A Distributed Model Identification Algorithm for Multi-Agent Systems

Preprint

Vivek Khatana,[1] Chin-Yao Chang,[2] and Wenbo Wang[2]

1 University of Minnesota, Minneapolis,
2 National Renewable Energy Laboratory

Presented at the Allerton Conference
Urbana, Illinois
September 25-27, 2024

NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Conference Paper
NREL/CP-5D00-89152
September 2024

A Distributed Model Identification Algorithm for Multi-Agent Systems

Preprint

Vivek Khatana,[1] Chin-Yao Chang,[2] and Wenbo Wang[2]

1 University of Minnesota, Minneapolis,
2 National Renewable Energy Laboratory

Suggested Citation
Khatana, Vivek, Chin-Yao Chang, and Wenbo Wang. 2024. *A Distributed Model Identification Algorithm for Multi-Agent Systems: Preprint*. Golden, CO: National Renewable Energy Laboratory. NREL/CP-5D00-89152. https://www.nrel.gov/docs/fy24osti/89152.pdf.

NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Conference Paper
NREL/CP-5D00-89152
September 2024

National Renewable Energy Laboratory
15013 Denver West Parkway
Golden, CO 80401
303-275-3000 • www.nrel.gov

NOTICE

This work was authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by DOE Office of Electricity, Advanced Grid Modeling Program, through agreement NO. 33652. The views expressed herein do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

Cover Photos by Dennis Schroeder: (clockwise, left to right) NREL 51934, NREL 45897, NREL 42160, NREL 45891, NREL 48097, NREL 46526.

NREL prints on paper that contains recycled content.

A Distributed Model Identification Algorithm for Multi-Agent Systems

Vivek Khatana, Chin-Yao Chang, Wenbo Wang

***Abstract*—In this study, we investigate agent-based approach for system model identification with emphasis on power distribution system applications. Departing from conventional practices of relying on historical data for offline model identification, we adopt online update approach utilizing real-time data by employing the latest data points for gradient computation. This methodology offers advantages including a large reduction in the communication network's bandwidth requirements by minimizing the data exchanged at each iteration and enabling the model to adapt in real-time to disturbances. Furthermore, we extend our model identification process from linear frameworks to more complex non-linear convex models. This extension is validated through numerical studies demonstrating improved control performance for a synthetic IEEE test case.**

***keywords*: Data-driven control, distributed optimization, model identification, online optimization, power grids.**

I. INTRODUCTION

System model identification is pivotal across numerous applications, where comprehending underlying processes is crucial for effective control and decision-making. Due to the complexity and dynamic nature of these systems, precise model identification is foundational to predictive analytics and system optimization. In this paper, we consider the following model formulation:

$$y(t) = g(u(t), \theta) \tag{1}$$

where $u(t) \in \mathbb{R}^{\mathbf{d_u}}$ and $y(t) \in \mathbb{R}^{\mathbf{d_y}}$ denote the input and output data at time t, and $\theta \in \mathbb{R}^{\mathbf{d}_\theta}$ contains the parameters to be identified. We define the dataset $\mathcal{M} := \{(u(t), y(t))\}_{t=1}^{T}$ compiled from inputs and outputs across times $t = 1, \cdots, T$, and aim to identify the optimal parameter θ^* that minimizes:

$$\mathbf{r}(\theta) = \tfrac{1}{2} \textstyle\sum_{t=1}^{T} \|y(t) - g(u(t), \theta)\|^2 . \tag{2}$$

In the context of power systems, especially at the distribution levels, model identification can be crucial for determining system topology, the LinDistFlow model, or the admittance matrix [1]–[4], which are essential for controlling power networks. Traditional approaches whether through conventional methods or machine learning techniques, typically rely on centralized data collection and processing. However, we contend that the decentralized nature of modern power systems, particularly with the increasing integration of distributed energy resources, calls for a multi-agent-based distributed approach [5], [6]. This strategy not only facilitates the autonomous operation of individual system components but also enhances the overall resilience of the network. Distributed algorithms promote localized decision-making, substantially reducing communication overhead and the risk of single points of failure inherent in centralized systems. Building on this foundation, our previous work [7] advanced distributed identification methods that protect local data privacy, although it was restricted to linear systems and demanded high bandwidth communication.

Building on the merits of the distributed and localized approach from our prior research, this paper introduces several advancements: (i) We have developed a distributed method for the identification of nonlinear systems using local input-output data; (ii) Our algorithm requires agents to use only current measurements for updates, thereby eliminating the need for storing historical data and substantially reducing communication bandwidth requirements. Moreover, it facilitates the sharing of non-linear estimates between agents during updates, enhancing the protection of individual agent parameters. (iii) Numerical studies demonstrate that identifying more accurate nonlinear models results in superior control performance compared to traditional linear models, underscoring the practical value of our advancements.

Notations and definitions: In this paper, we denote matrices in boldface. Let $\text{diag}\,(x)$ represent the matrix whose diagonal elements are the elements of the vector $x \in \mathbb{R}^n$. For matrices $\mathbf{A}^1, \mathbf{A}^2, \cdots, \mathbf{A}^N \in \mathbb{R}^{m\times n}$, we denote $\text{blkdiag}\,(\{\mathbf{A}^i\}_{i=1}^N)$ as the block diagonal matrix of the matrices $\mathbf{A}^i$. The vertical and horizontal concatenation of matrices $\mathbf{A}^i$ are denoted as $[\mathbf{A}^1; \mathbf{A}^2; \cdots; \mathbf{A}^n] \in \mathbb{R}^{Nm\times n}$ and $[\mathbf{A}^1, \mathbf{A}^2, \cdots, \mathbf{A}^n] \in \mathbb{R}^{m\times Nn}$. For a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\text{vec}(\mathbf{A}) \in \mathbb{R}^{mn}$ is a column vector created by concatenating the column vectors of $\mathbf{A}$ from left to right. For a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\text{null}\{\mathbf{A}\}$ denotes the null space of matrix $\mathbf{A}$. The scalar element of the i^{th} row and j^{th} column of $\mathbf{A}$ is denoted as $\mathbf{A}_{ij}$ and the j^{th} row and column of the matrix $\mathbf{A}$ are denoted as $\mathbf{A}_{j,:}$ and $\mathbf{A}_{:,j}$, respectively. The identity matrix and vector with all entries equal to 1 of dimension n are denoted as $\mathbf{I}_n$ and 1_n, respectively.

A graph $\mathcal{G}$ is denoted by a pair $(\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is a set of vertices (or nodes) and $\mathcal{E}$ is a set of edges, which are ordered subsets of two distinct elements of $\mathcal{V}$. If an edge from $j \in \mathcal{V}$ to $i \in \mathcal{V}$ exists then it is denoted as $(i, j) \in \mathcal{E}$. The set of neighboring sub-systems of node

V, Khatana is with Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, USA (Email: {khata010}@umn.edu).

C.-Y. Chang and W. Wang are with the National Renewable Energy Laboratory, Golden, CO 80401, USA (Email: {chinyao.chang, wenbo.wang}@nrel.gov).

This work was authored in part by NREL, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by DOE Office of Electricity, Advanced Grid Modeling Program, through agreement NO. 33652. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government.

$i \in \mathcal{V}$ is called the neighborhood of node i and is denoted by $\mathcal{N}_i = \{j \mid (i,j) \in \mathcal{E}\}$. In the subsequent, we use the terms agents, nodes, and sub-systems interchangeably. Given a norm $\|\cdot\|$ and a set $K \subset \mathbb{R}^p$, define the diameter of K with respect to this norm as $Diam_{\|\cdot\|}(K) := \sup_{x,y\in K} \|x-y\|$. In the subsequent text the $O(.)$ and $o(.)$ operations denote the standard *Big-O* and *Little-o* notations respectively [8].

II. Agent Based System Framework

In this section, we commence by demonstrating how (2) can be reinterpreted as a distributed optimization challenge across a network of sub-systems or agents. Subsequently, we delve into deriving analytical properties of these reformulations, which serve as the foundation for the convergence analysis of the online identification algorithm.

A. Distributed input-output data framework

Consider that model (1) is represented by an underlying network $\mathcal{G}(\mathcal{V},\mathcal{E})$ consisting of $|\mathcal{V}| := N$ sub-systems. Here, each sub-system has actuator and sensor measurements available within itself. Each agent $i \in \mathcal{V} := \{1, 2, \cdots, N\}$ has information on certain entries of the vectors $u(t)$ and $y(t)$ in (1) indexed by the sets $S_i^u \subset \{1, 2, \ldots, \mathbf{d_u}\}$ and $S_i^y \subset \{1, 2, \ldots, \mathbf{d_y}\}$ respectively. We assume the partitions S_i^u and S_i^y are such that $\mathbf{d_u}_i := |S_i^u|, \sum_{i=1}^N \mathbf{d_u}_i = \mathbf{d_u}, \mathbf{d_y}_i := |S_i^y|, \sum_{i=1}^N \mathbf{d_y}_i = \mathbf{d_y}$. Let $\mathrm{u}_i(t) \in \mathbb{R}^{\mathbf{d_u}_i}$ and $\mathrm{y}_i(t) \in \mathbb{R}^{\mathbf{d_y}_i}$ denote the respective entries of the input and output that agent i has information on. Without loss of generality, $u(t) = [\mathrm{u}_1(t), \mathrm{u}_2(t), \ldots, \mathrm{u}_N(t)]^\top \in \mathbb{R}^{\mathbf{d_u}}, y(t) = [\mathrm{y}_1(t), \mathrm{y}_2(t), \ldots, \mathrm{y}_N(t)]^\top \in \mathbb{R}^{\mathbf{d_y}}$. We make the following assumption on model (1):

Assumption 1. *The system* (1) *is BIBO stable, i.e. any bounded input $u(t)$ yields a bounded output $y(t)$. In addition, $\exists\ \bar{u} < \infty$ such that $u(t) \leq \bar{u}$ for all time $t \in \{1, 2, \ldots, T\}$.*

Assumption 2. *The function $g(u,\theta)$ is proper with respect to θ and is separable, i.e. $g(u,\theta) = \sum_{i=1}^N a_i\phi_i(u_i,\theta_i)$, for some $a_i > 0$ with $\theta = [\theta_1, \ldots, \theta_N]$, and the functions ϕ_i are convex and Lipschitz continuous with constant L_i.*

Under Assumption 2 and the network model, each agent i via ϕ_i captures how its regional controls u_i affect the output y by knowing the parameter θ_i. Therefore, the goal for each agent i is identifying θ_i by distributed communication and computations on locally available data u_i and y_i.

B. Distributed reformulation of the system modeling

In this section, we go through a series of reformulations of (1) for convenience of distributed algorithm design. With Assumption 2, we have the following formulation of $\mathbf{r}(\theta)$:

$$\mathbf{r}(\theta) = \tfrac{1}{2}\sum_{t=1}^T \left\| y(t) - \sum_{i=1}^N a_i\phi_i(\mathrm{u}_i(t),\theta_i)\right\|^2. \quad (3)$$

Define $\widehat{\mathbf{Y}}(t) = \mathrm{diag}\,(y(t)) = \mathrm{diag}\,([\mathrm{y}_1(t); \ldots; \mathrm{y}_N(t)]) \in \mathbb{R}^{\mathbf{d_y}\times\mathbf{d_y}}$. Because $\sum_{i=1}^N(\sum_{j\in S_i^y}[\widehat{\mathbf{Y}}(t)]_{:,j}) = y(t)$, we can rewrite (3) as

$$\mathbf{r}(\theta) = \tfrac{1}{2}\sum_{t=1}^T \left\| \sum_{i=1}^N \left(\sum_{j\in S_i^y}[\widehat{\mathbf{Y}}(t)]_{:,j} - a_i\phi_i(\mathrm{u}_i(t),\theta_i)\right)\right\|^2.$$

With the above set of reformulations, the system model identification problem is

$$\mathrm{argmin}_\theta\, \frac{1}{2}\sum_{t=1}^T \left\| \sum_{i=1}^N \left(a_i\phi_i(\mathrm{u}_i(t),\theta_i) - \sum_{j\in S_i^y}[\widehat{\mathbf{Y}}(t)]_{:,j}\right)\right\|^2. \quad (4)$$

The function $\mathbf{r}(\theta)$ couples the parameters and data for all the agents. We next consider a reformulation described in [9] to setup a formulation for the distributed algorithm, allowing the agents to do computation on locally available data and communicate with the neighboring agents in the network $\mathcal{G}(\mathcal{V},\mathcal{E})$ while recovering the solution of problem (4). Assuming the network $\mathcal{G}(\mathcal{V},\mathcal{E})$ is connected, we define $\mathbf{P} \in \mathbb{R}^{N\times N}$ as the Laplacian matrix associated with the graph. and consider the problem:

$$\mathrm{argmin}_{\theta, w\in\mathbb{R}^{N\mathbf{d_y}}}\, \frac{1}{2}\sum_{t=1}^T \left\| \Phi_a(\mathrm{u}(t),\theta) - \widehat{\mathbf{y}}(t) - \widehat{\mathbf{P}}w\right\|^2, \quad (5)$$

where $\Phi_a(\mathrm{u}(t),\theta) := [a_1\phi_1(\mathrm{u}_1(t),\theta_1); a_2\phi_2(\mathrm{u}_2(t),\theta_2); \ldots; a_N\phi_N(\mathrm{u}_N(t),\theta_N)] \in \mathbb{R}^{N\mathbf{d_y}}, \widehat{\mathbf{y}} := \big[\sum_{j\in S_1^y}[\widehat{\mathbf{Y}}(t)]_{:,j}; \sum_{j\in S_2^y}[\widehat{\mathbf{Y}}(t)]_{:,j}; \ldots; \sum_{j\in S_N^y}[\widehat{\mathbf{Y}}(t)]_{:,j}\big] \in \mathbb{R}^{N\mathbf{d_y}}, \widehat{\mathbf{P}} := \mathbf{P}\otimes\mathbf{I}_{\mathbf{d_y}}$. Define $\mathbf{x} = [\theta; \widehat{\mathbf{P}}w]$, $F(\mathbf{x}) = \sum_{t=1}^T f_t(\mathbf{x})$ with $f_t(\mathbf{x}) = \frac{1}{2}\left\|\Phi_a(\mathrm{u}(t),\theta) - \widehat{\mathbf{y}}(t) - \widehat{\mathbf{P}}w\right\|^2$ for convenience. Lemma 1 shows that the optimal solution of (5) is also a solution for (4).

Lemma 1. ***(Optimal solutions of*** (4) ***and*** (5)***).** If $\mathbf{x}^* = (\theta^*, \widehat{\mathbf{P}}w^*)$ is an optimal solution of problem* (5)*, then θ^* is also an optimal solution of problem* (4).

Proof. Using the first order optimality conditions for the convex function F (sum of composition of convex and increasing functions), we have $\nabla_\theta F(\mathbf{x}^*) = 0,\ \nabla_w F(\mathbf{x}^*) = 0$. Namely, for all $t \in \{1, 2, \ldots, T\}$,

$$\nabla_\theta\Phi_a(\mathrm{u}(t),\theta^*)^\top(\Phi_a(\mathrm{u}(t),\theta^*) - \widehat{\mathbf{y}}(t) - \widehat{\mathbf{P}}w^*) = 0, \quad (6)$$

$$\widehat{\mathbf{P}}^\top(\Phi_a(\mathrm{u}(t),\theta^*) - \widehat{\mathbf{y}}(t) - \widehat{\mathbf{P}}w^*) = 0. \quad (7)$$

From the property of the Laplacian matrix, $\mathrm{null}\{\mathbf{P}\} = \mathrm{span}\{1_N\}$, and (7), there exists z^* such that

$$1_N \otimes z^* := (\Phi_a(\mathrm{u}(t),\theta^*) - \widehat{\mathbf{y}}(t) - \widehat{\mathbf{P}}w^*) \quad (8)$$

Multiplying both by $1_N^\top \otimes \mathbf{I}_{\mathbf{d_y}}$ we get,

$$\begin{aligned} Nz^* &= (1_N^\top \otimes \mathbf{I}_{\mathbf{d_y}})1_N \otimes z^* \\ &= (1_N^\top \otimes \mathbf{I}_{\mathbf{d_y}})(\Phi_a(\mathrm{u}(t),\theta^*) - \widehat{\mathbf{y}}(t) - \widehat{\mathbf{P}}w^*) \\ &= \textstyle\sum_{i=1}^N \left(a_i\phi_i(\mathrm{u}_i(t),\theta_i^*) - \sum_{j\in S_i^y}[\widehat{\mathbf{Y}}(t)]_{:,j}\right) \\ &\quad - (1_N^\top \otimes \mathbf{I}_{\mathbf{d_y}})(\mathbf{P}\otimes\mathbf{I}_{\mathbf{d_y}})w^* \\ &= \textstyle\sum_{i=1}^N \left(a_i\phi_i(\mathrm{u}_i(t),\theta_i^*) - \sum_{j\in S_i^y}[\widehat{\mathbf{Y}}(t)]_{:,j}\right) \\ &\quad - (1_N^\top \otimes \mathbf{P})(\mathbf{I}_{\mathbf{d_y}}\otimes\mathbf{I}_{\mathbf{d_y}})w^* \\ &= \textstyle\sum_{i=1}^N \left(a_i\phi_i(\mathrm{u}_i(t),\theta_i^*) - \sum_{j\in S_i^y}[\widehat{\mathbf{Y}}(t)]_{:,j}\right). \end{aligned}$$

Thus,

$$z^* = \frac{1}{N}\Big(\sum_{i=1}^N \left(a_i\phi_i(\mathrm{u}_i(t),\theta_i^*) - \sum_{j\in S_i^y}[\widehat{\mathbf{Y}}(t)]_{:,j}\right)\Big) \quad (9)$$

Substituting (8) and (9) in (6), we have

$$a_j\nabla_{\theta_j}\phi_a(\mathrm{u}_j(t),\theta_j^*)^\top[\Big(\sum_{i=1}^N\big(a_i\phi_i(\mathrm{u}_i(t),\theta_i^\star) - \sum_{j\in S_i^y}[\widehat{\mathbf{Y}}(t)]_{:,j}\big)\Big] = 0,\ \forall j\in\{1,2,\ldots,N\}. \quad (10)$$

Because problem (4) is convex, by the optimality conditions, any $\theta^{'}$ is a solution of (4) if and only if, for all $t\in\{1,\ldots,T\}$,

$$a_j\nabla_{\theta_j}\phi_j(\mathrm{u}_j(t),\theta_j^{'})^\top\Big(\sum_{i=1}^N\big(a_i\phi_i(\mathrm{u}_i(t),\theta_i^{'}) - \sum_{j\in S_i^y}[\widehat{\mathbf{Y}}(t)]_{:,j}\big)\Big) = 0,\ \forall j\in\{1,2,\ldots,N\}.$$

Thus, we conclude θ^* is a solution of (4). □

Lemma 1 serves as an essential step for distributed algorithm design because it enables us to focus on solving (5) instead of (4) with distributed component $\mathbf{P}$. Following this, we introduce Lemma 2, which establishes the boundedness of the gradient steps involved in addressing (5).

Lemma 2. *Let Assumptions 1-2 hold. There exists constants $\eta_t>0$ and $\Delta<\infty$ such that $\|\nabla f_t\| := \left\|\begin{bmatrix}\nabla_\theta f_t\\ \nabla_w f_t\end{bmatrix}\right\| \le \Delta$ for all $t\in\{1,\ldots,T\}$ with $\mathbf{x}(t)$ updated by*

$$\mathbf{x}(t+1) = \mathbf{x}(t) - \eta_t\nabla f_t(\mathbf{x}(t)). \quad (11)$$

Proof. We start by presenting three supporting claims that we later utilize to prove the desired result.

Claim 1: Any γ sub-level set $C_\gamma := \{\mathbf{x}\mid f_t(\mathbf{x})\le\gamma\}$ of f_t is bounded.

Proof. Given $\beta>0$, let $v^*\in\operatorname{argmin}_{\mathbf{x}} f_t(\mathbf{x})$, with $\|v^*\|<\infty$. Define, $\Gamma_\beta := \{\mathbf{x}\mid \|\mathbf{x}-v^*\|=\beta\}$ and $v_\beta = \inf_{\mathbf{x}\in\Gamma_\beta} f_t(\mathbf{x})$. Note that Γ_β is non-empty and compact. Since, f_t is continuous, from the Weierstrass's theorem v_β is attained at some point of Γ_β, we have $v_\beta > f_t(v^*)$. For any $\mathbf{x}$ such that $\|\mathbf{x}-v^*\|>\beta$, let $\alpha=\frac{\beta}{\|\mathbf{x}-v^*\|}$, $\tilde{\mathbf{x}} = (1-\alpha)v^*+\alpha\mathbf{x}$. By convexity of f_t, we have

$$(1-\alpha)f_t(v^*)+\alpha f_t(\mathbf{x}) \ge f_t(\tilde{\mathbf{x}}).$$

Since $\|\tilde{\mathbf{x}}-v^*\| = \alpha\|\mathbf{x}-v^*\| = \beta$, $\tilde{\mathbf{x}}\in\Gamma_\beta$ and

$$f_t(\tilde{\mathbf{x}}) \ge v_\beta = \inf_{\mathbf{x}\in\Gamma_\beta} f_t(\mathbf{x}).$$

Combining the above two relations, we get

$$\begin{aligned} f_t(\mathbf{x}) &\ge \tfrac{f_t(\tilde{\mathbf{x}})-f_t(v^*)}{\alpha} + f_t(v^*) \ge f_t(v^*) + \tfrac{v_\beta - f_t(v^*)}{\alpha}\\ &= f_t(v^*) + \tfrac{v_\beta-f_t(v^*)}{\beta}\|\mathbf{x}-v^*\|. \end{aligned}$$

Because $v_\beta > f_t(v^*)$ and $f_t(\mathbf{x})\le\gamma$, we derive

$$\|\mathbf{x}-v^*\| \le \tfrac{\beta(\gamma-f_t(v^*))}{v_\beta-f_t(v^*)}.$$

Thus, $\|\mathbf{x}-v^*\|\le\max\left\{\beta, \frac{\beta(\gamma-f_t(v^*))}{v_\beta-f_t(v^*)}\right\}$. □

Claim 2: There exists a sufficiently small $\eta_t>0$ such that for all $t\in\{1,2,\ldots,T\}$, $\mathbf{x}(t+1)$ updated by (11) lies in the sub-level set $C_{f_t(\mathbf{x}(t))} := \{\mathbf{x}\mid f_t(\mathbf{x})\le f_t(\mathbf{x}(t))\}$.

Proof. By Taylor series expansion and $f_t\ge 0$,

$$\begin{aligned} &f_t(\mathbf{x}(t+1)) = f_t\big(\mathbf{x}(t)-\eta_t\nabla f_t(\mathbf{x}(t))\big)\\ &= f_t(\mathbf{x}(t)) - \eta_t\|\nabla f_t(\mathbf{x}(t))\|^2 + o(\eta_t\nabla f_t(\mathbf{x}(t)))\\ &= f_t(\mathbf{x}(t)) - \eta_t\Big(\|\nabla f_t(\mathbf{x}(t))\|^2 + \tfrac{o(\eta_t\nabla f_t(\mathbf{x}(t)))}{\eta_t}\Big) \le f_t(\mathbf{x}(t)), \end{aligned}$$

for sufficiently small $\eta_t>0$ by the definition of $o(\eta_t)$, which completes the proof. □

Claim 3: Let $\mathbf{z}(t) := \Phi_a(u(t),\theta(t)) - \widehat{\mathbf{y}}(t) - \widehat{\mathbf{P}}w(t)$. Then, $\exists\ \bar{\mathbf{z}}<\infty$ such that $\|\mathbf{z}(t)\|\le\bar{\mathbf{z}}$ for all $t\in\{1,2,\ldots,T\}$.

Proof. Under Assumptions 1 and 2,

$$\begin{aligned} \|\mathbf{z}(t)-\mathbf{z}(1)\| &= \|\Phi_a(\mathrm{u}(t),\theta(t)) - \widehat{\mathbf{y}}(t) - \widehat{\mathbf{P}}w(t) - \Phi_a(\mathrm{u}(1),\theta(1)) + \widehat{\mathbf{y}}(1) + \widehat{\mathbf{P}}w(1)\|\\ &\le \|\big(\Phi_a(\mathrm{u}(t),\theta(t)) - \widehat{\mathbf{P}}w(t)\big) - \big(\Phi_a(\mathrm{u}(1),\theta(1)) - \widehat{\mathbf{P}}w(1)\big)\| + \|\widehat{\mathbf{y}}(t)-\widehat{\mathbf{y}}(1)\|\\ &= L_m\left\|\begin{bmatrix}\mathrm{u}(t)\\ \mathbf{x}(t)\end{bmatrix} - \begin{bmatrix}\mathrm{u}(1)\\ \mathbf{x}(1)\end{bmatrix}\right\| + \|\widehat{\mathbf{y}}(t)-\widehat{\mathbf{y}}(1)\|. \end{aligned}$$

Therefore, $\|\mathbf{z}(t)-\mathbf{z}(1)\|\le 2\overline{\mathrm{y}} + 2L_m(\overline{\mathrm{u}}+D_m)$, where the results of Claims 1 and 2 are applied with $D_m := \max_t Diam_{\|.\|}(C_{f_t(\mathbf{x}(t))})$ and $L_m := \max_{1\le i\le N} L_i$. Therefore, there exists a $\bar{\mathbf{z}}<\infty$ that bounds $\|\mathbf{z}(t)\|$ for all $t\in\{1,2,\ldots,T\}$. □

With all the claims, we circle back to the proof of Lemma 2. At any time index t,

$$\nabla f_t = \begin{bmatrix}\nabla_\theta f_t\\ \nabla_w f_t\end{bmatrix} = \begin{bmatrix}\nabla_\theta\Phi_a(\mathrm{u}(t),\theta(t))^\top\mathbf{z}(t)\\ -\widehat{\mathbf{P}}^\top\mathbf{z}(t)\end{bmatrix}.$$

Thus, $\|\nabla f_t\|^2 \le \|\nabla_\theta\Phi_a(\mathrm{u}(t),\theta(t))^\top\mathbf{z}(t)\|^2 + \|\widehat{\mathbf{P}}^\top\mathbf{z}(t)\|^2 + 2\|\nabla_\theta\Phi_a(\mathrm{u}(t),\theta(t))^\top\mathbf{z}(t)\|\|\widehat{\mathbf{P}}^\top\mathbf{z}(t)\| \le (NL_m + \|\widehat{\mathbf{P}}\|)^2\bar{\mathbf{z}}^2$, where we used Assumption 2 and claim 3. Hence, $\|\nabla f_t\|\le (NL_m+\|\widehat{\mathbf{P}}\|)\bar{\mathbf{z}} := \Delta$. This completes the proof. □

III. Distributed Model identification

In this section, we develop an online algorithm designed to address the system model identification problem as formulated in (5). Subsequently, we elucidate the methodology for implementing this online algorithm within a distributed framework.

A. Model identification via online experiments

We begin the section by assuming that the input-output data for the system for every time instant in a sequential manner are available. In such a setting, the experimental data appears as an infinite sequence $(u(1),y(1)),(u(2),y(2)),(u(3),y(3)),\ldots$. At any time t after $(u(t),y(t))$ is obtained, the error function $f_t(\mathbf{x})$ is presented. As the information is received sequentially and all information is not available at once, we devise an algorithm that updates the model with every new measurement pair $(u(t),y(t))$. For this endeavor, we aim to minimize our "regret" with respect to a model that is devised using all the input-output pairs in hindsight. Let $\{\mathbf{x}(t)\}_{t\ge1}$ denote the model parameters generated by our algorithm, we formally define regret of our algorithm after any time T as,

$$\mathcal{R}_T := \textstyle\sum_{t=1}^T f_t(\mathbf{x}(t)) - \min_{\mathbf{x}}\sum_{t=1}^T f_t(\mathbf{x}). \quad (12)$$

Note that if $\mathcal{R}_T$ is zero, then the solution sequence $\{\mathbf{x}(t)\}_{t\ge1}$ is such that the total error incurred is equal to the error obtained by minimizing the error objective function in (5) created by using the entire input-output data. We propose the following online algorithm:

Algorithm 1: Online model identification

Input:
$\{\eta\}_{t\geq 1}$
Initialize:
$\mathbf{x}(1) \in \mathbb{R}^{\mathbf{d}_\theta + N\mathbf{d}_\mathbf{y}}$
For $t = 1, 2, \ldots$
- $\mathbf{x}(t+1) = \mathbf{x}(t) - \eta_t \nabla f_t(\mathbf{x}(t))$

Theorem 1. ***(Regret of Algorithm 1).*** *Let Assumption 1 holds. Let* $\mathbf{x}^* \in \operatorname{argmin}_\mathbf{x} \sum_{t=1}^T f_t(\mathbf{x})$ *and* $\eta_t = \frac{c_1}{\sqrt{t}}$, $c_1 > 0$. *Then, the regret of Algorithm 1 after any time* T *is bounded*

$$\mathcal{R}_T = \frac{\delta_1\sqrt{T}}{2} - \frac{\delta_2}{2} = O(\sqrt{T}),$$

where δ_1 *and* δ_2 *are some positive finite constants. Therefore,* $\limsup_{T\to\infty} \mathcal{R}_T/T \to 0$.

Proof. Let $\mathbf{x}^* \in \operatorname{argmin}_\mathbf{x} \sum_{t=1}^T f_t(\mathbf{x})$. Consider the update in Algorithm 1, $\mathbf{x}(t+1) - \mathbf{x}^* = \mathbf{x}(t) - \eta_t \nabla f_t(\mathbf{x}(t)) - \mathbf{x}^*$, then

$$\begin{aligned}\|\mathbf{x}(t+1) - \mathbf{x}^*\|^2 \leq \|\mathbf{x}(t) - \mathbf{x}^*\|^2 + \eta_t^2 \|\nabla f_t(\mathbf{x}(t))\|^2 \\ - 2\eta_t \nabla f_t(\mathbf{x}(t))^\top (\mathbf{x}(t) - \mathbf{x}^*).\end{aligned}$$

From Lemma 2, there exists $\Delta =: \sup_t \|\nabla f_t(\mathbf{x}(t))\| < \infty$,

$$\begin{aligned}&\|\mathbf{x}(t+1) - \mathbf{x}^*\|^2 \leq \|\mathbf{x}(t) - \mathbf{x}^*\|^2 + \eta_t^2 \Delta^2 \\ &\quad - 2\eta_t \nabla f_t(\mathbf{x}(t))^\top (\mathbf{x}(t) - \mathbf{x}^*) \\ &\implies \nabla f_t(\mathbf{x}(t))^\top (\mathbf{x}(t) - \mathbf{x}^*) \\ &\qquad \leq \frac{\|\mathbf{x}(t) - \mathbf{x}^*\|^2 - \|\mathbf{x}(t+1) - \mathbf{x}^*\|^2}{2\eta_t} + \frac{\eta_t \Delta^2}{2}\end{aligned} \tag{13}$$

By convexity of f_t,

$$f_t(\mathbf{x}(t)) - f_t(\mathbf{x}^*) \leq \nabla f_t(\mathbf{x}(t))^\top (\mathbf{x}(t) - \mathbf{x}^*). \tag{14}$$

Combining (13) and (14) gives

$$f_t(\mathbf{x}(t)) - f_t(\mathbf{x}^*) \leq \frac{\|\mathbf{x}(t) - \mathbf{x}^*\|^2 - \|\mathbf{x}(t+1) - \mathbf{x}^*\|^2}{2\eta_t} + \frac{\eta_t \Delta^2}{2}.$$

Summing over $t = 1$ to T,

$$\begin{aligned}\mathcal{R}_T &= \sum_{t=1}^T f_t(\mathbf{x}(t)) - \sum_{t=1}^T f_t(\mathbf{x}^*) \\ &\leq \sum_{t=1}^T \left(\frac{\|\mathbf{x}(t) - \mathbf{x}^*\|^2 - \|\mathbf{x}(t+1) - \mathbf{x}^*\|^2}{2\eta_t} \right) + \sum_{t=1}^T \frac{\eta_t \Delta^2}{2} \\ &= \frac{\|\mathbf{x}(1) - \mathbf{x}^*\|^2}{2\eta_1} - \frac{\|\mathbf{x}(T+1) - \mathbf{x}^*\|^2}{2\eta_T} + \frac{\Delta^2}{2} \sum_{t=1}^T \eta_t \\ &\quad + \frac{1}{2} \sum_{t=2}^T \|\mathbf{x}(t) - \mathbf{x}^*\|^2 \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} \right).\end{aligned}$$

From claims 1 and 2 in Lemma 2, there exists $\Xi < \infty$ such that $\sup_t \|\mathbf{x}(t) - \mathbf{x}^*\| \leq \sup_t \|\mathbf{x}(t)\| + \|\mathbf{x}^*\| \leq \Xi$. Therefore,

$$\begin{aligned}\mathcal{R}_T &\leq \tfrac{\Xi^2}{2} \left(\tfrac{1}{\eta_1} + \textstyle\sum_{t=2}^T \left(\tfrac{1}{\eta_t} - \tfrac{1}{\eta_{t-1}} \right) \right) + \tfrac{\Delta^2}{2} \textstyle\sum_{t=1}^T \eta_t \\ &= \tfrac{\Xi^2}{2\eta_T} + \tfrac{\Delta^2}{2} \textstyle\sum_{t=1}^T \eta_t.\end{aligned}$$

For $\eta_t = \frac{c_1}{\sqrt{t}}$, $\sum_{t=1}^T \eta_t = \sum_{t=1}^T \frac{c_1}{\sqrt{t}} \leq 1 + \int_{t=1}^T \frac{c_1}{\sqrt{t}} dt \leq 1 + [2c_1\sqrt{t}]_1^T \leq 2c_1\sqrt{T} + 1 - 2c_1$. Thus,

$$\mathcal{R}_T \leq \frac{(\Xi^2/c_1 + 2\Delta^2 c_1)\sqrt{T}}{2} - \frac{(2c_1 - 1)\Delta^2}{2}.$$

Therefore, $\limsup_{T\to\infty} \mathcal{R}_T/T \to 0$. □

The result of Theorem 1 establishes that on average Algorithm 1 provides a model close to the best-fixed model in hindsight and hence able to solve the model identification problem despite all the information not present at the start of the algorithm.

B. Distributed implementation

In the previous section, we described Algorithm 1 to solve the model identification problem under online (sequential) experimental scenarios. Here, we present how Algorithm 1 can be implemented (and synthesized) in a distributed manner. The updates in Algorithm 1 utilizes $\nabla f_t(\mathbf{x})$. From (6) and (7), we have

$$\nabla f_t(\mathbf{x}) = \begin{bmatrix} \nabla_\theta \Phi_a(\mathrm{u}(t), \theta(t))^\top \mathbf{z} \\ -\widehat{\mathbf{P}}^\top \mathbf{z} \end{bmatrix}. \tag{15}$$

Note that $\mathbf{z}$ can be decomposed as, $\mathbf{z} = [\mathbf{z}_1; \mathbf{z}_2; \ldots; \mathbf{z}_N]$, where

$$\mathbf{z}_i(t) = a_i \phi_i(\mathrm{u}_i(t), \theta_i) - \sum_{j \in S_i^y} [\widehat{\mathbf{Y}}(t)]_{:,j} - (\widehat{\mathbf{P}}_{ii} w_i + \sum_{j \in \mathcal{N}_i} \widehat{\mathbf{P}}_{ji} w_j).$$

A closer examination of (15) yields that ∇f_t can be further written as, $\nabla f_t = ([(\nabla_1 f_t)^\top; (\nabla_2 f_t)^\top; \ldots; (\nabla_N f_t)^\top])$, where

$$\nabla_i f_t = \begin{bmatrix} a_i \nabla_{\theta_i} \phi_a(\mathrm{u}_i(t), \theta_i)^\top \mathbf{z}_i(t) \\ -\widehat{\mathbf{P}}_{ii} \mathbf{z}_i(t) - \sum_{j \in \mathcal{N}_i} \widehat{\mathbf{P}}_{ji} \mathbf{z}_j(t) \end{bmatrix}, \tag{16}$$

for all $i \in \{1, 2, \ldots, N\}$. Thus, using (16) the updates in Algorithm 1 can be implemented in a distributed manner at any agent i while maintaining an auxiliary variable $\mathbf{z}_i$ as shown in Algorithm 2.

Algorithm 2: Distributed online model identification

Input:
$\{\eta\}_{t\geq 1}$
Initialize:
$\theta_i(1) \in \mathbb{R}^{\theta_i}, w_i(1) \in \mathbb{R}^{\mathbf{d}_\mathbf{y}}$ for all $i \in \{1, 2, \ldots, N\}$
For $t = 1, 2, \ldots$
For all $i \in \{1, 2, \ldots, N\}$
$\mathbf{z}_i(t) = a_i \phi_i(\mathrm{u}_i(t), \theta_i(t)) - \sum_{j \in S_i^y} [\widehat{\mathbf{Y}}(t)]_{:,j}$
$\quad -(\widehat{\mathbf{P}}_{ii} w_i(t) + \sum_{j \in \mathcal{N}_i} \widehat{\mathbf{P}}_{ij} w_j(t))$
$\theta_i(t+1) = \theta_i(t)$
$\quad - \eta_t (a_i \nabla_{\theta_i} \phi_i(\mathrm{u}_i(t), \theta_i(t))^\top \mathbf{z}_i(t))$
$w_i(t+1) = w_i(t)$
$\quad + \eta_t (\widehat{\mathbf{P}}_{ii} \mathbf{z}_i(t) + \sum_{j \in \mathcal{N}_i} \widehat{\mathbf{P}}_{ji} \mathbf{z}_j(t))$

In Algorithm 2, each agent i engages in two rounds of communication on auxiliary variables $\mathbf{z}_i$ and w_i. Importantly, the exchange of $\mathbf{z}_i$ and w_i among agents does not allow for the reconstruction of the model parameters θ_i or the local input-output pairs. As a result, the data transmitted across the communication network does not divulge any direct details regarding the system's parameters or local data, thereby bolstering the privacy and security of the systems.

IV. Numerical Simulations

In this section, we apply Algorithm 2 to identify the power flow model, which may be non-linear, for a modified IEEE 37 bus system. This identified model is then utilized within a feedback-based distributed algorithm to regulate voltage in the presence of photovoltaic energy sources (PES), as discussed in [10]. The simulation setup for the IEEE 37 bus system follows the parameters outlined in [7], and we refrain from repeating the system description for brevity.

During the simulation, PES agents exchange their estimates through an interconnected communication network, represented as a graph $\mathcal{G}$. This network's graph Laplacian is used as the weight matrix $\mathbf{P}$ in our algorithm. We explore two distinct model formats for identification:

1) A linear model where nodal active and reactive power injections ($u_i \in \mathbb{R}^2$) serve as inputs and voltage measurements ($y_i \in \mathbb{R}$) as outputs, described by $y(t) = Au(t)$ for all t. This model is known as the LinDistFlow model, and the goal is to identify matrix A.
2) A non-linear model that posits a polynomial relationship between local power injections and bus voltage, with $u_i \in \mathbb{R}$ representing voltage magnitude and $y_i \in \mathbb{R}^2$ encapsulating nodal active and reactive power. The constant power load model is expressed as

$$y_i(t) = A_i u_i^2(t) + B_i u_i(t) + C_i \tag{17}$$

for all $i \in \{1, \cdots, N\}$, where $A_i, B_i, C_i \in \mathbb{R}^{2\times 1}$. This reflects the quadratic correlation between power injection and voltage magnitude observed in power flow equations. Note that (17) satisfies Assumptions 1 and 2 for implementing Algorithm 2.

For Algorithm 2, we adopt a step-size $\eta_t = 0.01/\sqrt{t}$. Figure 1 showcases the voltage levels at PES buses throughout the online identification and distributed control process, updating the model in real-time as new input-output data becomes available. Figure 1(a) illustrates potential violations of voltage regulation limits without control measures. Figure 1(b) depicts the outcomes using distributed control based on the LinDistFlow model. The real-time estimated linear model closely approximates the LinDistFlow model, reflected a satisfactory control performance in Figure 1(c), albeit with some fluctuations during periods of reduced PES generation. Notably, the control performance using the identified non-linear model, as shown in Figure 1(d), surpasses that of the linear model, which aligns with expectations given the non-linear model's closer representation of actual power flow dynamics. Overall, the models identified through our proposed algorithm demonstrate effective voltage regulation capabilities.

V. Conclusion

In this paper, we developed an online distributed algorithm where each agent updates its estimate of the model via an online gradient descent scheme utilizing the most recent input-output pair. We prove that the developed distributed algorithm has a sub-linear regret and determines the original system model. The real-time updates of the agents, utilizing sequential data, significantly reduce the communication network's bandwidth requirements. Further, agents only share non-linear

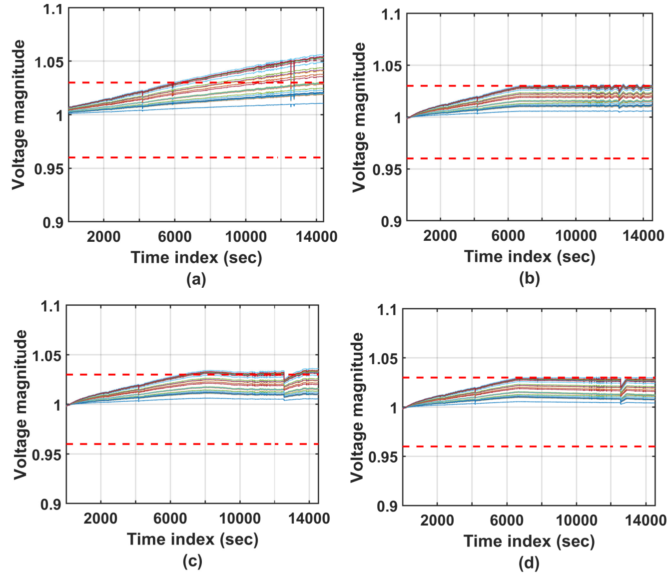

Fig. 1: The voltage magnitudes ($p.u.$) over time (a) without any control (b) with distributed feedback-based control and the LinDistFlow, [11] model (c) with online distributed estimated linear power flow model and feedback-based control (d) with online distributed estimated non-linear power flow model and feedback-based control.

estimates preserving their private information. The numerical simulation study corroborates the efficacy of our developed algorithm with the identification of a more accurate quadratic power flow model, which improves the voltage regulation performance of the control system.

References

[1] Y. Liao, Y. Weng, G. Liu, and R. Rajagopal, "Urban MV and LV distribution grid topology estimation via group lasso," *IEEE Transactions on Power Systems*, vol. 34, no. 1, pp. 12–27, 2018.

[2] O. Ardakanian, V. W. Wong, R. Dobbe, S. H. Low, A. von Meier, C. J. Tomlin, and Y. Yuan, "On identification of distribution grids," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 3, pp. 950–960, 2019.

[3] J. Yu, Y. Weng, and R. Rajagopal, "PaToPa: A data-driven parameter and topology joint estimation framework in distribution grids," *IEEE Transactions on Power Systems*, vol. 33, no. 4, pp. 4335–4347, 2017.

[4] J. Zhang, P. Wang, and N. Zhang, "Distribution network admittance matrix estimation with linear regression," *IEEE Transactions on Power Systems*, vol. 36, no. 5, pp. 4896–4899, 2021.

[5] S. D. McArthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, and T. Funabashi, "Multi-agent systems for power engineering applications—part i: Concepts, approaches, and technical challenges," *IEEE Transactions on Power systems*, vol. 22, no. 4, pp. 1743–1752, 2007.

[6] O. P. Mahela, M. Khosravy, N. Gupta, B. Khan, H. H. Alhelou, R. Mahla, N. Patel, and P. Siano, "Comprehensive overview of multi-agent systems for controlling smart grids," *CSEE Journal of Power and Energy Systems*, vol. 8, no. 1, pp. 115–131, 2020.

[7] C.-Y. Chang, "A privacy preserving distributed model identification algorithm for power distribution systems," in *62nd IEEE Conference on Decision and Control*, 2023.

[8] D. E. Knuth, *The Art of Computer Programming, Volume 1: Fundamental Algorithms*. Addison-Wesley, 1997.

[9] Y. Huang, Z. Meng, and J. Sun, "Scalable distributed least square algorithms for large-scale linear equations via an optimization approach," *Automatica*, vol. 146, p. 110572, 2022.

[10] C.-Y. Chang, M. Colombino, J. Cortés, and E. Dall'Anese, "Saddle-flow dynamics for distributed feedback-based optimization," *IEEE Control Systems Letters*, 2019.

[11] M. E. Baran and F. F. Wu, "Network reconfiguration in distribution systems for loss reduction and load balancing," *IEEE Transactions on Power delivery*, vol. 4, no. 2, pp. 1401–1407, 1989.