一种适用于配电系统的隐私保护分布式模型辨识算法

Chin-Yao Chang

arXiv:2304.03172v1 [eess.SY] 6 Apr 2023

摘要—**分布式控制/优化是网络系统中一种很有前景的方法，相较于集中式方案，它具有鲁棒性强、成本效益高且隐私保护更佳等优势。然而，分布式方法也存在缺点，例如，由于对整体网络模型的了解有限，收敛速度较慢。此外，在通信过程中确保敏感信息的隐私，也可能带来实施方面的挑战。为解决这一问题，我们提出了一种分布式模型辨识算法，使每个代理都能够辨识刻画其局部控制与整体系统输出之间关系的子模型。该算法仅通过虚拟变量进行通信，从而保护局部代理的隐私。我们将该算法应用于改进的 IEEE 配电系统的电压调节，以展示其在配电系统中的有效性。该算法非常适用于配电控制的需求，可有效解决网络系统中分布式模型辨识所面临的挑战。**

I. 引言

近年来，强调用户隐私的趋势日益明显。这促使人们采用更严格的法规和政策来保护用户隐私，例如欧洲的《通用数据保护条例》（GDPR） [1]以及美国的《加利福尼亚州消费者隐私法》（CCPA） [2] 。随着强调隐私的趋势可能会持续，如何在不牺牲隐私的前提下从数据中提取价值，已成为医疗保健、保险和金融科技等许多应用领域的重要议题。这项技术被称为联邦学习，已成功应用于语言模型、图像识别等领域 [3], [4]。近期的一些研究 [5], [6] 还发现，联邦学习有助于为未来的电网制定能源管理策略，而这些电网可能涉及数百万个可控设备 [7]。然而，这些研究仅限于了解某些类别的电力消耗或发电模式，无法帮助识别系统级模型（例如潮流模型），而这对于未来配电系统的精细化控制至关重要。

由于安装所有传感器不具成本效益，配电系统大多不可观测。因此，许多现有研究致力于识别对电网控制有价值的信息，例如配电系统的拓扑结构 [8], [9] 和状态估计 [10]–[12]。尽管配电系统状态估计（DSSE）旨在估算未安装传感器的母线电压和电流，但许多 DSSE 仍需要了解导纳矩阵或类似信息 [11], [12]。导纳信息对于配电系统的各种控制也很有价值，例如 [13], [14],，但这类信息并不总是容易获取，尤其是考虑到隐私问题带来的数据收集障碍。

上述研究假设由一个中心实体收集配电系统中所有传感器的数据以进行估计，导纳矩阵的估计 [15]也是如此。收集所有可用数据，尤其是本地用户或分布式能源资源的电力消耗模式，可能颇具挑战性。一般而言，有一些分布式算法可以通过分布式通信 [16], [17]估计系统模型，例如导纳矩阵。然而，分布式算法在机制上包含共识环节，通常涉及估计模型甚至代理的局部状态变量的共识，这会使有价值的信息留存于通信网络中，从而容易受到网络攻击。许多分布式算法的另一个缺点是，它们本质上属于梯度下降算法，因此能够耐受数据包丢失或延迟，但收敛速率会有所牺牲。Adam算法（自适应矩估计） [18]提供了精细的步长（学习率）选择机制，有助于提高分布式算法的收敛速率。研究还发现，Adam算法在大规模优化 [19], [20], 方面非常有效，因此逐渐成为机器学习领域的主流算法之一 [21]。在分布式算法中引入Adam算法的要素，有望提高收敛速率。

贡献： 该论文的贡献主要在于开发了一种分布式模型辨识算法，使网络系统中的每个代理都能够辨识其局部控制与整体系统输出之间相关性的子模型；如果输入是功率注入量、输出是电压幅值，则该子模型可以理解为配电系统LinDistFlow模型（或其等价形式）的子矩阵。所提出的算法

C.-Y. Chang就职于美国国家可再生能源实验室，戈尔登，CO80401，美国（电子邮件： {chinyao.chang}@nrel.gov）。该 s wor k 曾就职于hored in 部分 by NREL，运行 d by Alliance f or Sustainable Energy, LLC 受美国能源部（DOE）委托，根据合同编号 DE-AC36-08GO28308 开展工作。资金由 DOE 电力办公室的先进电网建模计划通过协议 NO.33652 提供。本文所表达的观点不一定代表 DOE 或美国政府的观点。美国政府保留相关权利；出版商接受本文发表，即承认美国政府保留一项非排他性、已付清、不可撤销、全球范围的许可，可为美国政府目的发表或复制本作品的已发表版本，或授权他人这样做。

采用Adam算法的要素来提高收敛速率。此外，该算法具有一个颇具吸引力的特性：代理之间交换的数据包是虚拟变量，因此本地数据和辨识出的模型得以保持私密。我们通过一个经过修改的IEEE测试系统展示了所提出的分布式模型辨识算法的有效性。

II. 预备知识

A. 符号说明

对于 $x \in \mathbb{R}^n$，其 l^2 范数和关于矩阵 $A \succ 0$ 的二次范数分别记为 $\|x\|_2$ 和 $\|x\|_A$。对于满足 $A_i \in \mathbb{R}^{m\times n}$ 对所有 $i = 1, 2, \cdots, N$ 均成立的矩阵 A_1、A_2、$\cdots$、A_N，$\mathbf{blkdiag}(\{A_i\}_{i=1}^N)$ 表示由所有 A_i 构成的块对角矩阵；$[A_1、A_2、\cdots、A_n] \in \mathbb{R}^{Nm\times n}$ 和 $[A_1, A_2, \cdots, A_n] \in \mathbb{R}^{m\times Nn}$ 分别表示纵向和横向拼接。A_i 和 A_j 的 Hadamard 乘积和除法分别记为 $A_i \odot A_j$ 和 $A_i./A_j$。对于矩阵 $A \in \mathbb{R}^{m\times n}$，$vec(A) \in \mathbb{R}^{mn}$ 是将 A 的列向量从左到右拼接而成的 A 向量化列向量。矩阵 A 的核记为 $\mathbf{ker}(A)$。A 的第 i^{th} 行、第 k^{th} 列的标量元素记为 $A(i,k)$；$A(:,k)$ 是 A 的 k^{th} 列向量。维度为 n 的单位矩阵和全 $\mathbf{1}$ 向量分别记为 I_n 和 $\mathbf{1}_n$。

B. 基于输入-输出数据的系统建模

下面简要概述本文后续部分将使用的数据驱动建模框架。设 $u(k) \in \mathbb{R}^m$ 和 $y(k) \in \mathbb{R}^n$ 分别为未知系统在时刻 k 的输入和输出数据。给定在时刻 $k = 0, \cdots, T-1$采集的输入-输出数据，我们定义如下所示的数据矩阵：

$$U = [\phi_u(u(0)), \cdots, \phi_u(u(T-1))], \tag{1a}$$
$$Y = [\phi_y(y(0)), \cdots, \phi_y(y(T-1))], \tag{1b}$$

其中 $\phi_u : \mathbb{R}^m \mapsto \mathbb{R}^m$ 和 $\phi_y : \mathbb{R}^n \mapsto \mathbb{R}^n$ 是用于捕捉待识别系统已知（可能是非线性的）物理特性的映射。假设目标系统中唯一未知的部分是线性的，并由$A \in \mathbb{R}^{n\times m}$表征，则以下方程成立：

$$Y = AU. \tag{2}$$

给定数据（U 和 Y）的模型辨识问题可以理解为求解线性方程（2）。如果 U 具有满行秩，则系统模型 A 是唯一确定的。利用数据（2）对系统模型进行直接数据表示，有助于控制器设计；更多详细信息见 [22], [23]。

III. 分布式模型辨识算法

本节首先展示如何在网络系统中构建式（2），然后说明如何将其改写为分布式优化问题。接下来，我们将开发一种利用 Adam 自适应步长的分布式模型辨识算法。

A. 网络系统的系统建模

我们考虑一个网络系统，该系统被划分为 N 个区域，每个区域都配有一个代理，用于收集该区域内的所有执行器数据和传感器数据。我们假设，这种网络划分设置有效地使每个代理 $i \in \mathcal{N} := \{1, 2, \cdots, N\}$ 只能获取 U 和 Y 中某些行的信息，以便构建模型辨识问题 (2)。分别将 $\mathcal{D}_{u,i}$ 和 $\mathcal{D}_{y,i}$ 定义为代理 i 已知的 U 和 Y 的行集合。为简单起见，我们假设系统完全可观测，即对于所有 $i \neq j$，$\cup_{i\in\mathcal{N}} \mathcal{D}_{u,i} = \{1, 2, \cdots, m\}$ 和 $\cup_{i\in\mathcal{N}} \mathcal{D}_{y,i} = \{1, 2, \cdots, n\}$，其中 $\mathcal{D}_{u,i} \cap \mathcal{D}_{u,j} = \emptyset$ 和 $\mathcal{D}_{y,i} \cap \mathcal{D}_{y,j} = \emptyset$。在此设置下，代理 i 只需知道 A 的部分列，而非完整矩阵，即可掌握其区域控制如何影响输出 Y；我们将该子矩阵定义为 $A_{\mathcal{D}_{u,i}} \in \mathbb{R}^{n\times|\mathcal{D}_{u,i}|}$，且不失一般性地，$A = [A_{\mathcal{D}_{u,1}}, A_{\mathcal{D}_{u,2}}, \cdots, A_{\mathcal{D}_{u,N}}]$。请注意，尽管代理 i只需要 $A_{\mathcal{D}_{u,i}}$，但所有代理仍能够辨识完整模型 A 并就其达成共识；然而，这可能导致代理之间产生大量数据交换，从而使分布式通信网络不堪重负。因此，每个代理 i 的目标是通过分布式通信以及本地可用的 U 和 Y 子矩阵，辨识 $A_{\mathcal{D}_{u,i}}$。

B. 系统建模的分布式重构

在本节中，我们将对（2）进行一系列重构，以便于分布式算法设计。定义$\mathbf{x}_i = vec(A_{\mathcal{D}_{u,i}})$和$\mathbf{x} = [\mathbf{x}_1; \mathbf{x}_2; \cdots; \mathbf{x}_n] \in \mathbb{R}^{nm}$。在此重排中，我们考虑（2）的以下形式：

$$\min_{\mathbf{x}} \frac{1}{2} \sum_{k=1}^{T} \left\| \Big(\sum_{i=1}^{N} U_{k,\mathcal{D}_{u,i}} \cdot \mathbf{x}_i \Big) - Y(:,k) \right\|_2^2, \tag{3}$$

其中，$U_{k,\mathcal{D}_{u,i}} \in \mathbb{R}^{n\times n\cdot|\mathcal{D}_{u,i}|}$是对所有$i \in \mathcal{D}_{u,i}$而言的 $U(i,k) \cdot I_n$的水平拼接。通过直接代数运算可以验证，(3) 的最优解是(2)的一个解。采用优化形式的原因在于，实际上，由于含噪数据、通信干扰或其他干扰，很难找到满足 (2) 的 $\mathbf{x}$。接下来，我们以更紧凑的方式重写 (3)。定义 $U_{\mathcal{D}_{u,i}} = [U_{1,\mathcal{D}_{u,i}}; U_{2,\mathcal{D}_{u,i}}; \ldots; U_{T,\mathcal{D}_{u,i}}]$，$Y_{\mathcal{D}_{y,i}} = [Y_{1,\mathcal{D}_{y,i}}; Y_{2,\mathcal{D}_{y,i}}; \cdots; Y_{T,\mathcal{D}_{y,i}}]$，其中 $Y_{k,\mathcal{D}_{y,i}} \in \mathbb{R}^n$ 使得

$$Y_{k,\mathcal{D}_{y,i}}(j) = \begin{cases} Y(i,k) & \text{if } j \in \mathcal{D}_{y,i}, \\ 0 & \text{otherwise,} \end{cases}$$

将式（3）改写为

$$\min_{\mathbf{x}} f(\mathbf{x}), \quad f(\mathbf{x}) := \frac{1}{2} \left\| \sum_{i=1}^{N} (U_{\mathcal{D}_{u,i}} \mathbf{x}_i - Y_{\mathcal{D}_{y,i}}) \right\|_2^2. \tag{4}$$

目标函数 $f(\mathbf{x})$ 将所有代理的变量和数据耦合在一起，因此目前尚无法通过分布式通信求解。接下来，我们首先利用 [17] 中的结果，采用一种能够分布式求解（4）的算法，然后对其进行一些修改，加入 Adam 步长，并说明其隐私保护特性。定义图。

与分布式通信网络相关，表示为 $\mathcal{G}=(\mathcal{N},\mathcal{E})$，其中 $\mathcal{E}\subseteq\mathcal{N}\times\mathcal{N}$是边（通信链路）的集合。接下来考虑以下优化：

$$\min_{\mathbf{x},\mathbf{w}} \hat{f}(\mathbf{x},\mathbf{w}), \quad \hat{f}(\mathbf{x},\mathbf{w}) := \frac{1}{2}\left\|\hat{U}\,\mathbf{x}-\hat{Y}-\mathbf{L}^{\frac{1}{2}}\,\mathbf{w}\right\|_2^2, \tag{5}$$

其中，$\mathbf{w}\in\mathbb{R}^{TNn}$是新引入的松弛变量，$\hat{U}=\text{blkdiag}(\{\hat{U}_{\mathcal{D}_{u,i}}\}_{i=1}^{N})$，$\hat{Y}=[Y_{\mathcal{D}_{y,1}};\ Y_{\mathcal{D}_{y,2}},\cdots,Y_{\mathcal{D}_{y,N}}]$，$\mathbf{L}=(L\otimes I_{nT})$，$L$是与$\mathcal{G}$相关联的拉普拉斯矩阵，而$\mathbf{L}^{\frac{1}{2}}$是$\mathbf{L}$的平方根。引理 III.1（ [17, 引理 3.1]的简洁版本）通过 KKT 最优性条件论证表明，问题 (5) 的最优解也是问题 (4) 的最优解。

引理 III.1。（（4）和（5）的最优解）。若 $\mathcal{G}$ 是连通的，且（$\mathbf{x}^\star$，$\mathbf{w}^\star$）是（5）的一个最优解，则$\mathbf{x}^\star$是（4）的一个最优解。

证明。 根据 KKT 条件，($\mathbf{x}^\star$,$\mathbf{w}^\star$) 是（5）的最优解，当且仅当

$$\hat{U}^\top(\hat{U}\,\mathbf{x}^\star-\hat{Y}-\mathbf{L}^{\frac{1}{2}}\,\mathbf{w}^\star)=0, \tag{6a}$$

$$\mathbf{L}^{\frac{1}{2}\top}(\hat{U}\,\mathbf{x}^\star-\hat{Y}-\mathbf{L}^{\frac{1}{2}}\,\mathbf{w}^\star)=0. \tag{6b}$$

根据 (6b) 以及 $\ker(\mathbf{L}^{\frac{1}{2}})=\ker(\mathbf{L})$的性质，我们可以定义 $\mathbf{z}^\star=\mathbf{1}_N\otimes z^\star=\hat{U}\mathbf{x}^\star-\hat{Y}-\mathbf{L}^{\frac{1}{2}}\mathbf{w}^\star$。通过在左侧乘以 $\mathbf{z}^\star$ by $\mathbf{1}_N^\top\otimes I_{nT}$，我们得到以下方程：

$$Nz^\star=\sum_{i=1}^{N}(\hat{U}_{\mathcal{D}_{u,i}}\,\mathbf{x}_i^\star-\hat{Y}_{\mathcal{D}_{y,i}}), \tag{7}$$

其中，推导时再次用到了 $\ker(\mathbf{L}^{\frac{1}{2}})=\ker(\mathbf{L})$。将 (7). (7) 代入 (6a) 得

$$\hat{U}_{\mathcal{D}_{u,j}}^\top\Big(\sum_{i=1}^{N}(\hat{U}_{\mathcal{D}_{u,i}}\,\mathbf{x}_i^\star-\hat{Y}_{\mathcal{D}_{y,i}})\Big)=0,\qquad \forall j\in\mathcal{N}. \tag{8}$$

由于 (8) 是优化问题 (4) 的最优性 KTT 条件，我们得出结论：$\mathbf{x}^\star$ 也是 (4) 的一个最优解，至此证明完毕。□ □

利用引理 III.1，我们着重以分布式方式求解原始模型辨识问题 (5)。

C. 模型辨识的分布式算法

分布式算法求解 (5) 的第一步是考察 (5) 的梯度下降：

$$\dot{\mathbf{x}}=-\hat{U}^\top(\hat{U}\,\mathbf{x}-\hat{Y}-\mathbf{L}^{\frac{1}{2}}\,\mathbf{w}), \tag{9a}$$

$$\dot{\mathbf{w}}=\mathbf{L}^{\frac{1}{2}}(\hat{U}\,\mathbf{x}-\hat{Y}-\mathbf{L}^{\frac{1}{2}}\,\mathbf{w}). \tag{9b}$$

由于$\mathbf{L}^{\frac{1}{2}}$ 不一定是稀疏的，(9)无法直接以分布式方式实现。即使我们通过将$\mathbf{L}^{\frac{1}{2}}$替换为**L</style}来表述(5)（引理III. 1的结果仍然成立），在(9)中将L**$^{\frac{1}{2}}$替换为L </style}后得到的相关梯度下降仍然需要在两跳邻居之间交换数据包，并且本地输入输出数据应在各代理之间共享，这是不理想的。通过引入变量变换

$$\mathbf{z}=\hat{U}\,\mathbf{x}-\hat{Y}-\mathbf{L}^{\frac{1}{2}}\,\mathbf{w}, \tag{10}$$

相反，我们考虑以下替代梯度方法：

$$\dot{\mathbf{x}}=-\hat{U}^\top\,\mathbf{z}, \tag{11a}$$

$$\dot{\mathbf{z}}=-\,\mathbf{L}\,\mathbf{z}+\hat{U}\dot{\mathbf{x}}=-(\mathbf{L}+\hat{U}\hat{U}^\top)\,\mathbf{z}\,. \tag{11b}$$

由于 $\hat{U}$ 是块对角矩阵，且非对角元素为零，因此，实现式 (11)仅需要对$\mathbf{z}$ 进行分布式通信，其中$\mathbf{z}$ 应适当地分区，使得$\mathbf{z}=[\mathbf{z}_1;\ \mathbf{z}_2;\ \cdots;\ \mathbf{z}_N]$，$\mathbf{z}_i\in\mathbb{R}^{nT}$。 [17] 已表明，式(11) 及其欧拉离散形式 (12) 均收敛于式 (5) 的最优解。

$$\mathbf{x}(k+1)=\mathbf{x}(k)-\alpha\hat{U}^\top\,\mathbf{z}(k), \tag{12a}$$

$$\mathbf{z}(k+1)=\mathbf{z}(k)-\alpha\Big(\mathbf{L}+\hat{U}\hat{U}^\top\Big)\,\mathbf{z}(k). \tag{12b}$$

为提高式（12）的收敛速率，我们提出在式（12）中引入 Adam 自适应步长。将 $D=\mathbf{L}+\hat{U}\hat{U}^\top$和 N_I 定义为模型辨识的迭代次数，我们在算法1中提出了采用 Adam 自适应步长的分布式模型辨识算法。

算法1 分布式模型辨识算法与 Adam 自适应步长

要求： $\beta_1,\beta_2\in[0,1)$, $\epsilon>0$, $k=0$

要求： 初始化 $s_1(0)=s_2(0)=\hat{s}_1(0)=\hat{s}_2(0)=0$，并且满足（10）的$\mathbf{x}(0)$、$\mathbf{w}(0)$、$\mathbf{z}(0)$,

1: **当** $k\le N_I-1$ **时**
2: $k\leftarrow k+1$
3: $g(k)\leftarrow\hat{U}^\top\,\mathbf{z}(k-1)$
4: $s_1(k)\leftarrow\beta_1 s_1(k-1)+(1-\beta_1)g(k)$
5: $s_2(k)\leftarrow\beta_2 s_2(k-1)+(1-\beta_2)\Big(g(k)\odot g(k)\Big)$
6: $\hat{s}_1(k)\leftarrow s_1(k)/(1-\beta_1^k)$
7: $\hat{s}_2(k)\leftarrow s_2(k)/(1-\beta_2^k)$
8: $\mathbf{x}(k)\leftarrow\mathbf{x}(k-1)-\alpha\Big(\hat{s}_1(k)./(\sqrt{\hat{s}_2(k)}+\epsilon)\Big)$
9: $\mathbf{z}(k)\leftarrow\mathbf{z}(k-1)-\alpha D\mathbf{z}(k-1)$
10: **结束 while**

注意，自适应步长仅用于第 8 行的$\mathbf{x}$动力学，而不用于第 9 行的$\mathbf{z}$ ，因为在$\mathbf{z}$ 上采用这样的自适应步长可能会破坏通过$\mathbf{L}$建立的共识。尽管我们在定理 III.2 中陈述的算法1收敛性证明中没有得出更快的收敛速率，但鉴于 Adam算法通常优于梯度下降法，我们预计算法1的收敛速度会快于 (12) [21]。

定理 III.2. （算法*1*的收敛性）。算法*1* $\hat{}$ $\hat{}$满足$\mathbf{x}(k)\to\mathbf{x}^\star$，当且仅当 $k\to\infty$ ***if*** $\|U\|_2<\infty$, $\|Y\|_2<\infty$,$\beta_1,\beta_2\in[0,1)$且 α 的选取满足

$$(\alpha^2+\mu)D^\top PD-\alpha\Big(D^\top P+PD\Big)\preceq 0 \tag{13}$$

对于某些 $P\succ 0$ 和 $\mu>0$。

证明。 根据算法1第9行，(13)意味着

$$\begin{aligned}&\mathbf{z}^\top(k)P\,\mathbf{z}(k)-\mathbf{z}^\top(k-1)P\,\mathbf{z}(k-1)\\&\quad\le-\mu\,\mathbf{z}^\top(k-1)D^\top PD\,\mathbf{z}(k-1),\\&\Rightarrow\|\,\mathbf{z}(k)\|_P-\|\,\mathbf{z}(k-1)\|_P\le-\mu\|D\,\mathbf{z}(k-1)\|_P.\end{aligned} \tag{14}$$

接下来，我们对 $k=1$、$\cdots$、N_I 中的(14)求和，得到

$$\|\mathbf{z}(N_I)\|_P \le \|\mathbf{z}(0)\|_P - \sum_{k=0}^{N_I-1} \mu \|D\,\mathbf{z}(k)\|_P. \tag{15}$$

不等式（15）意味着，当 $k \to \infty$ 时，$\|D\mathbf{z}(k)\|_P \to 0$。由于 $D = \mathrm{L} + \hat{U}\hat{U}^\top$，且 L 和 $\hat{U}\hat{U}^\top$ 均为半正定矩阵，因此对于任意 $\mathbf{z}(k) \neq 0$，$\|D\mathbf{z}(k)\|_P = 0$ 当且仅当 $\mathrm{L}\,\mathbf{z}(k) = \hat{U}\hat{U}^\top \mathbf{z}(k) = 0$。回想一下，$\mathbf{z}^\star := \hat{U}\mathbf{x}^\star - \hat{Y} - \mathrm{L}^{\frac{1}{2}}\mathbf{w}^\star$ 是最优解当且仅当式（6）成立，因此我们可以根据 $\|D\,\mathbf{z}(k)\|_P \to 0$ 得出 $\mathbf{z}(k) \to \mathbf{z}^\star$。由于 $\|\mathbf{z}^\star\|_2 < \infty$ 且 $\mathrm{L}^{\frac{1}{2}}\mathbf{z}^\star = 0$，对于任意满足 $\|\mathbf{x}(k)\|_2 < \infty$ 的 $\mathbf{x}(k)$，都存在一个 $\mathbf{w}(k)$ 满足

$$\mathbf{z}^\star = \hat{U}\,\mathbf{x}(k) - \hat{Y} - \mathbf{L}^{\frac{1}{2}}\,\mathbf{w}(k).$$

求解上述方程的$\mathbf{x}(k)$和$\mathbf{w}(k)$是(5)的最优解。换言之，我们可以得出$\mathbf{x}(k) \to \mathbf{x}^\star$ 给定$\mathbf{z}(k) \to \mathbf{z}^\star$ 为 $k \to \infty$，通过证明$\|\mathbf{x}(k)\|_2 < \infty$ 对于所有 $k \in \mathbb{N}$，或者

$$\|\mathbf{x}(k_1) - \mathbf{x}(k_1)\|_2 < \infty, \quad \forall\, k_1,\, k_2 \in \mathbb{N} \tag{16}$$

为证明（16）成立，我们首先指出存在一个 $C_g < \infty$，使得

$$\sum_{k=1}^{\infty} \|\hat{U}^\top \mathbf{z}(k)\|_2 = \sum_{k=1}^{\infty} \|g(k)\|_2 \le C_g. \tag{17}$$

否则，当 $N_I \to \infty$ 时，(15)不成立。接下来，我们分析在算法1中$\mathbf{x}(k)$的更新中出现的$\hat{s}_1(k)./(\sqrt{\hat{s}_2(k)}+\epsilon)$级数：

$$\begin{aligned}
&\left\|\sum_{k=1}^{\infty}\left(\hat{s}_1(k)./(\sqrt{\hat{s}_2(k)}+\epsilon)\right)\right\|_2 \le \frac{1}{\epsilon}\left\|\sum_{k=1}^{\infty}\hat{s}_1(k)\right\|_2 \\
=&\frac{1}{\epsilon}\left\|\sum_{k=1}^{\infty}\frac{\beta_1^k s_1(0) + (1-\beta_1)\sum_{i=1}^{k}\beta_1^{k-i}g(i)}{1-\beta_1^k}\right\|_2,
\end{aligned} \tag{18}$$

其中，算法1的第4行用于推导式（18）。由于$s_1(0) = 0$，我们可以将式（18）简化为：

$$\begin{aligned}
&\left\|\sum_{k=1}^{\infty}\left(\hat{s}_1(k)./(\sqrt{\hat{s}_2(k)}+\epsilon)\right)\right\|_2 \\
\le&\frac{1-\beta_1}{\epsilon}\left\|\sum_{k=1}^{\infty}\frac{1}{1-\beta_1^k}\sum_{i=1}^{k}\beta_1^{k-i}g(i)\right\|_2 \\
=&\frac{1-\beta_1}{\epsilon}\left\|\sum_{j=1}^{\infty}g(j)\cdot\left(\sum_{h=j}^{\infty}\frac{\beta_1^{h-j}}{1-\beta_1^h}\right)\right\|_2 \\
\le&\frac{1-\beta_1}{\epsilon}\cdot\frac{1}{(1-\beta_1)^2}\sum_{j=1}^{\infty}\|g(j)\|_2 \le \frac{C_g}{\epsilon(1-\beta_1)} < \infty
\end{aligned} \tag{19}$$

不等式（19）意味着（16）成立，从而完成证明。

□
□

注释 III.3.（算法*1*与***Adam***算法的比较）。通过删除算法*1*的第*9*行，并将 $g\,(k)$ 从 $\hat{U}^\top \mathbf{z}\,(k-1)$ 的含义改为关于 $\mathbf{x}$ 的某个目标函数的梯度，算法*1*实际上就是*Adam*算法。这种改变导致证明路径大不相同收敛到最优的 $\mathbf{x}^\star$。然而，基本假设相似。*Adam*算法 *[18,* 定理 *10.5]* 中的收敛性结论直接假设(16) 和(17)成立；对于算法*1*，这一假设则隐含在(13)中，而这实际上会导出(16) 和(17)，如证明所示。最后但同样重要的是，*Adam*算法 *[18,* 定理 *10.5]* 中的收敛性结论还对 β_1 和 β_2 提出了额外假设，用于推导遗憾界（或刻画收敛速率）。由于算法*1*并未对所有状态变量使用 β_1 和 β_2 （它们仅用于更新$\mathbf{x}$），因此尚不清楚如何对 β_1 和β_2 作出额外假设，以帮助刻画算法*1*的收敛速率。实践中，根据文献中 *Adam* 算法的数值经验为算法*1*选择 β_1 和 β_2，效果相当不错。

我们通过指出算法1具有一些理想的性质来结束本节：(1)只有虚拟变量$\mathbf{z}$ 通过通信网络进行交换；(2)检索子模型，$\mathbf{x}_i$（或 $A_{\mathcal{D}_{u,\,i}}$），该子模型对代理i 有用，需要从$\mathbf{z}$ 中获取本地数据 $U_{\mathcal{D}_{u,\,i}}$ 和 $Y_{\mathcal{D}_{u,\,i}}$，这同时保护了隐私并增强了安全性。我们认为，与其他模型辨识算法相比，这些是算法1的独特优势。

IV. 数值研究

我们通过两个测试案例验证所提出的分布式模型辨识算法。首先，我们展示了通过加入Adam算法自适应步长，与(12)相比，算法1实现了更快的收敛速度。随后，我们应用所提出的算法来辨识改进的IEEE 37节点系统的LinDistFlow模型 [24]（或其等价形式），并展示了利用所辨识模型所获得的令人满意的控制性能。

A. 一个小规模示例

我们随机生成一个线性网络系统，如下所示

$$y = [y_1; y_2; \cdots, y_N] = \sum_{i=1}^{N} A_{\mathcal{D}_{u,i}} u_i, \tag{20}$$

其中$u_i \in \mathbb{R}^{|\,\mathcal{D}_{u,i}\,|}$和$y_i \in \mathbb{R}^{|\,\mathcal{D}_{y,i}\,|}$是代理 i的输入和输出，$|\,\mathcal{D}_{u,i}\,| \in \{4,5\}$、$|\,\mathcal{D}_{y,i}\,| \in \{3,4\}$以及 $N = 5$。矩阵 $A = [A_{\mathcal{D}_{u,1}}, A_{\mathcal{D}_{u,2}}, \cdots, A_{\mathcal{D}_{u,N}}]$和 u_i 的所有元素均按照标准差为1的标准正态分布随机生成。我们假设，当数据矩阵 U 和 Y 以 T个步骤构建完成时，模型辨识算法（12）和算法1开始运行，$T = 2\sum_{i=1}^{N}|\,\mathcal{D}_{u,i}\,|$。算法参数设置为 $\alpha = 10^{-3}$、$\beta_1 = 0.9$、$\beta_2 = 0.95$和 $\epsilon = 10^{-8}$。如图1和图2所示，（12）和算法1都使估计模型收敛到实际模型。算法1具有明显更快的收敛速率，尤其是在最初的几百次迭代中。我们推测，几百步之后影响收敛的主导因素是$\mathbf{z}$的共识，因此此后（12）与算法1之间没有太大差异。

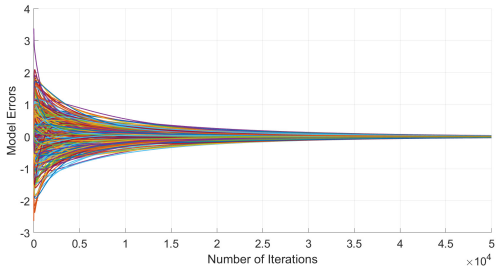


图1。通过实施（12）得到的估计 A 和 $A^\star$ 中每个元素的绝对值差。

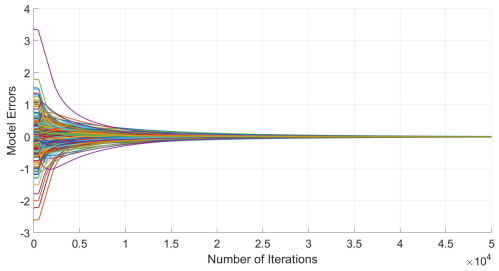


图2. 算法1中估计的 A 和 $A^\star$ 各元素的绝对值差异。

B. 改进的 *IEEE 37* 母线测试系统

在本节中，我们应用算法1来识别用于电压幅值调节的 LinDis-tFlow 模型 A 。我们对 IEEE 37 母线测试系统进行修改，使其接入光伏系统，如图 3 所示。我们假设每个光伏母线都充当一个代理，并且知道其本地有功/无功功率注入以及该母线的电压幅值，还可以调整其功率注入以帮助调节电压幅值。光伏母线 i 的功率注入被收集为控制变量 $u_i \in \mathbb{R}^2$ ，电压幅值测量值则由 $y_i \in \mathbb{R}_+$收集。我们假设已建立一个连通的分布式通信网络 $\mathcal{G}$ ，用于运行算法1以及 [14]中提出的基于分布式反馈的算法。

实施在 [14] 中开发的分布式反馈控制算法需要一个 LinDistFlow模型 A 或其等价形式。尽管该模型 A 不需要非常

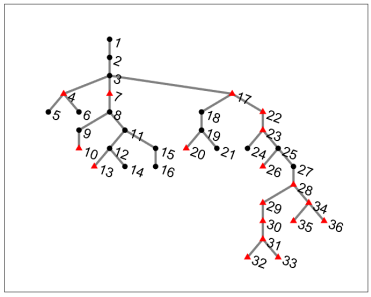


图3. 修改后的IEEE 37 节点系统示意图，其中以红色三角形突出显示的节点为 PV 节点

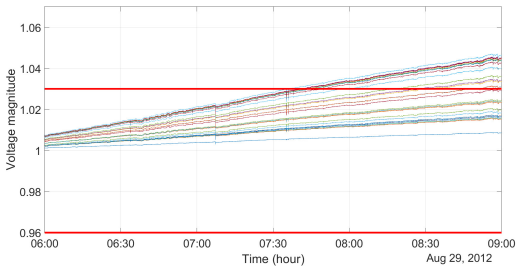


图4. 无控制时电压幅值（标幺值）随时间的变化。

如 [25], 中所述，仍然需要一个较为准确的 A 。我们模拟以下情形：（i）不对电压幅值进行控制；（ii）采用分布式反馈控制，其中 A 根据给定的线路阻抗和网络拓扑得出；（iii）采用分布式反馈控制，其中 A 由算法1离线辨识；（iv）采用分布式反馈控制，其中 A 由算法1在线辨识，每当有新的 u 和 y 数据对可用时，就替换最旧的数据对。算法参数 α、 β_1、 β_2 和 ϵ 与算法1的第一个数值示例相同。用于构建输入和输出数据矩阵的时间实例数 U 和 Y由 $T = 140$设定。

图4–7展示了四种场景下的电压幅值。不出所料，已知 A 时的控制性能名列前茅。一个有趣的观察是，对于场景(iii)，所有代理识别得到的$A_{\mathcal{D}_{u,i}}$所拼接成的 A $A_{\mathcal{D}_u}$，与根据给定阻抗和拓扑推导出的 A 截然不同，无论我们如何初始化$A_{\mathcal{D}_{u,i}}$ （或$\mathbf{x}_i$）都是如此。然而，该场景实现了与场景(ii)相似的电压调节效果。我们的解释是，由于可能存在多种有效的线性化模型，因此通过算法1识别出了另一种形式的线性化潮流模型。预计场景(iv)的性能最佳，因为它会根据在线运行点调整模型A （可以理解为调整线性化点）。然而，如图7所示，与场景(ii)和(iii)相比，我们观察到受调节的电压幅值波动略大。在更新U 和 Y 之前的步数可能需要进行一些微调，以获得更好的性能。总的来说，场景(iii)–(iv)均实现了令人满意的电压调节，同时将功率注入和电压幅值测量保留在本地，并且数据包交换仅限于与电压约束相关的对偶变量，以及分布式算法中的虚拟变量。

V. CONCLUSION

在本文中，我们提出了一种分布式模型辨识算法，使每个代理都能辨识出一个描述其局部控制与整体系统输出的子模型。该算法经过设计，使代理无需共享其局部数据；由于加入了Adam算法的自适应步长，其收敛速率实际上优于梯度下降共识算法。本文重点关注该算法在配电系统中的应用，但我们设想，所提出的算法可能对许多其他网络化系统中的去中心化控制或分布式控制同样有用。

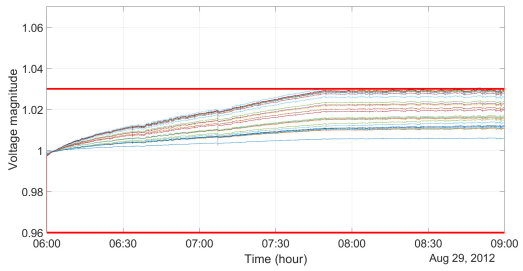


图5.采用基于分布式反馈控制和已知LinDistFlow 模型时的电压幅值（标幺值）随时间的变化。

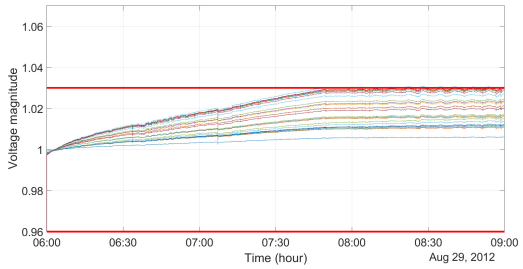


图 6.采用基于分布式反馈控制和所识别的LinDistFlow 模型时，电压幅值（标幺值）随时间的变化。

机器人学、经济学、电信等系统。我们近期的工作将致力于改进在线分布式模型辨识，以获得适用于非线性或时变线性系统的更优线性化模型。我们未来的工作还包括针对一些潜在应用，对所提出的算法进行更深入的测试。

参考文献

[1] P. Voigt 和 A. Von dem Bussche，《欧盟通用数据保护条例（GDPR）》，实用指南，第*1*版，瑞士沙姆：*Springer* 国际出版公司，第10卷，第3152676期，第10–5555页，2017年。[2] P. Bukaty，《加利福尼亚州消费者隐私法（*CCPA*）：实施指南》。IT Governance Ltd，2019年。[3] J. Kone˘cn`y、H. B. McMahan、F. X. Yu、P. Richt´arik、A. T. Suresh 和 D. Bacon，“联邦学习：提升通信效率的策略”，*arXiv*预印本 *arXiv:1610.05492*，2016年。[4] L. Li、Y. Fan、M. Tse 和 K.-Y. Lin，“联邦学习应用综述”，《计算机与工业工程》，第149卷，第106854页，2020年。[5] Y. Wang、I. L. Bennani、X. Liu、M. Sun 和 Y. Zhou，“电力消费者特征识别：一种联邦学习方法”，《*IEEE*智能电网汇刊》，第12卷，第4期，第3637–3647页，2021年。

[6] J. Lin、J. Ma 和 J. Zhu，“一种用于概率性社区级表后太阳能发电分解的隐私保护联邦学习方法”，《IEEE智能电网汇刊》，第13卷，第1期，第268–279页，2021年。[7] B. Kroposki、A. Bernstein、J. King、D. Vaidhynathan、X. Zhou、C.-Y. Chang 和 E. Dall’Anese，“自主能源电网：利用大量分布式能源资源控制未来电网”，《IEEE电力与能源杂志》，第18卷，第6期，第37–46页，2020年。[8] S. Bolognani、N. Bof、D. Michelotti、R. Muraro 和 L. Schenato，“通过电压相关性分析识别配电网拓扑”，载于第52届IEEE决策与控制会议，2013年，第1659–1664页。[9] O. Ardakanian、V. W. Wong、R. Dobbe、S. H. Low、A. von Meier、C. J. Tomlin 和 Y. Yuan，“关于配电网辨识”，《IEEE网络系统控制汇刊》，第6卷，第3期，第950–960页，2019年。[10] A. Primadianto 和 C.-N. Lu，“配电系统状态估计综述”，《IEEE电力系统汇刊》，第32卷，第5期，第3875–3883页，2016年。[11] M. C. de Almeida 和 L. F. Ochoa，“一种改进的三相不平衡配电系统状态估计器”，《IEEE电力系统汇刊》，第32卷，第2期，第1463–1473页，2016年。[12] Y. Guo、X. Zhou、C. Zhao、Y. Chen、T. Summers 和 L. Chen，“利用状态估计反馈求解配电网最优潮流”，载于2020年美国控制会议（ACC），2020年，第3148–3155页。[13] R. H. Salim 和 R. A. Ramos，“一种基于模型的不平衡电力系统小信号稳定性评估方法”，《IEEE电力系统汇刊》，第27卷，第4期，第2006–2014页，2012年。[14] C.-Y. Chang、M. Colombino、J. Cort´es 和 E. Dall’Anese，“用于分布式基于反馈的优化的鞍点流动力学”，《IEEE控制系统快报》，2019年。[15] J. Zhang、P. Wang 和 N. Zhang，“采用线性回归估计配电网导纳矩阵”，《IEEE电力系统汇刊》，第36卷，第5期，第4896–4899页，2021年。[16] P. Stegagno 和 C. Yuan，“多智能体系统的分布式协作自适应状态估计与系统辨识”，《IET控制理论与应用》，第13卷，第6期，第815–822页，2019年。[17] Y. Huang、Z. Meng 和 J. Sun，“通过优化方法求解大规模线性方程的可扩展分布式最小二乘算法”，《自动化》，第146卷，第110572页，2022年。[18] D. P. Kingma 和 J. Ba，“Adam：一种随机优化方法”，arXiv预印本 arXiv:1412.6980，2014年。[19] T. Chilimbi、Y. Suzue、J. Apacible 和 K. Kalyanaraman，“Project Adam：构建高效且可扩展的深度学习训练系统”，载于第11届 {USENIX} 操作系统设计与实现研讨会（{OSDI} *14*），2014年，第571–582页。[20] C. Jose 和 F. Fleuret，“通过加权近似秩分量分析实现可扩展度量学习”，载于《计算机视觉—ECCV 2016：第14届欧洲会议》，荷兰阿姆斯特丹，2016年10月11日至14日，第V部分论文集，Springer，第875–890页。[21] S. Ruder，“梯度下降优化算法综述”，arXiv预印本 arXiv:1609.04747，2016年。[22] C. De Persis 和 P. Tesi，“数据驱动控制公式：稳定化、最优性与鲁棒性”，《IEEE自动控制汇刊》，第65卷，第3期，第909–924页，2019年。[23] C.-Y. Chang 和 A. Bernstein，“含噪数据系统的鲁棒数据驱动控制”，arXiv预印本 arXiv:2207.09587，2022年。[24] M. E. Baran 和 F. F. Wu，“配电系统中的网络重构：降低损耗与实现负荷平衡”，《IEEE电力输送汇刊》，第4卷，第2期，第1401–1407页，1989年。[25] M. Colombino、J. W. Simpson-Porco 和 A. Bernstein，“迈向基于反馈的优化的鲁棒性保证”，arXiv预印本 arXiv:1905.07363，2019年。

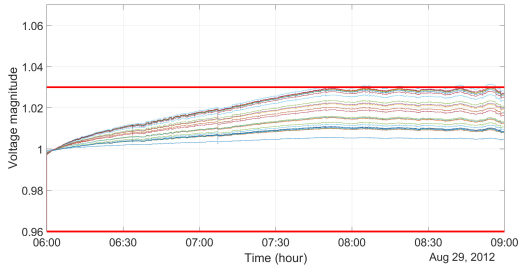


图 7. 采用分布式反馈控制和在线更新的已辨识 LinDistFlow 模型下随时间变化的电压幅值（标幺值）。