-TLS does not require careful tuning of any design parameter. Numerical experiments demonstrate that the DA-TLS converges to the centralized solution significantly faster than the existing consensus-based TLS algorithms.

**Index Terms**— ADMM, consensus, distributed estimation, total least squares, semidefinite programming.

## 1. INTRODUCTION

With the recent advances in technology, large quantities of data are collected by numerous sensors, which are often geographically dispersed. Hence, performing data analysis tasks such as estimation and classification at a central processing unit is impractical due to transmission cost or privacy reasons. Furthermore, collecting all the data in a fusion center creates a single point of failure. Therefore, it is imperative to develop algorithms that are capable of processing data spread across multiple agents [1–7].

In the realm of linear estimation, the total least-squares (TLS) method has been introduced as an alternative to the ordinary least-squares method to deal with errors-in-variables models. In such models, both independent and dependent variables are corrupted by noise or perturbation. TLS has been successfully used in several signal processing applications, e.g., frequency estimation of power systems [8–10], cognitive spectrum sensing [11], system identification [12], and wireless sensor networks [13].

The distributed TLS problem has previously been considered in [13–21]. The works in [13, 14] are based on the consensus strategy and rely on the dual-based subgradient

method. Their relatively high computational complexity has partially motivated the works in [16–20]. While the approach of [16] is based on the average consensus strategy, the algorithms in [17–21] are based on diffusion strategies and, therefore, suffer from relatively slow convergence [6]. The convergence speed of the algorithm proposed in [13] greatly depends on the network topology and dimensionality of the data. Although these shortcomings are mitigated in [14], the convergence rate of algorithms in [13] and [14] highly depends on the choice of the step-size, whose optimal tuning requires the global knowledge of the data and network topology.

In this paper, we solve the distributed TLS problem when each agent has access to parts of a set of linear equations, i.e., a subset of the rows of the observation matrix and the output vector. This is a common scenario in wireless sensor networks, e.g., distributed system identification [22]. Through a change of variable from a vector to a rank-one matrix and subsequent semidefinite relaxation (SDR), we transform the non-convex distributed TLS problem into a semidefinite program. We solve the modified problem using the alternating direction method of multipliers (ADMM) and a generalization of the algorithm proposed in [23] for fractional programming. Since the optimal solution is rank-one, the relaxation is tight and does not incur any loss of optimality [24]. In addition, as the objective function in the modified problem is the sum of fractions of linear functions, the convergence of the proposed algorithm to the globally optimal solution is guaranteed.

The proposed algorithm, called distributed ADMM TLS (DA-TLS), is fully distributed in the sense that it requires the agents to share data only with their immediate neighbors at each iteration. Furthermore, the performance of DA-TLS is not sensitive to the tuning of its parameters. This makes DA-TLS more flexible and suitable for distributed deployment in comparison with the algorithms of [13, 14]. Simulation results show faster convergence of DA-TLS to the centralized solution at all agents in comparison with the existing algorithms.

## 2. SYSTEM MODEL

We consider a connected network of  $K \in \mathbb{N}$  agents modeled as an undirected graph  $\mathcal{G}(\mathcal{K}, \mathcal{E})$  where the set of vertices  $\mathcal{K} = \{1, \dots, K\}$  corresponds to the agents and the edge set

---

This work was partly supported by the Research Council of Norway.

$\mathcal{E}$  represents the communication links between the pairs of agents. Agent  $k \in \mathcal{K}$  can communicate with its neighbors whose indexes are in the set  $\mathcal{N}_k$  with cardinality  $|\mathcal{N}_k|$ . By convention,  $\mathcal{N}_k$  does not include the agent  $k$  itself.

Let  $\mathbf{X} \in \mathbb{R}^{N \times P}$  be the observation matrix,  $\mathbf{\Delta} \in \mathbb{R}^{N \times P}$  the error in the observation matrix,  $\mathbf{y} \in \mathbb{R}^{N \times 1}$  the response vector,  $\mathbf{\delta} \in \mathbb{R}^{N \times 1}$  the error in the response vector, and  $\mathbf{w} \in \mathbb{R}^{P \times 1}$  the sought-after parameter vector that relates  $\mathbf{X}$  and  $\mathbf{y}$  through  $(\mathbf{X} - \mathbf{\Delta})\mathbf{w} = \mathbf{y} - \mathbf{\delta}$ . The matrix  $\mathbf{X}$  consists of  $K$  submatrices  $\mathbf{X}_k$ , i.e.,  $\mathbf{X} = [\mathbf{X}_1^T, \mathbf{X}_2^T, \dots, \mathbf{X}_K^T]^T$ , and the vector  $\mathbf{y} \in \mathbb{R}^{N \times 1}$  of  $K$  subvectors  $\mathbf{y}_k$ , i.e.,  $\mathbf{y} = [\mathbf{y}_1^T, \mathbf{y}_2^T, \dots, \mathbf{y}_K^T]^T$ , as the data are distributed among the agents and each agent  $k$  holds its respective  $\mathbf{X}_k \in \mathbb{R}^{N_k \times P}$  and  $\mathbf{y}_k \in \mathbb{R}^{N_k \times 1}$  where  $\sum_{k=1}^K N_k = N$  and  $(\cdot)^T$  denotes the matrix transpose.

The TLS estimate of the unknown parameter vector  $\mathbf{w}$  can be found by solving the constrained optimization problem

$$\begin{aligned} \min_{\mathbf{w}, \mathbf{\Delta}, \mathbf{\delta}} \quad & \|\mathbf{\Delta}\|_F + \|\mathbf{\delta}\| \\ \text{s.t.} \quad & (\mathbf{X} - \mathbf{\Delta})\mathbf{w} = \mathbf{y} - \mathbf{\delta} \end{aligned} \quad (1)$$

where  $\|\cdot\|_F$  and  $\|\cdot\|$  denote the Frobenius norm and Euclidean norm, respectively. When the entries of  $\mathbf{\Delta}$  and  $\mathbf{\delta}$  are independent and identically distributed (i.i.d.), a centralized TLS solution  $\mathbf{w}^c$  of (1) can be obtained as

$$\mathbf{w}^c = \frac{-1}{v_{P+1}} [v_1, v_2, \dots, v_P]^T \quad (2)$$

where  $\mathbf{v} = [v_1, v_2, \dots, v_{P+1}]^T$  is the right singular vector corresponding to the smallest singular value of  $[\mathbf{X}, \mathbf{y}]$  [25].

An equivalent but more practical solution can be obtained by minimizing the Rayleigh quotient cost function as [25]

$$\min_{\mathbf{w}} \frac{\|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2}{\|\mathbf{w}\|^2 + 1} \quad \text{or} \quad \min_{\mathbf{w}} \sum_{k=1}^K \frac{\|\mathbf{X}_k \mathbf{w} - \mathbf{y}_k\|^2}{\|\mathbf{w}\|^2 + 1}. \quad (3)$$

Since finding a centralized solution of (3) over a network may be inefficient, we propose a distributed algorithm for this purpose in the following section.

### 3. DISTRIBUTED TLS

We first discuss the SDR technique that allows us to transform the TLS problem into a parametric semidefinite program, which we solve iteratively through two nested loops. Then, we describe the consensus-based reformulation of the resultant parametric semidefinite program that enables its distributed solution via the ADMM, which forms the inner loop. Finally, we describe the steps of the inner and outer loops of the algorithm.

#### 3.1. Semidefinite Relaxation

Using the properties of the matrix trace operator, we rewrite the Rayleigh quotient cost function in (3) as

$$\sum_{k=1}^K \frac{\text{tr}(\mathbf{w}\mathbf{w}^T \mathbf{X}_k^T \mathbf{X}_k) - 2\mathbf{y}_k^T \mathbf{X}_k \mathbf{w} + \|\mathbf{y}_k\|^2}{\text{tr}(\mathbf{w}\mathbf{w}^T) + 1}. \quad (4)$$

Considering (4) and defining

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}\mathbf{w}^T & \mathbf{w} \\ \mathbf{w}^T & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{C}_k = \begin{bmatrix} \mathbf{X}_k^T \mathbf{X}_k & -\mathbf{X}_k^T \mathbf{y}_k \\ -\mathbf{y}_k^T \mathbf{X}_k & \|\mathbf{y}_k\|^2 \end{bmatrix}, \quad (5)$$

(3) can be recast as

$$\begin{aligned} \min_{\mathbf{W} \succeq \mathbf{0}} \quad & \sum_{k=1}^K \frac{\text{tr}(\mathbf{C}_k \mathbf{W})}{\text{tr}(\mathbf{W})} \\ \text{s.t.} \quad & \text{rank}(\mathbf{W}) = 1. \end{aligned} \quad (6)$$

Relaxing the rank constraint in (6) turns it into the following aggregate linear-fractional program

$$\min_{\mathbf{W} \succeq \mathbf{0}} \sum_{k=1}^K \frac{\text{tr}(\mathbf{C}_k \mathbf{W})}{\text{tr}(\mathbf{W})}. \quad (7)$$

Both numerator and denominator of the summands in the objective function of (7) are linear functions of the matrix variable  $\mathbf{W}$ . Therefore, (7) can be converted to a parametric semidefinite program whose objective is in the subtractive form as per the following proposition.

**Proposition 1.** *Let  $\mathbf{W}^*$  denote the optimal solution to (7). Then, there exists a vector  $\beta^* = [\beta_1^*, \dots, \beta_K^*]$  such that  $\mathbf{W}^*$  is also the optimal solution of the following semidefinite program*

$$\mathbf{W}^* = \arg \min_{\mathbf{W} \succeq \mathbf{0}} \sum_{k=1}^K \text{tr}(\mathbf{C}_k \mathbf{W}) - \beta_k^* \text{tr}(\mathbf{W}). \quad (8)$$

*In addition,  $\mathbf{W}^*$  also satisfies the following system of equations:*

$$\text{tr}(\mathbf{C}_k \mathbf{W}^*) - \beta_k^* \text{tr}(\mathbf{W}^*) = 0, \quad k = 1, 2, \dots, K. \quad (9)$$

*Proof.* The Karush-Kuhn-Tucker (KKT) conditions of optimality [26] for problem (8) give the same solution set as the KKT conditions for the epigraph form of (7). Since the KKT conditions for both problems are sufficient for optimality, the two problems are equivalent. The system of equation (9) is due to the KKT conditions.  $\square$

In the next subsection, we describe a consensus-based reformulation of (8), which allows the application of the ADMM to solve (8) for any given  $\beta^*$ .

### 3.2. Building Consensus

In order to tackle (8) in a distributed fashion, we introduce  $\mathcal{W} := \{\mathbf{W}_k\}_{k=1}^K$  representing the local copies of  $\mathbf{W}$  at the agents. Therefore, we rewrite (8) in the following equivalent form

$$\begin{aligned} \min_{\{\mathbf{W}_k \succeq \mathbf{0}\}} & \sum_{k=1}^K \text{tr}(\mathbf{C}_k \mathbf{W}_k) - \beta_k^* \text{tr}(\mathbf{W}_k) \\ \text{s.t.} & \mathbf{W}_k = \mathbf{W}_l, \quad l \in \mathcal{N}_k, \quad k \in \mathcal{K}. \end{aligned} \quad (10)$$

The equality constraints enforce consensus over  $\mathbf{W}_k$ ,  $k = 1, \dots, K$ , across each agent's neighborhood  $\mathcal{N}_k$ .

To solve (10) in a distributed fashion, we employ the ADMM [1]. Hence, we introduce the auxiliary local variables  $\mathcal{Z} := \{\mathbf{Z}_k^l\}_{l \in \mathcal{N}_k}$  and rewrite (10) as

$$\begin{aligned} \min_{\{\mathbf{W}_k \succeq \mathbf{0}\}} & \sum_{k=1}^K \text{tr}(\mathbf{C}_k \mathbf{W}_k) - \beta_k^* \text{tr}(\mathbf{W}_k) \\ \text{s.t.} & \mathbf{W}_k = \mathbf{Z}_k^l, \mathbf{W}_l = \mathbf{Z}_l^k, \quad l \in \mathcal{N}_k, \quad k \in \mathcal{K}. \end{aligned} \quad (11)$$

Using the auxiliary variables  $\mathcal{Z}$ , we obtain an equivalent alternative representation of the constraints in (10). These variables are only used to derive the local recursions and are eventually eliminated. By associating the Lagrange multipliers  $\mathcal{V} := \{\{\mathbf{\Gamma}_k^l\}_{l \in \mathcal{N}_k}, \{\mathbf{\Lambda}_k^l\}_{l \in \mathcal{N}_k}\}_{k=1}^K$  with the constraints in (11), we get the following augmented Lagrangian function:

$$\begin{aligned} \mathcal{L}_\rho(\mathcal{W}, \mathcal{Z}, \mathcal{V}) &= \sum_{k=1}^K \text{tr}(\mathbf{C}_k \mathbf{W}_k) - \beta_k^* \text{tr}(\mathbf{W}_k) \\ &+ \sum_{k=1}^K \sum_{l \in \mathcal{N}_k} \text{tr} \left( (\mathbf{\Lambda}_k^l)^T (\mathbf{W}_k - \mathbf{Z}_k^l) + (\mathbf{\Gamma}_k^l)^T (\mathbf{W}_l - \mathbf{Z}_l^k) \right) \\ &+ \frac{\rho}{2} \sum_{k=1}^K \sum_{l \in \mathcal{N}_k} \left( \|\mathbf{W}_k - \mathbf{Z}_k^l\|_F^2 + \|\mathbf{W}_l - \mathbf{Z}_l^k\|_F^2 \right), \end{aligned} \quad (12)$$

where the constant  $\rho > 0$  is a penalty parameter.

Obtaining the solution through the ADMM entails an iterative process consisting of the following steps at each iteration: 1)  $\mathcal{L}_\rho$  is minimized with respect to  $\mathcal{W}$ ; 2)  $\mathcal{L}_\rho$  is minimized with respect to  $\mathcal{Z}$ ; and, 3) the Lagrange multipliers  $\mathcal{V}$  are updated through gradient-ascent [1].

Thanks to the reformulation of (8) as (11), the Lagrangian function (12) can be decoupled with respect to variables in  $\mathcal{W}$  and  $\mathcal{Z}$  as well as across the network agents  $\mathcal{K}$ . It can be shown that, in the ADMM steps, the auxiliary variables  $\mathcal{Z}$  and the Lagrange multipliers  $\{\mathbf{\Gamma}_k^l\}_{l \in \mathcal{N}_k}$  are eliminated. Hence, we end up with the following iterative updates at the  $k$ th agent

$$\mathbf{W}_k(m+1) = \arg \min_{\mathbf{W}_k \succeq \mathbf{0}} \mathcal{L}_\rho(\mathbf{W}_k, \mathbf{\Lambda}_k(m)) \quad (13)$$

$$\mathbf{\Lambda}_k(m+1) = \mathbf{\Lambda}_k(m) + \rho \sum_{l \in \mathcal{N}_k} [\mathbf{W}_k(m+1) - \mathbf{W}_l(m+1)], \quad (14)$$

where  $\mathbf{\Lambda}_k(m) = 2 \sum_{l \in \mathcal{N}_k} \mathbf{\Lambda}_k^l(m)$  and  $m$  is the iteration index.

The constrained minimization problem in (13) can be expressed as the following semidefinite least-squares problem

$$\min_{\mathbf{W}_k \succeq \mathbf{0}} \text{tr}[\mathbf{W}_k^T (\mathbf{W}_k - 2\mathbf{G}_k(m))], \quad (15)$$

where

$$\begin{aligned} \mathbf{G}_k(m) &= \frac{1}{2\rho|\mathcal{N}_k|} \left( \rho|\mathcal{N}_k| \mathbf{W}_k(m) + \rho \sum_{l \in \mathcal{N}_k} \mathbf{W}_l(m) \right. \\ &\quad \left. - \mathbf{C}_k + \beta_k \mathbf{I} - \mathbf{\Lambda}_k(m) \right). \end{aligned} \quad (16)$$

The solution of (15) is given by

$$\mathbf{W}_k(m+1) = \mathbf{U}(m) \max(\mathbf{\Sigma}(m), \mathbf{0}) \mathbf{U}(m)^T, \quad (17)$$

where  $\mathbf{U}(m)$  and  $\mathbf{\Sigma}(m)$  are the orthogonal and diagonal matrices coming from the eigen-decomposition (EVD)  $\mathbf{G}_k(m) = \mathbf{U}(m)\mathbf{\Sigma}(m)\mathbf{U}^T(m)$  and  $\max(\mathbf{\Sigma}(m), \mathbf{0})$  denotes the diagonal matrix whose entries are the maxima of the diagonal entries of  $\mathbf{\Sigma}(m)$ , i.e., the eigenvalues of  $\mathbf{G}_k(m)$ , and zero. Note that the most computationally intensive operation is the EVD.

### 3.3. Algorithm

The DA-TLS algorithm consists of two loops. In the inner loop, the solution of (8) is obtained using the ADMM for a given  $\beta^*$ . In the outer loop, we use a single iteration of the Newton's method [27] to find the solution of (9), i.e.,  $\beta_k(j+1) = \beta_k(j) - [\text{tr}(\mathbf{W})]^{-1} [\beta_k(j) \text{tr}(\mathbf{W}) - \text{tr}(\mathbf{C}_k \mathbf{W})]$ . The proposed algorithm is summarized in Algorithm 1.

---

#### Algorithm 1 DA-TLS

---

All agents  $k \in \mathcal{K}$  initialize  $\beta_k(1) = \text{tr}(\mathbf{C}_k)/(p+1)$  and locally run

**for**  $j = 1, 2, \dots$  **do**

Initialize  $\mathbf{W}_k(0) = \mathbf{0}$  and  $\mathbf{\Lambda}_k(0) = \mathbf{0}$

**for**  $m = 1, 2, \dots$  **do**

Receive  $\mathbf{W}_k(m)$  from neighbors in  $\mathcal{N}_k$

Update  $\mathbf{\Lambda}_k(m+1)$  as in (14)

Compute  $\mathbf{G}_k(m)$  as in (16)

Compute EVD of  $\mathbf{G}_k(m) = \mathbf{U}(m)\mathbf{\Sigma}(m)\mathbf{U}^T(m)$

Update  $\mathbf{W}_k(m+1) = \mathbf{U}(m) \max(\mathbf{\Sigma}(m), \mathbf{0}) \mathbf{U}^T(m)$

**end for**

Update  $\beta_k(j+1) = \frac{\text{tr}(\mathbf{C}_k \mathbf{W}_k(m+1))}{\text{tr}(\mathbf{W}_k(m+1))}$

**end for**

---

After estimating  $\mathbf{W}_k$ , the vector estimate  $\mathbf{w}_k$  is found as follows. Let  $\check{\mathbf{W}}_k = \mathbf{W}_k/\omega_k$  where  $\omega_k$  is the  $(P+1), (P+1)$  entry of  $\mathbf{W}_k$ . Then,  $\mathbf{w}_k$  is the eigenvector corresponding to the smallest eigenvalue of the  $P \times P$  upper-left submatrix of  $\check{\mathbf{W}}_k$ .

Using the results in [24, 28], it can be observed that the solution of (7) and consequently (8) is rank-one. Hence, optimizing with respect to the matrix variable  $\mathbf{W}$  and relaxing the rank constraint do not lead to any loss of optimality [24]. Therefore, the solutions to (3) and (10) coincide.

Convergence of the proposed DA-TLS algorithm to the global centralized solution can be proven by checking that both inner and outer loops converge. The convergence of the inner loop can be verified following [29, Proposition 3], i.e., for all  $k \in \mathcal{K}$ , the iterates  $\{\mathbf{W}_k(m)\}$ ,  $\{\mathbf{\Lambda}_k(m)\}$  produced by (13) and (14) are convergent and  $\mathbf{W}_k(m) \rightarrow \mathbf{W}^*$  as  $m \rightarrow \infty$ . Moreover, the convergence of the outer loop follows setting  $\bar{\mathbf{C}} = \sum_{k=1}^K \mathbf{C}_k$  and  $\bar{\beta}^* = \sum_{k=1}^K \beta_k^*$  and observing that the optimization in (8) is equivalent to

$$\min_{\mathbf{W} \succeq 0} \text{tr}(\bar{\mathbf{C}}\mathbf{W}) - \bar{\beta}^* \text{tr}(\mathbf{W}). \quad (18)$$

Since the objective function in (18) is linear, (18) is a standard semidefinite program with a unique solution. Therefore, DA-TLS naturally inherits the theoretical properties of the algorithm proposed in [23] for fractional programming whose convergence is guaranteed.

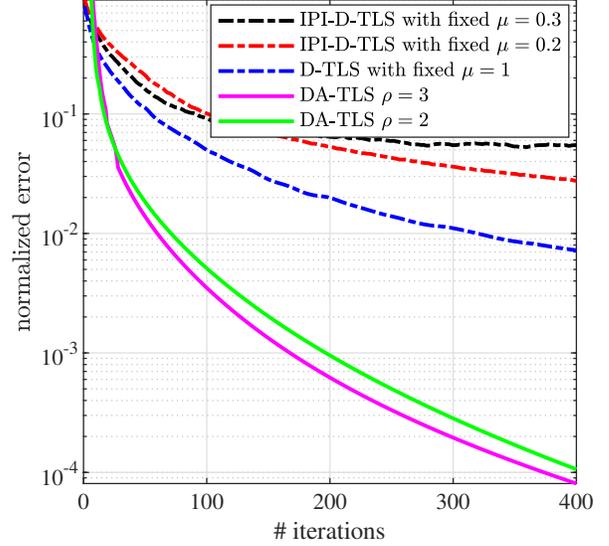
#### 4. SIMULATIONS

The simulated network is connected with a random topology and consists of  $K = 20$  agents where each agent is linked to three other agents on average. We average results over 100 independent trials. In each trial, the scenario is generated according to the same procedure as described in the simulation sections of [13, 14]. For each agent  $k \in \mathcal{K}$ , we create a  $2P \times P$  local observation matrix  $\mathbf{X}_k$  whose entries are drawn from a standard normal distribution. The entries of the parameter vector  $\mathbf{w}$  are also drawn from a standard normal distribution. The entries of the error matrix  $\mathbf{\Delta}$  and error vector  $\delta$  are i.i.d. zero-mean Gaussian with variance 0.25. To evaluate the performance of the proposed algorithm, we use the normalized error between the centralized TLS solution  $\mathbf{w}^c$  as per (2) and the local estimates that is defined as  $\sum_{k=1}^K \|\mathbf{w}_k - \mathbf{w}^c\|^2 / \|\mathbf{w}^c\|^2$  where  $\mathbf{w}_k$  denotes the local estimate at agent  $k$ . In Figs. 1-2, we plot the normalized error versus the total number of iterations, which is given by the product between the number of iterations of the inner and the outer loop. The former is set to 80 for Fig. 1 and 40 for Fig. 2, while the latter is set to 5 for both the plots.

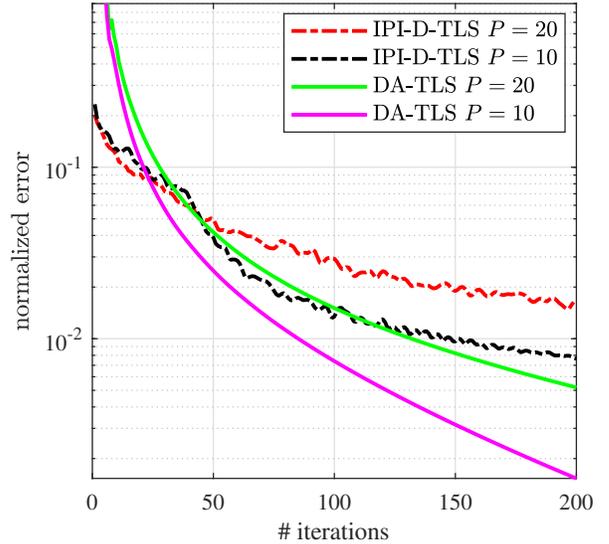
Fig. 1 shows that, for  $P = 9$ , DA-TLS with  $\rho = 2$  and  $\rho = 3$  converges significantly faster than the existing approaches, i.e., the distributed TLS (D-TLS) algorithm of [13] and the inverse-power-iteration-based distributed TLS (IPI-D-TLS) algorithm of [14]. Fig. 2 shows the superiority of DA-TLS with  $\rho = 1$  over IPI-D-TLS with  $\mu = 1$  for two different values of  $P$ . Although not further substantiated here due to the space constraints, we have observed that DA-TLS consistently outperforms its contenders in various scenarios.

#### 5. CONCLUSION

In this paper, we developed a new distributed algorithm for solving the TLS problem. We recast the original optimization problem into an equivalent linear-fractional program. Then,



**Fig. 1.** Normalized error of the DA-TLS, D-TLS, and IPI-D-TLS algorithms with two values of penalty parameter ( $\rho = 2$  and  $\rho = 3$ ) for DA-TLS and two values of the step-size ( $\mu = 0.2$  and  $\mu = 0.3$ ) for IPI-D-TLS.



**Fig. 2.** Normalized error for different values of  $P$ . For DA-TLS, we set  $\rho = 1$  and, for IPI-D-TLS, we set  $\mu = 1$ .

employing semidefinite relaxation, we transformed the resultant problem into a parametric semidefinite program whose structure is suitable for distributed treatment via ADMM. Simulation results showed that the proposed algorithm converges faster than the existing alternative algorithms while being less sensitive to tuning of the parameters involved in the algorithm.

## 6. REFERENCES

- [1] G. B. Giannakis, Q. Ling, G. Mateos, and I. D. Schizas, *Splitting Methods in Communication, Imaging, Science, and Engineering*, ser. Scientific Computation, R. Glowinski, S. J. Osher, and W. Yin, Eds. Cham: Springer International Publishing, 2016.
- [2] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5262–5276, Oct 2010.
- [3] S. Kumar, R. Jain, and K. Rajawat, "Asynchronous optimization over heterogeneous networks via consensus admm," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 1, pp. 114–129, Mar 2017.
- [4] J. Akhtar and K. Rajawat, "Distributed sequential estimation in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 1, pp. 86–100, Jan 2018.
- [5] N. K. D. Venkatesowda and S. Werner, "Privacy-preserving distributed precoder design for decentralized estimation," in *Proc. IEEE Global Conference on Signal and Information Processing*, Nov 2018.
- [6] C. Gratton, N. K. D. Venkatesowda, R. Arablouei, and S. Werner, "Distributed ridge regression with feature partitioning," in *Proc. Asilomar Conf. Signals Syst. Comput.*, Oct 2018.
- [7] S. P. Talebi, S. Werner, and D. P. Mandic, "Distributed adaptive filtering of  $\alpha$ -stable signals," *IEEE Signal Processing Letters*, vol. 25, no. 10, pp. 1450–1454, Oct 2018.
- [8] R. Arablouei, K. Doganay, and S. Werner, "Recursive total least-squares estimation of frequency in three-phase power systems," in *Proc. 22nd European Signal Processing Conference*, Sept 2014, pp. 2330–2334.
- [9] R. Arablouei, K. Doğançay, and S. Werner, "Adaptive frequency estimation of three-phase power systems," *Signal Processing*, vol. 109, pp. 290–300, Apr 2015.
- [10] R. Arablouei, S. Werner, and K. Doğançay, "Estimating frequency of three-phase power systems via widely-linear modeling and total least-squares," in *Proc. 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, Dec 2013, pp. 464–467.
- [11] E. Dall'Anese and G. B. Giannakis, "Distributed cognitive spectrum sensing via group sparse total least-squares," in *Proc. 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, Dec 2011, pp. 341–344.
- [12] I. Markovsky, J. Willems, S. Van Huffel, Bart De Moor, and R. Pintelon, "Application of structured total least squares for system identification and model reduction," *IEEE Transactions on Automatic Control*, vol. 50, no. 10, pp. 1490–1500, Oct 2005.
- [13] A. Bertrand and M. Moonen, "Consensus-based distributed total least squares estimation in ad hoc wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2320–2330, May 2011.
- [14] —, "Low-complexity distributed total least squares estimation in ad hoc sensor networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4321–4333, Aug 2012.
- [15] S. Huang and C. Li, "Distributed sparse total least-squares over networks," *IEEE Trans. Signal Process.*, vol. 63, no. 11, pp. 2986–2998, Jun 2015.
- [16] R. Lopez-Valcarce, S. S. Pereira, and A. Pages-Zamora, "Distributed Total Least Squares estimation over networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2014, pp. 7580–7584.
- [17] R. Arablouei, S. Werner, and K. Doğançay, "Diffusion-based distributed adaptive estimation utilizing gradient-descent total least-squares," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 5308–5312.
- [18] R. Abdolee and B. Champagne, "Diffusion LMS strategies in sensor networks with noisy input data," *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, pp. 3–14, Feb 2016.
- [19] L. Lu, H. Zhao, and B. Champagne, "Diffusion total least-squares algorithm with multi-node feedback," *Signal Processing*, Jul 2018.
- [20] S. Silva Pereira, A. Pages-Zamora, and R. Lopez-Valcarce, "Distributed TLS estimation under random data faults," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Apr 2015, pp. 2949–2953.
- [21] C. Li, S. Huang, Y. Liu, and Y. Liu, "Distributed TLS over multitask networks with adaptive intertask cooperation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 6, pp. 3036–3052, Dec 2016.
- [22] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3122–3136, Jul 2008.
- [23] W. Dinkelbach, "On nonlinear fractional programming," *Management Science*, vol. 13, no. 7, pp. 492–498, Mar 1967.
- [24] Z.-Q. Luo, W.-K. Ma, A. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization Problems," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 20–34, May 2010.
- [25] R. Arablouei, S. Werner, and K. Doğançay, "Analysis of the gradient-descent total least-squares adaptive filtering algorithm," *IEEE Transactions on Signal Processing*, vol. 62, no. 5, pp. 1256–1264, Mar 2014.
- [26] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [27] J.-P. Crouzeix and J. A. Ferland, "Algorithms for generalized fractional programming," *Mathematical Programming*, vol. 52, no. 1, pp. 191–207, May 1991.
- [28] G. Pataki, "On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues," *Mathematics of Operations Research*, vol. 23, no. 2, pp. 339–358, Jan. 1998.
- [29] E. Dall'Anese, H. Zhu, and G. B. Giannakis, "Distributed optimal power flow for smart microgrids," *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1464–1475, Sep 2013.