2018 Annual American Control Conference (ACC)
June 27–29, 2018. Wisconsin Center, Milwaukee, USA

Cloud-aided collaborative estimation by ADMM-RLS algorithms for connected diagnostics and prognostics

Valentina Breschi[1], Ilya Kolmanovsky[2], Alberto Bemporad[1]

***Abstract*— As the connectivity of consumer devices is rapidly growing and cloud computing technologies are becoming more widespread, cloud-aided algorithms for parameter estimation can be developed to exploit the theoretically unlimited storage memory and computational power of the "cloud", while relying on information provided by multiple sources. With the ultimate goal of developing monitoring, diagnostic and prognostic strategies, this paper focuses on the design of a Recursive Least-Squares (RLS) based estimator for identification over a multitude of similar devices (such as a mass production) connected to the cloud. The proposed approach, that relies on Node-to-Cloud-to-Node (N2C2N) transmissions, is designed so that: (i) estimates of the unknown parameters are computed locally and (ii) the local estimates are refined on the cloud by exploiting the additional information that the devices have similar characteristics. The proposed approach requires minimal changes to local (pre-existing) RLS estimators.**

I. INTRODUCTION

With the increasing connectivity between devices, the interest in distributed solutions (e.g., for control [4] and machine learning [3]) has been rapidly growing. In particular, the problem of parameter estimation over networks has been extensively studied in the context of Wireless Sensor Networks (WSNs) [2], [9], [10]. Due to the low communication power of the nodes in WSNs, research has mainly been devoted to obtain fully distributed approaches, i.e., methods that allow exchanges of information between neighbor nodes only. Even though such a choice enables to reduce multi-hop transmissions and improve robustness to node failures, convergence speed to reach consensus is limited by the fact that nodes only communicate with neighbors. Moreover, to attain a global consensus the topology of the network must be carefully chosen to enable exchanges of information between different groups of neighbor nodes.

At the same time, with recent advances in cloud computing [11] it is possible to acquire and release resources with minimum effort so that each node can have on-demand access to shared resources, theoretically characterized by unlimited storage space and computational power. This motivates to reconsider the approach towards a more centralized strategy, where some computations are still performed at the node level, but the most time and memory consuming ones are executed on the cloud. This requires the communication between the nodes and a fusion center, i.e., the cloud, where the data gathered from the nodes are properly merged.

[1]Alberto Bemporad is with the IMT School for Advanced Studies Lucca, Piazza San Francesco 19, 55100 Lucca, Italy. At the time of submission Valentina Breschi was with the IMT School for Advanced Studies Lucca. valentina.breschi@imtlucca.it; alberto.bemporad@imtlucca.it

[2]Ilya Kolmanovsky is with the Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA. ilya@umich.edu

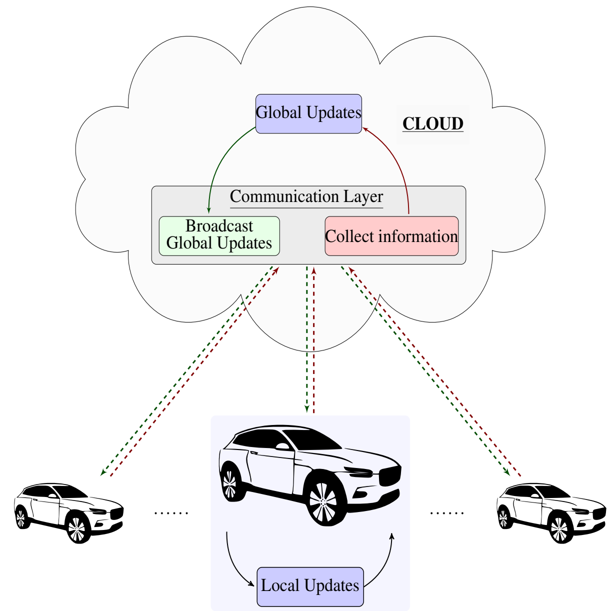


Fig. 1. Cloud-connected vehicles.

Cloud computing has been considered for automotive vehicle applications in [6], [7] and [12]. As a motivating example for another potential automotive application, consider a vehicle fleet with vehicles connected to the cloud (see Figure 1). In such a setting, measurements taken on-board of the vehicles can be used for cloud-based diagnostics and prognostics purposes. In particular, the measurements can be used to estimate parameters that may be common to all vehicles, such as parameters in models of component wear or fuel consumption models, and parameters that may be specific to each individual vehicle. References [5], [13] suggest potential applications of such approaches for prognostics of automotive fuel pumps and brake pads. Specifically, the rate of component wear as a function of the workload (cumulative fuel flow or energy dissipated in the brakes) can be estimated as a common parameter to all vehicles or at least to all vehicles in the same class.

In this paper a centralized approach for recursive estimation of parameters in the least-squares sense is presented. The method has been designed under the hypothesis of (i) ideal transmission, i.e., the information exchanged between the cloud and the nodes is not corrupted by noise, and the assumption that (ii) all the nodes are described by the same model, which is supposed to be known a priori. Differently from what is done in many distributed estimation methods (e.g., see [10]), where the nodes estimate common unknown parameters, the strategy we propose allows one to account for more general consensus constraints. As a consequence, our method can be applied to problems where only a subset of the unknowns is common to all the nodes, while other parameters are purely local, i.e., they are different for each node. A priori specified constraints on the parameters can also be handled by our algorithms.

Our estimation approach is based on defining a separable optimization problem which is then solved through the Alternating Direction Method of Multipliers (ADMM), similarly to what has been done in [10] but performing the computation both locally and on the cloud instead of using a fully distributed scheme. The estimation problem is thus solved through a two-step strategy: (i) local estimates are recursively retrieved by each node using the measurements acquired from the sensors available locally; (ii) global computations are performed to refine the local estimates, which are supposed to be transmitted to the cloud by each node. Note that, based on the aforementioned characteristics, back and forth transmissions to the cloud are required. A transmission scheme referred to as Node-to-Cloud-to-Node (N2C2N) is thus employed.

The main features of our strategies are: (i) the use of recursive formulas to update the local estimates of the unknown parameters; (ii) the possibility to account for the presence of both purely local and global parameters, that can be estimated in parallel; (iii) the straightforward integration of the proposed techniques with pre-existing Recursive Least-Squares (RLS) estimators already running on board of the nodes.

The paper is organized as follows. Section II is devoted to the problem formulation and the presentation of our basic approach, along with the display of the results of simulation examples that show the effectiveness of the approach and its performance in different scenarios. In Section III, the approach proposed in Section II is extended to be able to deal with the more general case of constrained consensus-based estimation. The results of a simulation example are then reported. Concluding remarks and directions for future research are summarized in Section IV.

A. Notation

Let $\mathbb{R}^{\mathsf{n}}$ be the set of real vectors of dimension n, $\mathbb{N}$ is the set of natural numbers and $\mathbb{R}^{+}$ be the set of positive real numbers, excluding zero. Given a set $\mathcal{A}$, let $\bar{\mathcal{A}}$ be the complement of $\mathcal{A}$ and $\mathcal{P}_{\mathcal{A}}$ denote the Euclidean projection onto $\mathcal{A}$. Given a vector $a \in \mathbb{R}^{\mathsf{n}}$, a_i denotes the ith component of a and $\|a\|_2$ is the Euclidean norm of a. Let A' denote the transpose of A, with $A \in \mathbb{R}^{\mathsf{n}\times\mathsf{p}}$, I_{n} be the identity matrix of size n and 0_{n} be an n-dimensional column vector of ones.

II. Collaborative Estimation for Partial Consensus

Suppose that the output/regressor pairs $\{y_n(\tau), X_n(\tau)\}_{\tau=1}^{t}$, $n = 1, \ldots, N$, collected from N data sources until time t, are available to identify a set of unknown parameters. Furthermore, assume that the N data-generating systems share the same model, which is supposed to be accurately approximated as

$$y_n(\tau) = X_n(\tau)'\theta_n + e_n(\tau) \text{ for } n = 1, \ldots, N, \qquad (1)$$

where $y_n(\tau) \in \mathbb{R}^{n_y}$, $\theta_n \in \mathbb{R}^{n_\theta}$ is the vector of unknown parameters to be estimated and e_n is a zero-mean white noise sequence independent on X_n. As the behavior of all the N systems is supposed to be described by the same model, n_y and n_θ are equal for all $n \in \{1, \ldots, N\}$.

In addition, let us suppose that there exists a parameter vector $\theta^g \in \mathbb{R}^{n_g}$, with $n_g \leq n_\theta$, such that

$$P\theta_n = \theta^g, \ \forall\ n \in \{1, \ldots, N\} \qquad (2)$$

with $P \in \mathbb{R}^{n_g \times n_\theta}$ known a priori. Depending on how P is chosen, different situations can be considered, e.g., $\theta_m = \theta_n$ $\forall n, m \in \{1, \ldots, N\}$ with $n \neq m$, if $P = I_{n_\theta}$.

To exploit all the available information to generate least-square estimates of $\{\theta_n\}_{n=1}^{N}$ and θ^g, we formulate the following problem

$$\begin{aligned} \min_{\{\theta_n\}_{n=1}^{N}} \quad & \sum_{n=1}^{N} f_n(\theta_n) \\ \text{s.t.} \quad & P\theta_n = \theta^g, \quad n = 1, \ldots, N \end{aligned} \qquad (3)$$

with $f_n : \mathbb{R}^{n_\theta} \to \mathbb{R} \cup \{+\infty\}$ equal to

$$f_n(\theta_n) = \frac{1}{2}\sum_{\tau=1}^{t} \left[\lambda_n^{t-\tau} \|y_n(\tau) - X_n(\tau)'\theta_n\|_2^2\right], \qquad (4)$$

where $\lambda_n \in (0, 1]$ represents the forgetting factor [8] associated to the nth data-generating system. Problem (3) should be solved so to (i) to retrieve an estimate for θ_n locally and (ii) to refine such estimates and identify θ^g using the data gathered from the N sources. As a consequence, (i) N local processors and (ii) and the cloud, where the data are merged, are required. Under the hypothesis that the computational power available locally is limited, it is also desirable to update the estimates recursively to reduce the amount of computations that has to be performed by the N local processors.

The Alternating Direction Method of Multipliers (ADMM) [1] can be used to solve problem (3). According to [1], we define the augmented Lagrangian associated with (3) as

$$\mathcal{L} = \sum_{n=1}^{N} \left\{ f_n(\theta_n) + \delta_n'(P\theta_n - \theta^g) + \frac{\rho}{2}\|P\theta_n - \theta^g\|_2^2 \right\}, \qquad (5)$$

where $\rho \in \mathbb{R}^{+}$ is a tunable parameter and $\delta_n \in \mathbb{R}^{n_g}$ is the Lagrange multiplier associated with the consensus constraint. The ADMM steps that have to be performed to solve problem (3) are thus:

$$\hat{\theta}_n^{(k+1)}(t) = \underset{\theta_n}{\operatorname{argmin}}\ \mathcal{L}(\theta_n, \hat{\theta}^{g,(k)}, \delta_n^{(k)}), \qquad (6a)$$

$$\hat{\theta}^{g,(k+1)} = \underset{\theta^g}{\operatorname{argmin}}\ \mathcal{L}(\{\hat{\theta}_n^{(k+1)}(t)\}_{n=1}^{N}, \theta^g, \{\delta_n^{(k)}\}_{n=1}^{N}), \qquad (6b)$$

$$\delta_n^{(k+1)} = \delta_n^{(k)} + \rho(P\hat{\theta}_n^{(k+1)}(t) - \hat{\theta}^{g,(k+1)}), \qquad (6c)$$

with $k \in \mathbb{N}$ indicating the ADMM iteration. It has to be pointed out that (6a)–(6c) are similar to the formulas proposed in [1] to solve the consensus problem, with the exception that in this case the more general consensus constraint (2) is considered.

Let us focus on the update of $\hat{\theta}^g$. The explicit solution for (6b) is:

$$\hat{\theta}^{g,(k+1)} = \frac{1}{N}\sum_{n=1}^{N}\left(P\hat{\theta}_n^{(k+1)}(t) + \frac{1}{\rho}\delta_n^{(k)}\right). \tag{7}$$

The updated estimate $\hat{\theta}^{g,(k+1)}$ is thus computed as the combination of the sample means of $\{P\hat{\theta}_n^{(k+1)}(t)\}_{n=1}^N$ and of the dual variables $\delta_n^{(k)}$.

The explicit solution of (6a), with f_n defined as in (4), is:

$$\hat{\theta}_n^{(k+1)}(t) = \phi_n(t)\left\{\mathcal{Y}_n(t) + P'(\rho\hat{\theta}^{g,(k)} - \delta_n^{(k)})\right\}, \tag{8}$$

where

$$\mathcal{Y}_n(t) = \sum_{\tau=1}^{t}\left[\lambda_n^{t-\tau}X_n(\tau)y_n(\tau)\right], \tag{9}$$

$$\phi_n(t) = \left(\left[\sum_{\tau=1}^{t}\lambda_n^{t-\tau}X_n(\tau)X_n(\tau)'\right] + \rho P'P\right)^{-1}. \tag{10}$$

As we want to find recursive updates for the local estimates $\hat{\theta}_n$, consider the expression for the nth local estimate obtained at $t-1$ after ADMM iterations are terminated

$$\hat{\theta}_n(t-1) = \phi_n(t)\Big[\mathcal{Y}_n(t-1) + P'(\rho\hat{\theta}^g(t-1) - \delta_n(t-1))\Big], \tag{11}$$

where $\hat{\theta}^g(t-1)$ and $\delta_n(t-1)$ are the global estimate and the Lagrange multipliers, respectively, at step $t-1$.

Based on the definition of $\phi_n(t)$ in (10), it can be proven that $\phi_n(t)$ can be recursively updated as

$$\mathcal{R}_n(t) = \lambda_n I_{n_{\tilde{X}}} + \tilde{X}_n(t)'\phi_n(t-1)\tilde{X}_n(t),$$
$$K_n(t) = \phi_n(t-1)\tilde{X}_n(t)\mathcal{R}_n(t)^{-1}, \tag{12}$$
$$\phi_n(t) = \lambda_n^{-1}(I_{n_\theta} - K_n(t)\tilde{X}_n(t)')\phi_n(t-1), \tag{13}$$

with $n_{\tilde{X}} = n_y + n_g$ and

$$\tilde{X}_n(t) = \left[\, X_n(t) \;\; \sqrt{\rho(1-\lambda_n)}P' \,\right] \in \mathbb{R}^{n_\theta \times n_{\tilde{X}}}. \tag{14}$$

We note that the updates (12)-(13) agree with the standard Recursive Least Squares (RLS) algorithm [8], with $\tilde{X}_n(t)$ replacing the regressor $X_n(t)$ and the dimensions of the identity matrix in (12) being properly changed. Furthermore, the updates (12)-(13) are independent of the ADMM iteration k, and they depend on local quantities only. Consequently, $K_n(t)$ and $\phi_n(t)$ can be updated by the local processor once per time-step using the recursive formulas (12)-(13).

Furthermore, it can be proven that $\hat{\theta}_n^{(k+1)}(t)$ can be computed as

$$\hat{\theta}_n^{(k+1)}(t) = \hat{\theta}_n^{rls}(t) + \hat{\theta}_n^{admm,(k+1)}(t), \tag{15}$$

where

$$\hat{\theta}_n^{admm,(k+1)}(t) = \phi_n(t)P'\left(\rho\Delta_{g,\lambda_n}^{(k+1)}(t) - \Delta_{\lambda_n}^{(k+1)}\right), \tag{16}$$

$$\hat{\theta}_n^{rls}(t) = \hat{\theta}_n(t-1) + K_n(\tilde{y}_n(t) - \tilde{X}_n(t)'\hat{\theta}_n(t-1)), \tag{17}$$

and $\tilde{y}_n(t) = \left[\, y_n(t)' \;\; 0_{1\times n_g} \,\right]'$.

The quantities $\Delta_{g,\lambda_n}^{(k+1)}(t)$ and $\Delta_{\lambda_n}^{(k+1)}$ in (16) are defined as

$$\Delta_{g,\lambda_n}^{k+1}(t) = \hat{\theta}^{g,(k)} - \lambda_n\hat{\theta}^g(t-1), \tag{18a}$$

$$\Delta_{\lambda_n}^{(k+1)}(t) = \delta_n^{(k)} - \lambda_n\delta_n(t-1). \tag{18b}$$

Based on (18a), $\hat{\theta}_n^{admm,(k+1)}(t)$ depends on the global estimate $\hat{\theta}^{g,(k)}$. Consequently, at each step $t \in \mathbb{N}$, $\hat{\theta}_n^{admm,(k+1)}(t)$ should be computed as in (16) on the cloud, not to require the local processors and the center of fusion to exchange information at each ADMM iteration. On the other hand, $\hat{\theta}_n^{rls}(t)$ can be updated recursively and once per time-step by the n-th local processor using Recursive Least Squares (17), thus allowing to integrate the proposed approach with pre-existing RLS estimators available locally.

The approach, summarized in Algorithm 1, thus requires the local processors to transmit $\{\hat{\theta}_n^{rls}, \phi_n\}$ to the cloud, while the cloud has to communicate $\hat{\theta}_n$ to each system.

Remark 1: Algorithm 1 requires the initialization of the local and global estimates. If some data $\{y_n(\tau), X_n(\tau)\}_{\tau=1}^{T_o}$ are available to be processed in a batch mode, $\hat{\theta}_n(0)$ can be chosen as the best linear model

$$\hat{\theta}_n(0) = \underset{\theta}{\operatorname{argmin}} \sum_{\tau=1}^{T_o}\|y_n(\tau) - X_n(\tau)'\theta\|_2^2$$

and $\hat{\theta}^g(0)$ can be computed as the mean of $\{P\hat{\theta}_n(0)\}_{n=1}^N$. Moreover, the matrices ϕ_n, $n = 1, \ldots, N$, can be initialized as $\phi_n(0) = \gamma I_{n_\theta}$, with $\gamma > 0$. ■

A. Example 1 ($\theta^g = \theta_n$)

Suppose that the N systems are described by the following model

$$y_n(t) = 0.9y_n(t-1) + 0.4u_n(t-1) + e_n(t), \tag{19}$$

where $y_n(t) \in \mathbb{R}$, $X_n(t) = \begin{bmatrix} y_n(t-1) & u_n(t-1)\end{bmatrix}'$, u_n is known and it is generated as a sequence of i.i.d. elements uniformly distributed in the interval [2, 3], and $e_n \sim \mathcal{N}(0, R_n)$ is a white noise sequence, with $\{R_n \in \mathbb{N}\}_{n=1}^N$ randomly chosen in the interval [1, 30]. Evaluating the effect of the noise on the output y_n through the Signal-to-Noise Ratio SNR_n, i.e.,

$$\text{SNR}_n = 10\log\frac{\sum_{t=1}^{T}(y_n(t) - e_n(t))^2}{\sum_{t=1}^{T}e_n(t)^2}\ dB \tag{20}$$

the chosen covariance matrices yield to SNR_n in the interval [7.8, 20.8] dB, $n = 1, \ldots, N$.

Initializing ϕ_n as $\phi_n(0) = 0.1I_{n_\theta}$, while $\hat{\theta}_n(0)$ and $\hat{\theta}_o^g$ are sampled from the distributions $\mathcal{N}(\hat{\theta}^g, 2I_{n_\theta})$ and $\mathcal{N}(\hat{\theta}^g, I_{n_\theta})$, respectively, $\{\lambda_n = \Lambda\}_{n=1}^N$, with $\Lambda = 1$, and $\rho = 0.1$, the performance of the proposed approach are quantitatively

Algorithm 1 ADMM-RLS for partial consensus

Input: Data $X_n(1), y_n(1), X_n(2), y_n(2), \dots$, initial matrices $\phi_n(0) \in \mathbb{R}^{n_\theta \times n_\theta}$, initial local estimates $\hat{\theta}_n(0)$, initial Lagrange multipliers $\delta_{n,\text{o}}$, forgetting factors λ_n, $n = 1, \dots, N$, initial global estimate $\hat{\theta}^g_{\text{o}}$, parameter $\rho \in \mathbb{R}^+$.

1. **iterate for** $t = 1, 2, \dots$

   <u>**Local**</u>

   1.1. **for** $n = 1, \dots, N$ **do**

   1.1.1. **compute** $\tilde{X}_n(t)$ as in (14);

   1.1.2. **compute** $K_n(t)$ and $\phi_n(t)$ with (12) - (13);

   1.1.3. **compute** $\hat{\theta}^{rls}_n(t)$ with (17);

   1.2. **end for**;

   <u>**Global**</u>

   1.1. **do**

   1.1.1. **compute** $\hat{\theta}^{admm,(k+1)}_n(t)$ with (16), $n = 1, \dots, N$;

   1.1.2. **compute** $\hat{\theta}^{(k+1)}_n(t)$ as in (15);

   1.1.3. **compute** $\hat{\theta}^{g,(k+1)}$ with (7);

   1.1.4. **compute** $\delta^{(k+1)}_n$ with (6c), $n = 1, \dots, N$;

   1.1.5. $k \leftarrow k + 1$;

   1.2. **until** a stopping criteria is satisfied (e.g. maximum number of iterations attained);

2. **end**.

Output: Estimated global parameters $\hat{\theta}^g(t)$, estimated local parameters $\hat{\theta}_n(t)$, $n = 1, \dots, N$.

TABLE I
ADMM-RLS: $\|\text{RMSE}^g\|_2$

N \ T	10	10^2	10^3	10^4
2	1.07	0.33	0.16	0.10
10	0.55	0.22	0.09	0.03
10^2	0.39	0.11	0.03	0.01

assessed for different values of N and T through the Root Mean Square Error (RMSE)

$$\text{RMSE}^g_i = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(\theta^g_i - \hat{\theta}^g_i(t)\right)^2}, \quad i = 1, \dots, n_g. \qquad (21)$$

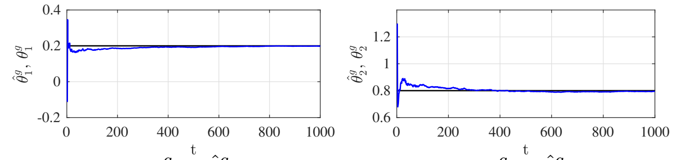
As shown in Table I the accuracy of the estimates tends to increase if the number of data sources N and the estimation horizon T increase. For $N = 100$ and $T = 1000$, the estimates obtained applying Algorithm 1 are compared with the ones retrieved performing RLS on the cloud using the lumped data pairs $\{\breve{y}(t), \breve{X}(t)\}_{t=1}^{T}$[1]. As expected, both the approaches lead to the same $RMSE^g = 0.03$.

B. Example 2: Non-informative systems

Consider a set of $N = 100$ dynamical systems modelled as

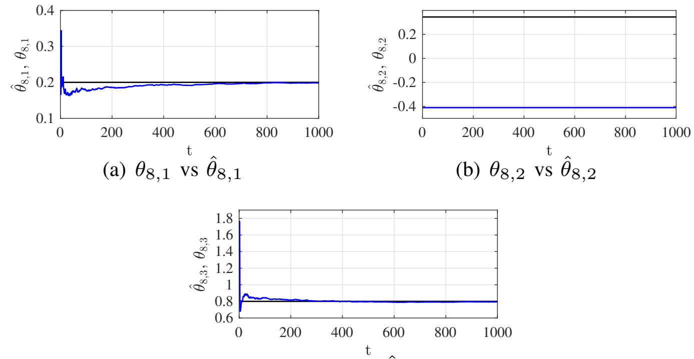
$$y_n(t) = \theta^g_1 y_n(t-1) + \theta_{n,2} y_n(t-2) + \theta^g_2 u_n(t-1) + e_n(t), \quad (22)$$

[1] $\breve{y}(t)$ is obtained stacking all the measured output $\{y_n(t)\}_{n=1}^{N}$ in a single vector. The lumped regressor $\breve{X}(t)$ is built similarly.



(a) θ^g_1 vs $\hat{\theta}^g_1$ (b) θ^g_2 vs $\hat{\theta}^g_2$

Fig. 2. Global parameters. Black : true, blue : estimate with ADMM-RLS.



(a) $\theta_{8,1}$ vs $\hat{\theta}_{8,1}$ (b) $\theta_{8,2}$ vs $\hat{\theta}_{8,2}$ (c) $\theta_{8,3}$ vs $\hat{\theta}_{8,3}$

Fig. 3. Local parameters $\theta_{8,i}$, $i = 1, 2, 3$. Black : true, blue : ADMM-RLS.

where $\theta^g = [0.2 \;\; 0.8]'$, $\theta_{n,2}$ is sampled from the normal distribution $\mathcal{N}(0.4, 0.0025)$, so that it is different for the N systems, and $e_n \sim \mathcal{N}(0, R_n)$. The noise covariance matrices R_n are randomly chosen within the interval $[1, \; 20]$ and, thus, $SNRs$ are in the interval $[3.1, 14.6]$. We assume that $N_{ni} = 20$ systems randomly chosen among the N available data sources are not excited enough to be able to retrieve locally an accurate estimate of the unknowns [8]. Such a condition is simulated by setting $u_n = 0$ and $R_n = 10^{-8}$. By using the same initial setting and parameters as in Example 1, we obtain the global estimates reported in Figure 2. The resulting estimates $\{\hat{\theta}^g_i\}_{i=1}^2$ converge to the actual values of the global parameters even if 20% of the systems provide non-informative data.

The local estimates $\{\hat{\theta}_{8,i}\}_{i=1}^3$ are reported in Figure 3, with the 8th system being among the ones with non exciting inputs. As shown in Figure 3(b) $\hat{\theta}_{8,2}$ is constant ($\hat{\theta}_{8,2} = \hat{\theta}_{8,2}(0)$) over the estimation horizon, as expected. Instead, the proposed collaborative approach allows us to accurately estimate $\theta_{8,1}$ and $\theta_{8,3}$ (see Figures 3(a) and 3(c)). We can thus conclude that the proposed estimation method "forces" the estimates of the global components of θ_n to follow $\hat{\theta}^g$, while contributions from the systems that lacked excitation are discarded.

III. Constrained collaborative estimation

Suppose that an additional constraint is added to problem (3), so that the optimization problem to be solved takes the form

$$\begin{aligned} &\text{minimize} && \sum_{n=1}^{N} f_n(\theta_n) \\ &\text{s.t.} && P\theta_n = \theta^g \quad n = 1, \dots, N, \\ & && \theta_n \in \mathcal{C}_n, \quad n = 1, \dots, N, \end{aligned} \qquad (23)$$

with f_n as in (4) and $\mathcal{C}_n$ being a convex set representing constraints on the parameter values. These additional constraints can be used to enforce the global parameter θ^g to belong to a convex set $\mathcal{C}$ (with $\mathcal{C}_n = \mathcal{C} \cup \{\mathcal{C}_n \cap \bar{\mathcal{C}}\}$).

Following [1], the new optimization problem in (23) can be reformulated as

$$\begin{aligned} \text{minimize} \quad & \sum_{n=1}^{N} \{f_n + g_n(z_n)\} \\ \text{s.t.} \quad & P\theta_n = \theta^g \quad n = 1, \dots, N, \\ & \theta_n = z_n, \quad n = 1, \dots, N, \end{aligned} \tag{24}$$

where $\{z_n \in \mathbb{R}^{n_\theta}\}_{n=1}^N$ are auxiliary variables and $\{g_n\}_{n=1}^N$ are the indicator functions of the sets $\{\mathcal{C}_n\}_{n=1}^N$, with

$$g_n(z_n) = \begin{cases} 0 \text{ if } z_n \in \mathcal{C}_n \\ +\infty \text{ otherwise.} \end{cases} \tag{25}$$

The augmented Lagrangian associated with the considered problem is:

$$\mathcal{L} = \sum_{n=1}^{N} \Big\{ f_n(\theta_n) + g_n(z_n) + \delta_{n,1}'(\theta_n - z_n) + \\ + \delta_{n,2}'(P\theta_n - \theta^g) + \frac{\rho_1}{2}\|\theta_n - z_n\|_2^2 + \frac{\rho_2}{2}\|P\theta_n - \theta^g\|_2^2 \Big\}, \tag{26}$$

where two sets of Lagrange multipliers, $\{\delta_{n,1} \in \mathbb{R}^{n_g}\}_{n=1}^N$ and $\{\delta_{n,2} \in \mathbb{R}^{n_\theta}\}_{n=1}^N$, have been introduced.

Based on [1], the ADMM steps that have to be performed to solve (24) are:

$$\hat{\theta}_n^{(k+1)}(t) = \underset{\theta_n}{\operatorname{argmin}}\ \mathcal{L}(\theta_n, \hat{\theta}^{g,(k)}, z_n^{(k)}, \delta_n^{(k)}), \tag{27a}$$

$$z_n^{(k+1)} = \underset{z_n}{\operatorname{argmin}}\ \mathcal{L}(\hat{\theta}_{n,(k+1)}(t), \hat{\theta}^{g,(k)}, z_n, \delta_n^{(k)}), \tag{27b}$$

$$\hat{\theta}^{g,(k+1)} = \underset{\theta^g}{\operatorname{argmin}}\ \mathcal{L}(\{\hat{\theta}_n^{(k+1)}, z_n^{(k+1)}, \delta_n^{(k)}\}_{n=1}^N, \theta^g), \tag{27c}$$

$$\delta_{n,1}^{(k+1)} = \delta_{n,1}^{(k)} + \rho_1(\hat{\theta}_n^{(k+1)}(t) - z^{(k+1)}), \tag{27d}$$

$$\delta_{n,2}^{(k+1)} = \delta_{n,2}^{(k)} + \rho_2(P\hat{\theta}_n^{(k+1)}(t) - \hat{\theta}^{g,(k+1)}). \tag{27e}$$

Solving (27b)–(27c), the updates for the auxiliary variables and $\hat{\theta}^g$ are:

$$z_n^{(k+1)} = \mathcal{P}_{\mathcal{C}_n}\left(\hat{\theta}_n^{(k+1)}(t) + \frac{1}{\rho_1}\delta_{n,1}^{(k)}\right), \tag{28}$$

$$\hat{\theta}^{g,(k+1)} = \frac{1}{N}\sum_{n=1}^{N}\left(P\hat{\theta}_n^{(k+1)}(t) + \frac{1}{\rho_2}\delta_{n,2}^{(k)}\right), \tag{29}$$

where $z_n^{(k+1)}$ is obtained through a projection onto the set $\mathcal{C}_n$. It has to be pointed out that both the z and the Lagrange multiplier updates in (28) and (27d), respectively, depend on local quantities only. However, (28) and (27d) depend also on the ADMM iteration and, consequently, z_n and $\delta_{n,1}$ should be updated on the cloud.

Consider the local update (27a). The explicit solution of (27a) is given by,

$$\hat{\theta}_n^{(k+1)}(t) = \phi_n(t)\Big\{\mathcal{Y}_n(t) - \delta_{n,1}^{(k)} - P'\delta_{n,2}^{(k)} + \\ + \rho_1 z_n^{(k)} + \rho_2 P'\hat{\theta}^{g,(k)}\Big\}, \tag{30}$$

with $\mathcal{Y}_n$ defined as in (9) and

$$\phi_n(t) = \left(\left[\sum_{\tau=1}^{t} \lambda_n^{t-\tau} X_n(\tau) X_n(\tau)'\right] + \rho_1 I_{n_\theta} + \rho_2 P'P\right)^{-1}. \tag{31}$$

As the ultimate goal is to obtain recursive formulas to update $\hat{\theta}_n$, consider the estimates at $t-1$, obtained once the stopping criteria for ADMM has been satisfied:

$$\hat{\theta}_n(t-1) = \phi_n(t-1)\{\mathcal{Y}_n(t-1) - \delta_{n,1}(t-1) + \\ - P'\delta_{n,2}(t-1) + \rho_1 z_n(t-1) + \rho_2 P'\hat{\theta}^g(t-1)\}, \tag{32}$$

where $\delta_{n,1}(t-1)$, $\delta_{n,2}(t-1)$, $z_n(t-1)$ and $\hat{\theta}^g(t)$ are the estimates obtained at $t-1$.
It can thus be proven that $\phi_n(t)$ can be updated as in (12)–(13), with the extended regressor $\tilde{X}_n(t)$ defined as

$$\tilde{X}_n(t) = \left[X_n(t)\ \sqrt{(1-\lambda_n)\rho_1} I_{n_\theta}\ \sqrt{(1-\lambda_n)\rho_2} P'\right], \tag{33}$$

with $\tilde{X}_n(t) \in \mathbb{R}^{n_\theta \times n_{\tilde{X}}}$ and $n_{\tilde{X}} = n_y + n_\theta + n_g$.
Furthermore, it can also be shown that $\hat{\theta}_n$ can be updated as in (15), where $\hat{\theta}_n^{rls}(t)$ is computed as in (17) with $\tilde{y}_n(t) = \left[\, y_n(t)'\ 0_{1\times n_\theta}\ 0_{1\times n_g} \,\right]$, and

$$\hat{\theta}_n^{admm,(k+1)}(t) = \phi_n(t)\left[\rho_1 \Delta_{z,\lambda_n}^{(k+1)}(t) + \rho_2 P' \Delta_{g,\lambda_n}^{(k+1)}(t) + \\ - \Delta_{1,\lambda_n}^{(k+1)} - P'\Delta_{2,\lambda_n}^{(k+1)}\right], \tag{34}$$

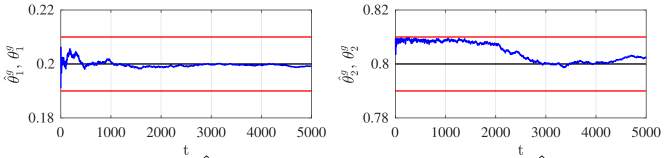
where

$$\begin{aligned} \Delta_{z,\lambda_n}^{(k+1)}(t) &= z_n^{(k)} - \lambda_n z_n(t-1) \\ \Delta_{g,\lambda_n}^{(k+1)}(t) &= \hat{\theta}^{g,(k)} - \lambda_n \hat{\theta}^g(t-1) \\ \Delta_{1,\lambda_n}^{(k+1)} &= \delta_{n,1}^{(k)} - \lambda_n \delta_{n,1}(t-1) \\ \Delta_{2,\lambda_n}^{(k+1)} &= \delta_{n,2}^{(k)} - \lambda_n \delta_{n,2}(t-1). \end{aligned}$$

The same observations made with respect to the computation of $\hat{\theta}_n^{admm}$ and $\hat{\theta}_n^{rls}$ reported in Section II can be extended to the considered setting. The proposed ADMM-RLS scheme for constrained collaborative estimation is summarized in Algorithm 2.

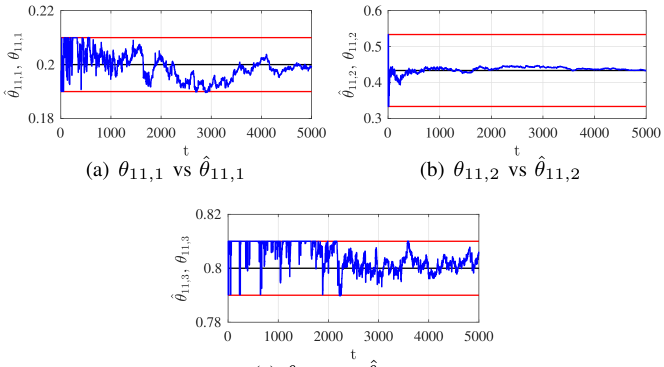
A. Example 3

Suppose that the data collected from $N = 100$ systems, described by the model in (22), over an estimation horizon $T = 5000$ are available. Furthermore, assume that we know a priori that $\theta_{n,1} \in [\,0.19,\ 0.21\,]$, $\theta_{n,3} \in [\,0.79,\ 0.81\,]$ and we constrain $\hat{\theta}_{n,2}$ in the set $[\,\theta_{n,2}-0.1,\ \theta_{n,2}+0.1\,]$. By using the same initial conditions and forgetting factors as in Example 2, with $\rho_1 = 10$ and $\rho_2 = 0.1$, we retrieve the global estimates reported in Figure 4. The estimated parameters satisfy the constraints on $\theta_{n,1}$ and $\theta_{n,3}$, thus proving that the constraints on the global estimate are automatically enforced imposing $\theta_n \in \mathcal{C}_n$. The local estimates for the 11-th system ($SNR_{11} = 10.6$ dB), i.e., $\{\hat{\theta}_{11,i}\}_{i=1}^3$, are shown in Figure 5. As expected, the estimates satisfy the imposed constraints and converge to the true values of the unknowns.



(a) θ_1^g vs $\hat{\theta}_1^g$ (b) θ_2^g vs $\hat{\theta}_2^g$

Fig. 4. Global parameters. Black : true, blue : ADMM-RLS, red : bounds.



(a) $\theta_{11,1}$ vs $\hat{\theta}_{11,1}$ (b) $\theta_{11,2}$ vs $\hat{\theta}_{11,2}$

(c) $\theta_{11,3}$ vs $\hat{\theta}_{11,3}$

Fig. 5. Local parameter θ_{11}. Black : true, blue : ADMM-RLS, red : bounds

IV. Concluding Remarks and Future Work

This paper has presented a method for collaborative least-squares parameter estimation based on output measurements from multiple systems which can perform local computations and are also connected to a centralized resource in the cloud. The approach includes two stages: (i) a *local* step, where estimates of the unknown parameters are obtained using the locally available data, and (ii) a *global* step, performed on the cloud, where the local estimates are fused.

Future research will address extensions of the method to the nonlinear and multi-class consensus cases. Moreover, an alternative solution will be studied so to replace the adopted N2C2N transmission scheme to alleviate problems associated with the communication latency between the cloud and the nodes. Other solutions to further reduce the transmission complexity and to obtain an asynchronous scheme with the same characteristics as the one presented in this paper will be also addressed in future research.

Algorithm 2 ADMM-RLS for constrained consensus

Input: Data $X_n(1), y_n(1), X_n(2), y_n(2), \ldots$, initial matrices $\phi_n(0) \in \mathbb{R}^{n_\theta \times n_\theta}$, initial local estimates $\hat{\theta}_n(0)$, initial Lagrange multipliers $\delta_{n,1}^{\text{o}}$ and $\delta_{n,2}^{\text{o}}$, initial auxiliary variables $z_{n,\text{o}}$, forgetting factors λ_n, $n = 1, \ldots, N$, initial global estimate $\hat{\theta}_{\text{o}}^g$, parameters $\rho_1, \rho_2 \in \mathbb{R}^+$.

1. **iterate for** $t = 1, 2, \ldots$

   <u>**Local**</u>

   1.1. **for** $n = 1, \ldots, N$ **do**

   1.1.1. **compute** $\tilde{X}_n(t)$ as in (33);

   1.1.2. **compute** $K_n(t)$ and $\phi_n(t)$ with (12) - (13);

   1.1.3. **compute** $\hat{\theta}_n^{rls}(t)$ with (17);

   1.2. **end for**;

   <u>**Global**</u>

   1.1. **do**

   1.1.1. **compute** $\hat{\theta}_n^{admm,(k+1)}(t)$ with (34), $n = 1, \ldots, N$;

   1.1.2. **compute** $\hat{\theta}_n(t)$ with (15), $n = 1, \ldots, N$;

   1.1.3. **compute** $z_n^{(k+1)}$ with (28), $n = 1, \ldots, N$;

   1.1.4. **compute** $\hat{\theta}^{g,(k+1)}$ with (29);

   1.1.5. **compute** $\delta_{n,1}^{(k+1)}$ with (27d), $n = 1, \ldots, N$;

   1.1.6. **compute** $\delta_{n,2}^{(k+1)}$ with (27e), $n = 1, \ldots, N$;

   1.1.7. $k \leftarrow k + 1$;

   1.2. **until** a stopping criteria is satisfied (e.g. maximum number of iterations attained);

2. **end**.

Output: Estimated global parameters $\hat{\theta}^g(t)$, estimated local parameters $\hat{\theta}_n(t)$, $n = 1, \ldots, N$.

Acknowledgment

The first and second author thank Professor Anouck Girard for the opportunity to discuss progress in this research.

References

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, January 2011.

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

[3] P.A. Forero, A. Cano, and G.B. Giannakis. Consensus-based distributed support vector machines. *The Journal of Machine Learning Research*, 11:1663–1707, Aug 2010.

[4] F. Garin and L. Schenato. *A Survey on Distributed Estimation and Control Applications Using Linear Consensus Algorithms*, pages 75–107. Springer London, London, 2010.

[5] M.N. Howell, J.P. Whaite, P. Amatyakul, Y.K. Chin, M.A. Salman, C.H. Yen, and M.T. Riefe. Brake pad prognosis system, Apr 2010. US Patent 7,694,555.

[6] Z. Li, I. Kolmanovsky, E. Atkins, J. Lu, D. P. Filev, and J. Michelini. Road risk modeling and cloud-aided safety-based route planning. *IEEE Transactions on Cybernetics*, 46(11):2473–2483, Nov 2016.

[7] Z. Li, I. Kolmanovsky, E. M. Atkins, J. Lu, D. P. Filev, and Y. Bai. Road disturbance estimation and cloud-aided comfort-based route planning. *IEEE Transactions on Cybernetics*, PP(99):1–13, 2017.

[8] L. Ljung. *System identification: theory for the user*. Prentice-Hall Englewood Cliffs, NJ, 1999.

[9] C. G. Lopes and A. H. Sayed. Incremental adaptive strategies over distributed networks. *IEEE Transactions on Signal Processing*, 55(8):4064–4077, Aug 2007.

[10] G. Mateos, I. D. Schizas, and G. B. Giannakis. Distributed recursive least-squares for consensus-based in-network adaptive estimation. *IEEE Transactions on Signal Processing*, 57(11):4583–4588, Nov 2009.

[11] P. Mell and T. Grance. The nist definition of cloud computing. Technical Report 800-145, National Institute of Standards and Technology (NIST), Gaithersburg, MD, September 2011.

[12] E. Ozatay, S. Onori, J. Wollaeger, U. Ozguner, G. Rizzoni, D. Filev, J. Michelini, and S. Di Cairano. Cloud-based velocity profile optimization for everyday driving: A dynamic-programming-based solution. *IEEE Transactions on Intelligent Transportation Systems*, 15(6):2491–2505, Dec 2014.

[13] E. Taheri, O. Gusikhin, and I. Kolmanovsky. Failure prognostics for in-tank fuel pumps of the returnless fuel systems. In *Dynamic Systems and Control Conference*, Oct 2016.