Deep Learning for Scalable Optimal Design of Incremental Volt/VAR Control Rules

Sarthak Gupta, *Graduate Student Member, IEEE,* Ali Mehrizi-Sani, *Senior Member, IEEE,* Spyros Chatzivasileiadis, *Senior Member, IEEE,* and Vassilis Kekatos, *Senior Member, IEEE*

***Abstract*—Volt/VAR control rules enable distributed energy resources (DER) to autonomously regulate voltage in distribution grids. The Volt/VAR rules provisioned by the IEEE Standard 1547 take on a piecewise-linear shape. However, its maximum slope is upper bounded to ensure stability, and that may hamper their voltage regulation performance. This limitation can be surpassed by adding a memory term to the control rule, and thus, obtain a so-termed incremental control rule. This letter aims to optimally customize the shape of incremental rules across buses to attain desirable voltage profiles. Albeit this task can be posed as a bilevel program, we pursue a more scalable approach by reformulating it as a deep learning task. The idea is that Volt/VAR dynamics can be captured by a recursive neural network (RNN). Interestingly, the RNN weights correspond to the parameters of the control rule; the RNN input to the grid loading conditions; and the RNN output to the equilibrium voltages. Therefore, the optimal rule parameters can be found upon training the RNN so its output (equilibrium voltages) approach unity. Training is performed by feeding the RNN with representative scenarios of the anticipated grid loading conditions. The RNN depth depends on the settling time of Volt/VAR dynamics. Because the discrete-time Volt/VAR dynamics can be viewed as iterations of a proximal gradient descent (PGD) algorithm, we also leverage Nesterov's accelerated PGD iterations to reduce the RNN depth. The RNN is never implemented in the field. Training this RNN is equivalent to solving the optimal rule design in a more computationally efficient manner. Analytical findings and numerical tests corroborate that the proposed solution can be neatly adapted to single- and multi-phase feeders. The proposed approach could be of general interest in designing piecewise-linear controllers acting on linear plants.**

***Index Terms*—Multiphase feeders, gradient backpropagation.**

I. Introduction

Local Volt/VAR control facilitates voltage regulation on distribution grids by providing reactive power compensation from DERs equipped with smart inverters. Different from centralized control schemes that entail high communication/computational burden, local rules decide DER setpoints based on local measurements. Although the general shape of such rules is prespecified, it is not clear how a utility can fine-tune these rules on a per-bus basis and depending on grid loading conditions to achieve better voltage profiles. This letter deals with this exact problem of *optimal rule design* (ORD).

Volt/VAR control rules can be categorized into *non-incremental* and *incremental* ones. The former compute DER reactive power setpoints based on local voltage readings. The IEEE Standard 1547.8 prescribes such non-incremental rules as piecewise-linear functions of voltage [1]. On the other hand, incremental Volt/VAR rules compute the *change* in VAR setpoints as a function of voltage [2]–[5].

The existing literature on ORD can be classified into *stability*- and *optimality*-centric approaches. Stability-centric ones study the effect of Volt/VAR rules as a closed-loop dynamical system, which may be rendered unstable under steep slopes of non-incremental rules [6], [7]. In fact, to ensure stability, non-incremental rules may have to compromise on the quality of their steady-state voltage profile [4], [7]. Incremental rules do not experience stability limitations [8], and can thus, achieve improved voltage profiles. Nonetheless, such improvements may come at the expense of longer settling times of the associated Volt/VAR dynamics [7].

Optimality-centric works focus on designing stable control rules to minimize a voltage regulation objective. To this end, optimization-based strategies have been employed to design affine non-incremental rules using heuristics [9]–[11]. Two of our recent works in [12] and [13] have addressed the problem of optimally designing the slope, deadband, saturation, and reference voltage. ORD in single-phase feeders is tackled as a bilevel optimization in [12]. Reference [13] proposes RNN-based digital twins that emulate non-incremental Volt/VAR dynamics, and reformulates ORD as an RNN training task for single-/multi-phase feeders.

This letter considers optimally designing incremental Volt/VAR control rules. Although the general shape of the rules is prespecified, fine-tuning the precise location of breakpoints and customizing them per bus seem to have an important effect on the steady-state equilibrium voltage profiles. Technical contributions are on three fronts: *c1)* Although the ORD task can be posed as a mixed-integer nonlinear program, it does not scale well with the numbers of DERs, nodes, and grid loading scenarios. To address this challenge, the genuine idea here is to reformulate ORD as a deep-learning task and leverage the fast software modules for training RNNs. We have put forth a similar approach for designing non-incremental control rules in [13]. However, migrating from non-incremental to incremental rules is non-trivial due to the different curve shapes, stability, and convergence rate properties. The convergence rate is directly tied to the depth of the RNN modeling the Volt/VAR dynamics. *c2)* Using the properties of the proximal operator, we simplify the convergence analysis and derive the convergence rate for multiphase feeders. *c3)* To further expedite ORD for incremental rules, we suggest implementing accelerated Nesterov-type variants of the rules to yield a shallower RNN emulator.

S. Gupta, A. Mehrizi-Sani, and V. Kekatos are with the ECE Dept., Virginia Tech, Blacksburg, VA 24061, USA. Emails: gsarthak,mehrizi,kekatos@vt.edu. S. Chatzivasileiadis is with the Wind Dept., Technical University of Denmark, Lyngby, Denmark. Work supported by US NSF under grant 2034137.

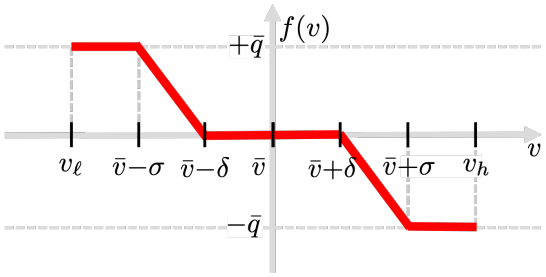

Fig. 1. Non-incremental Volt/VAR control rule provisioned by the IEEE Std. 1547 for the interconnection of DERs [1].

Recent works [14], [15] also design incremental Volt/VAR rules using NNs. Different from our approach where a RNN is used only to speed up ORD, [14], [15] train single-layer NNs to be used on the field. The NN of [14] is trained on a per-bus basis using reinforcement learning to handle transient voltage dynamics, whereas our RNN is trained prior to deployment over different scenarios to drive steady-state voltages to unity over the next couple of hours. Similar to our approach, reference [15] trains NNs beforehand in a supervised fashion: It first creates a labeled dataset upon solving a number of optimal power flow (OPF) problems. It subsequently trains the NN for each bus *independently* based on optimal voltage-injection pairs locally for this bus at equilibrium. However, this local mapping may not be a bijection because the same value of optimal voltage at a specific bus may correspond to different values of optimal injection at that bus. Reference [16] also pursues an *OPF-then-learn* approach to fit OPF solutions to piecewise-linear rules using a sum-of-squares approach. Outside the domain of Volt/VAR control, [17] employs an RNN to capture frequency control dynamics. Focusing again on transient behavior, the depth of this RNN [17] is controlled by the length of the considered transient period. In contrast, we analytically relate the RNN depth to the convergence rate of Volt/VAR dynamics. Distinctly from [14], [15], [17], our RNN is never implemented by DERs. Training this RNN is equivalent to solving the ORD, yet more efficiently.

II. Volt/VAR Control Rules

Consider a radial feeder serving N buses indexed by $n = 1, \ldots, N$. Let $(\mathbf{q}^\ell, \mathbf{q})$ collect reactive loads and generations at all nodes. Vectors $(\mathbf{p}, \mathbf{v})$ collect the net active power injections and voltage magnitudes at all nodes. The impact of $\mathbf{q}$ on $\mathbf{v}$ can be approximately captured using the next linearized grid model widely adopted in the literature [6], [7]

$$\mathbf{v} \simeq \mathbf{R}\mathbf{p} + \mathbf{X}(\mathbf{q} - \mathbf{q}^\ell) + v_0\mathbf{1} = \mathbf{X}\mathbf{q} + \tilde{\mathbf{v}} \tag{1}$$

where $\tilde{\mathbf{v}} := \mathbf{R}\mathbf{p} - \mathbf{X}\mathbf{q}^\ell + v_0\mathbf{1}$ models the underlying *grid conditions*, and v_0 is the substation voltage. Vector $\tilde{\mathbf{v}}$ represents the impact of non-controlled quantities $(\mathbf{p}, \mathbf{q}^\ell)$ on voltages. Matrices $(\mathbf{R}, \mathbf{X})$ depend on the feeder topology. For single-phase feeders, they are symmetric positive definite with positive entries [18]. For multiphase feeders, they are non-symmetric and have positive and negative entries [4], [13].

Vector $\mathbf{q}$ in (1) carries the reactive injections by DERs we would like to control. Per the non-incremental rules of the IEEE Std. 1547 [1], DER setpoints are decided based on the Volt/VAR curve of Fig. 1, which is parameterized by $(\bar{v}, \delta, \sigma, \bar{q})$. The standard further constrains these parameters within a polytopic feasible set [1], [12]. The negative slope of the linear segment of the curve in Fig. 1 can be expressed as $\alpha := \frac{\bar{q}}{\sigma - \delta}$. The interaction of Volt/VAR rules with the feeder gives rise to the piecewise-linear discrete-time dynamics:

$$\mathbf{v}^t = \mathbf{X}\mathbf{q}^t + \tilde{\mathbf{v}} \quad \text{(feeder)} \tag{2a}$$

$$\mathbf{q}^{t+1} = \mathbf{f}(\mathbf{v}^t) \tag{2b}$$

where $\mathbf{f}$ abstracts the control rule of Fig. 1 across inverters. If $\|\operatorname{dg}(\boldsymbol{\alpha})\mathbf{X}\|_2 < 1$, where $\operatorname{dg}(\boldsymbol{\alpha})$ is a diagonal matrix carrying the rule slopes over all buses on its diagonal, then the aforesaid dynamics are globally exponentially stable [7].

DER equilibrium setpoints cannot be expressed in closed form, but coincide with the minimizer of [6]

$$\min_{-\bar{\mathbf{q}} \le \mathbf{q} \le \bar{\mathbf{q}}} \frac{1}{2}\mathbf{q}^\top \mathbf{X}\mathbf{q} + \mathbf{q}^\top(\tilde{\mathbf{v}} - \bar{\mathbf{v}}) + \frac{1}{2}\mathbf{q}^\top \operatorname{dg}^{-1}(\boldsymbol{\alpha})\mathbf{q} + \boldsymbol{\delta}^\top|\mathbf{q}| \tag{3}$$

where $|\mathbf{q}|$ applies the absolute value on $\mathbf{q}$ entrywise. Problem (3) depends on rule parameters $(\bar{v}, \delta, \alpha, \bar{q})$ across all buses. We stack all rule parameters into the $4N$-long column vector $\mathbf{z} := [\bar{\mathbf{v}}; \boldsymbol{\delta}; \boldsymbol{\alpha}; \bar{\mathbf{q}}]$ borrowing MATLAB's shorthand notation. We denote by $\mathbf{q}_\mathbf{z}(\tilde{\mathbf{v}})$ the equilibrium setpoints, and by

$$\mathbf{v}_\mathbf{z}(\tilde{\mathbf{v}}) = \mathbf{X}\mathbf{q}_\mathbf{z}(\tilde{\mathbf{v}}) + \tilde{\mathbf{v}} \tag{4}$$

the related equilibrium voltages reached by Volt/VAR rules parameterized by $\mathbf{z}$ under grid conditions $\tilde{\mathbf{v}}$. Since (3) has a strictly convex objective, the equilibrium $\mathbf{q}_\mathbf{z}(\tilde{\mathbf{v}})$ is unique [6].

Optimal rule design (ORD) can be stated as the task of selecting $\mathbf{z}$ to bring equilibrium voltages $\mathbf{v}_\mathbf{z}(\tilde{\mathbf{v}})$ close to unity. To cater to diverse conditions, the utility may sample S loading scenarios $\{\tilde{\mathbf{v}}_s\}_{s=1}^S$ for the next hour, and find $\mathbf{z}$ as

$$\mathbf{z}^* \in \arg\min_{\mathbf{z}} \quad F(\mathbf{z}) := \frac{1}{S}\sum_{s=1}^{S} \|\mathbf{v}_\mathbf{z}(\tilde{\mathbf{v}}_s) - \mathbf{1}\|_2^2 \qquad \text{(ORD)}$$

$$\text{subject to} \quad (4) \text{ and } \mathbf{z} \in \mathcal{Z}.$$

The focus is on equilibrium voltages as utilities and reliability standards focus primarily on time-averaged voltages and ignore transient effects. Once found, the customized rules $\mathbf{z}^*$ are sent to DERs to operate autonomously over the next hour. Note that $\mathbf{v}_\mathbf{z}(\tilde{\mathbf{v}}_s)$ depends on $\mathbf{z}$ because the equilibrium setpoints $\mathbf{q}_\mathbf{z}(\mathbf{v}_s)$ in (4) are the minimizers of problem (3), which is parameterized by $\mathbf{z}$. When solving (ORD) for non-incremental rules, the feasible set $\mathcal{Z}$ consists of the polytopic constraints imposed on $\mathbf{z}$ by the IEEE Std. 1547 as well as additional constraints on $\boldsymbol{\alpha}$ to ensure $\|\operatorname{dg}(\boldsymbol{\alpha})\mathbf{X}\|_2 < 1$; see [19]. Hence, the feasible set $\mathcal{Z}$ can be quite confined. This can lead to less desirable voltage profiles, i.e., higher objective values $F(\mathbf{z}^*)$.

The aforesaid issue can be addressed by replacing the non-incremental Volt/VAR rules of IEEE Std. 1547 by incremental ones as suggested in [2]–[4], [8], [20]. Incremental rules express the *change* rather than the actual value in reactive setpoints as a function of voltage. One option for incremental rules is to implement a proximal gradient descent (PGD) algorithm solving (3) as proposed in [4]. In this case, the control rule updates coincide with the iterations of PGD. Using

incremental rules, there are no major stability limitations. Thus, feasible set $\mathcal{Z}$ is enlarged as we only need to ensure

$$\mathbf{z} \geq \mathbf{0} \quad \text{and} \quad 0.95 \cdot \mathbf{1} \leq \bar{\mathbf{v}} \leq 1.05 \cdot \mathbf{1} \quad \text{and} \quad \bar{\mathbf{q}} \leq \hat{\mathbf{q}} \tag{5}$$

where $\hat{\mathbf{q}}$ is the vector of reactive power ratings across DERs.

The PGD algorithm is an extension of gradient descent to handle constraints and non-differentiable costs [4]. At iteration t, PGD proceeds in two steps: *s1)* It first computes the gradient of the first two terms of the cost in (3), which is $\mathbf{X}\mathbf{q}^t+\tilde{\mathbf{v}}-\bar{\mathbf{v}} = \mathbf{v}^t - \bar{\mathbf{v}}$. Here $\mathbf{q}^t$ is the latest estimate of the minimizer of (3); *s2)* PGD then updates $\mathbf{q}^{t+1}$ as the minimizer of

$$\min_{-\bar{\mathbf{q}} \leq \mathbf{q} \leq \bar{\mathbf{q}}} \frac{1}{2}\mathbf{q}^\top \operatorname{dg}^{-1}(\boldsymbol{\alpha})\mathbf{q} + \boldsymbol{\delta}^\top|\mathbf{q}| + \frac{1}{2\mu}\|\mathbf{q} - (\mathbf{v}^t - \bar{\mathbf{v}})\|_2^2 \tag{6}$$

for a step size $\mu > 0$. The last problem involves the last two terms in the cost of (3) regularized by the Euclidean distance of $\mathbf{q}$ to the gradient $(\mathbf{v}^t - \bar{\mathbf{v}})$ computed in step *s1)*.

Converting PGD to control rules, step *s1)* is run by the feeder when injecting $\mathbf{q}^t$ and measuring local voltage deviations $\mathbf{v}^t - \bar{\mathbf{v}}$; see (7a). Step *s2)* is run by each DER independently as (6) is separable across buses; see (7b)–(7c). Solving (6) yields the next update as derived before in [4]:

$$\mathbf{v}^t = \mathbf{X}\mathbf{q}^t + \tilde{\mathbf{v}} \quad \text{(feeder)} \tag{7a}$$

$$y_n^t = \tilde{\alpha}_n \cdot \left(q_n^t - \mu(v_n^t - \bar{v}_n)\right) \tag{7b}$$

$$q_n^{t+1} = g_n\left(y_n^t\right) \tag{7c}$$

where $g_n(y_n)$ is the *proximal operator* shown in Figure 2. The operator depends on the transformed parameters

$$\tilde{\alpha}_n := \frac{1}{1+\mu/\alpha_n} \quad \text{and} \quad \tilde{\delta}_n := \frac{\delta_n}{1+\mu/\alpha_n}. \tag{8}$$

We henceforth switch the representation of rule parameters from $\mathbf{z} = [\bar{\mathbf{v}}; \boldsymbol{\delta}; \boldsymbol{\alpha}; \bar{\mathbf{q}}]$ to $\tilde{\mathbf{z}} := [\bar{\mathbf{v}}; \tilde{\boldsymbol{\delta}}; \tilde{\boldsymbol{\alpha}}; \bar{\mathbf{q}}]$. This is without loss of generality as the transformation in (8) is a bijection, and so one can work exclusively with $\tilde{\mathbf{z}}$. The feasible set $\tilde{\mathcal{Z}}$ is defined by (5) along with the additional constraint $\tilde{\boldsymbol{\alpha}} \leq \mathbf{1}$.

Albeit the problem physics predetermine the shape of the rule (7b)–(7c), its breakpoints in Figure 2 can be fine-tuned on a per-bus basis to improve voltage profiles. As with non-incremental rules, the rules in (7b)–(7c) are driven by local data, but now q_n^{t+1} depends on (v_n^t, q_n^t), and not v_n^t alone. Non-incremental and incremental rules alike solve (3). Hence, they both converge to the same unique equilibrium. The advantage of incremental rules is that they are globally exponentially stable for all $\boldsymbol{\alpha}$ as long as $\mu < 2/\lambda_{\max}(\mathbf{X})$; see [4] and Proposition 1 here. It is worth stressing that $\mathbf{z}$ here does not have the same physical interpretation as in non-incremental rules (slopes, deadband, or saturation), though $\mathbf{z}$ parameterizes (3) in the same way for both rules. Our ORD does not optimize over μ, which is assumed fixed. As evident from (3), μ does not affect steady-state profiles, but only transient behavior. One could augment $\mathbf{z}$ with step sizes customized per bus and pursue ORD formulations to minimize settling times subject to voltage constraints. This goes beyond the scope of this letter.

Accelerated incremental rules. Although PGD rules enlarge $\mathcal{Z}$, their settling times can be long. They reach an ε-optimal cost of (3) exponentially fast; particularly within $-\frac{2\log\varepsilon}{\log 2}\kappa(\mathbf{X})$ iterations, where $\kappa(\mathbf{X})$ is the condition number of $\mathbf{X}$. Works [4], [21] put forth *accelerated* incremental rules based on accelerated PGD (APGD), shown to be globally exponentially stable. These rules need $-\frac{2\log\varepsilon}{\log 2}\sqrt{\kappa(\mathbf{X})}$ iterations to attain an ε-optimal cost, and take the form

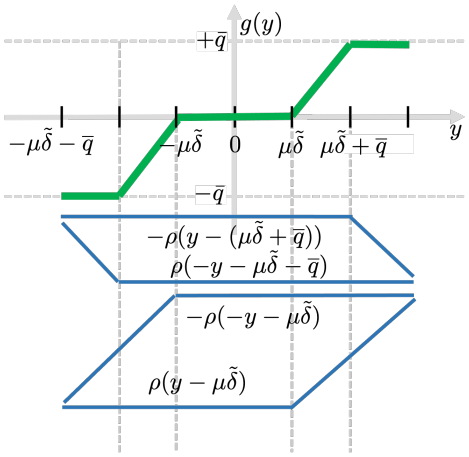

Fig. 2. Proximal operator $g(y)$ expressed as a sum of four shifted ReLUs.

$$\mathbf{v}^t = \mathbf{X}\mathbf{q}^t + \tilde{\mathbf{v}} \quad \text{(feeder)} \tag{9a}$$

$$\tilde{y}_n^t := (1+\beta_t)\, y_n^t - \beta_t y_n^{t-1} \tag{9b}$$

$$q_n^{t+1} := g_n\left(\tilde{y}_n^t\right) \tag{9c}$$

where $\beta_t := \frac{t-1}{t+2}$, while y_n^t and $g_n(y_n)$ are as defined in (7b)–(7c). Updates (9) remain local, but introduce additional memory as q_n^{t+1} depends on (v_n^t, q_n^t) and (v_n^{t-1}, q_n^{t-1}).

Remark 1. *For a given stable* $\mathbf{z}$*, dynamics under (non)-incremental rules converge to the same equilibrium* $\mathbf{q}_\mathbf{z}(\tilde{\mathbf{v}})$*. However, the feasible set for ORD with non-incremental rules includes additional polytopic constraints [13] and the stability constraint* $\|\operatorname{dg}(\boldsymbol{\alpha})\mathbf{X}\|_2 < 1$*, whereas set* $\tilde{\mathcal{Z}}$ *for ORD with incremental rules does not. Because* $\mathcal{Z} \subseteq \tilde{\mathcal{Z}}$*, incremental rules can attain lower* $F(\mathbf{z}^*)$*, and thus, improved voltage profiles.*

III. Deep Learning for ORD in 1ϕ Feeders

Solving (ORD) is challenging as it is a nonconvex bilevel program. Although it can be modeled as a mixed-integer nonlinear program, such an approach does not scale well with the number of DERs or scenarios for non-incremental rules [13]. Seeking a more scalable solution, we reformulate (ORD) as a deep learning task. The key idea is to design an RNN that emulates Volt/VAR dynamics under the rule of (7). To this end, note that $g_n(y_n)$ is a piecewise-linear function with four breakpoints. Interestingly, this operator can be expressed as the superposition of four rectified linear units (ReLUs) as illustrated in Fig. 2, where ReLUs are denoted by $\rho(\cdot)$. The intercepts of the ReLUs depend linearly on $(\tilde{\delta}_n, \bar{q}_n)$.

Building on this, one APGD iteration for DER n can be implemented by the 4-layer Deep Neural Network (DNN) in Fig. 3, whose weights depend affinely on $(\bar{v}_n, \tilde{\delta}_n, \tilde{\alpha}_n, \bar{q}_n)$. This DNN takes (q_n^t, v_n^t) as its input, and computes (q_n^{t+1}, y_n^t) at its output. It is termed IC_n and will be used as a building block to emulate Volt/VAR dynamics. This is achieved by the RNN

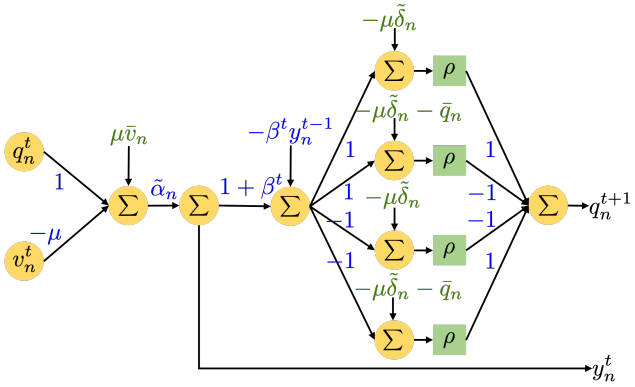

Fig. 3. A NN emulating the accelerated rules of (9). Plain rules can be modeled by dropping the second layer (setting $\beta^t = 0$) and output y_n^t.

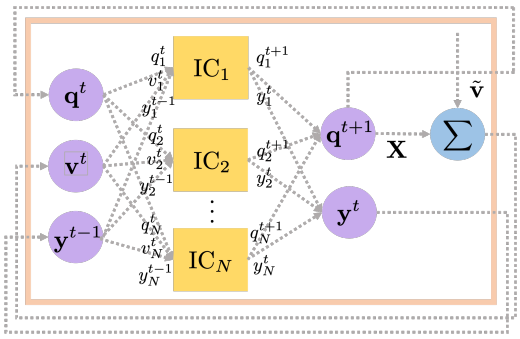

Fig. 4. RNN implementation for accelerated incremental Volt/VAR rules.

of Fig. 4. Here blocks IC_n are arranged vertically to model the parallel operation of DERs. Their outputs $\mathbf{q}^{t+1}$ are multiplied by $\mathbf{X}$, and the new voltage is computed as $\mathbf{v}^{t+1} = \mathbf{X}\mathbf{q}^{t+1} + \tilde{\mathbf{v}}$. This is repeated T times, with each repetition being referred to as *layer*. In other words, the RNN of Figure 4 is T layers *deep*. Thanks to the RNN structure, there is *weight sharing*, so the number of RNN weights is $4N$ rather than $4NT$.

The RNN takes a grid loading vector $\tilde{\mathbf{v}}_s$ as its input, and the rule parameters $\tilde{\mathbf{z}}$ as weights. At its output, it computes $\Phi(\tilde{\mathbf{v}}_s; \tilde{\mathbf{z}}) = \mathbf{v}_{\tilde{\mathbf{z}}}^T(\tilde{\mathbf{v}}_s)$, that is the voltage induced by the control rules (7a) or (9a) at time $t = T$ under conditions $\tilde{\mathbf{v}}_s$ and rule parameters $\tilde{\mathbf{z}}$. The voltage $\mathbf{v}_{\tilde{\mathbf{z}}}^T(\tilde{\mathbf{v}}_s)$ approaches the equilibrium voltage $\mathbf{v}_{\tilde{\mathbf{z}}}(\tilde{\mathbf{v}}_s)$ as T approaches infinity. *How can one select the RNN depth T so that the RNN output approximates well equilibrium voltages?* This is determined by the rate at which Volt/VAR dymamics converge to their equilibrium. As Volt/VAR dynamics coincide with the PGD iterations, we next determine T leveraging the convergence rate of PGD.

Proposition 1. *For the RNN of Fig. 4 to ensure $\|\Phi(\tilde{\mathbf{v}}; \tilde{\mathbf{z}}) - \mathbf{v}_{\tilde{\mathbf{z}}}(\tilde{\mathbf{z}})\|_2 \leq \epsilon_1$ for all $\tilde{\mathbf{v}}$, while emulating the control rules of (7) with $\mu = 2/(\lambda_{\min}(\mathbf{X}) + \lambda_{\max}(\mathbf{X}))$ and $\kappa = \lambda_{\max}(\mathbf{X})/\lambda_{\min}(\mathbf{X})$, its depth T should be*

$$T \geq \left(\frac{\kappa - 1}{2}\right) \log\left(\frac{2\|\mathbf{X}\|_2 \|\hat{\mathbf{q}}\|_2}{\epsilon_1}\right). \tag{10}$$

Proof: If $\mathbf{q}_{\mathbf{z}}(\tilde{\mathbf{v}}) = \mathbf{g}_{\tilde{\mathbf{z}}}(\tilde{\mathbf{y}})$ at equilibrium, rule (7c) yields:

$$\begin{aligned}
\|\mathbf{q}^t - \mathbf{q}_{\mathbf{z}}(\tilde{\mathbf{v}})\|_2 &= \|\mathbf{g}(\mathbf{y}^t) - \mathbf{g}_{\tilde{\mathbf{z}}}(\tilde{\mathbf{y}})\|_2 \leq \|\mathbf{y}^t - \tilde{\mathbf{y}}|_2 \\
&= \|\operatorname{dg}(\tilde{\boldsymbol{\alpha}})(\mathbf{I} - \mu\mathbf{X})\left(\mathbf{q}^{t-1} - \mathbf{q}_{\mathbf{z}}(\tilde{\mathbf{v}})\right)\|_2 \\
&\leq \|\operatorname{dg}(\tilde{\boldsymbol{\alpha}})\|_2 \cdot \|\mathbf{I} - \mu\mathbf{X}\|_2 \cdot \|\mathbf{q}^{t-1} - \mathbf{q}_{\mathbf{z}}(\tilde{\mathbf{v}})\|_2 \\
&\leq \|\mathbf{I} - \mu\mathbf{X}\|_2 \cdot \|\mathbf{q}^{t-1} - \mathbf{q}_{\mathbf{z}}(\tilde{\mathbf{v}})\|_2. \qquad (11)
\end{aligned}$$

The first inequality stems from the non-expansive property of the proximal operator $\mathbf{g}$. The next equality follows from (7b). The second inequality from the sub-multiplicative property of the spectral norm. The last inequality follows by the definition of spectral norm and because $\tilde{\alpha}_n \leq 1$ for all n.

If $\|\mathbf{I} - \mu\mathbf{X}\|_2 < 1$, inequality (11) implies that the dynamics in (7) are a non-expansive mapping, and thus, they converge exponentially fast to $\mathbf{q}^*$. Condition $\|\mathbf{I} - \mu\mathbf{X}\|_2 < 1$ holds when $\mu < 2/\lambda_{\max}(\mathbf{X})$. The norm $\|\mathbf{I} - \mu\mathbf{X}\|_2$ achieves its minimum of $\frac{\kappa-1}{\kappa+1}$ when $\mu_0 := 2/(\lambda_{\min}(\mathbf{X}) + \lambda_{\max}(\mathbf{X}))$. Plugging μ_0 in (11) and unfolding dynamics over t yields

$$\|\mathbf{q}^t - \mathbf{q}_{\mathbf{z}}(\tilde{\mathbf{v}})\| \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^t \|\mathbf{q}^0 - \mathbf{q}_{\mathbf{z}}(\tilde{\mathbf{v}})\| \leq 2\left(\frac{\kappa - 1}{\kappa + 1}\right)^t \|\hat{\mathbf{q}}\|.$$

For the voltage approximation error $\|\mathbf{v}^T - \mathbf{v}_{\tilde{\mathbf{z}}}(\tilde{\mathbf{v}})\|_2 = \|\mathbf{X}\left(\mathbf{q}^T - \mathbf{q}_{\tilde{\mathbf{z}}}(\tilde{\mathbf{v}})\right)\|_2$ at time T to be smaller than ϵ_1, we need

$$\|\mathbf{v}^T - \mathbf{v}_{\tilde{\mathbf{z}}}(\tilde{\mathbf{v}})\|_2 \leq 2\left(\frac{\kappa - 1}{\kappa + 1}\right)^T \|\mathbf{X}\|_2 \|\hat{\mathbf{q}}\|_2 \leq \epsilon_1.$$

This can be achieved by selecting T such that

$$T \geq \frac{\log\left(\frac{2\|\mathbf{X}\|_2\|\hat{\mathbf{q}}\|_2}{\epsilon_1}\right)}{\log\left(1 + \frac{2}{\kappa - 1}\right)} \geq \left(\frac{\kappa - 1}{2}\right) \log \frac{2\|\mathbf{X}\|_2 \|\hat{\mathbf{q}}\|_2}{\epsilon_1}.$$

where the last inequality follows from $\log(1 + x) \leq x$. □

Plugging the values $\|\mathbf{X}\|_2 = 0.463$ and $\kappa = 848$ for the IEEE 37-bus feeder, $\|\hat{\mathbf{q}}\|_2 = 0.1$, and $\epsilon_1 = 10^{-5}$ in (10), yields $T \geq 2,892$ layers, which is relatively large. A key contributor to this large T is the κ term in (10). This promulgates the adoption of accelerated rules (9), which are known to also converge exponentially fast, but with a $\mathcal{O}(\sqrt{\kappa})$ dependence. Interestingly, during implementation, one does not need to fix T to the above worst-case bounds. Leveraging dynamic computation graphs offered by Python libraries such as Pytorch, one may determine T *'on the fly'* depending on the convergence of $\mathbf{v}^t$ between pairs of successive layers.

Since the RNN emulates Volt/VAR dynamics, it can surrogate $\mathbf{v}_{\mathbf{z}}(\tilde{\mathbf{v}}_s)$ in (ORD). Then (ORD) can be posed as training a RNN over its weights $\tilde{\mathbf{z}} \in \tilde{\mathcal{Z}}$ or $\mathbf{z} \in \mathcal{Z}$. Grid loading scenarios $\{\tilde{\mathbf{v}}_s\}_{s=1}^S$ are treated as features and equilibrium voltages $\mathbf{v}_{\mathbf{z}}(\tilde{\mathbf{v}}_s)$ as predictions that should be brought close to the target value of $\mathbf{1}$ for scenarios s. The RNN can be trained using stochastic projected gradient descent (SPGD) as [13]

$$\tilde{\mathbf{z}}^{i+1} = \left[\tilde{\mathbf{z}}^i - \frac{\lambda}{2B}\nabla_{\tilde{\mathbf{z}}^i}\left(\sum_{s\in\mathcal{B}_i} \|\Phi(\tilde{\mathbf{v}}_s; \tilde{\mathbf{z}}) - \mathbf{1}\|_2^2\right)\right]_{\tilde{\mathcal{Z}}} \tag{12}$$

where $\lambda > 0$ is the learning rate; set $\mathcal{B}_i$ is a batch of B scenarios; and $[\cdot]_{\tilde{\mathcal{Z}}}$ is the projection onto $\tilde{\mathcal{Z}}$. Since $\tilde{\mathcal{Z}}$ consists of simple box constraints, projection essentially means clipping the values to the box. Lastly $\nabla_{\tilde{\mathbf{z}}^i}(\cdot)$ represents the gradient with respect to $\tilde{\mathbf{z}}$ evaluated at $\tilde{\mathbf{z}} = \tilde{\mathbf{z}}^i$, and is calculated efficiently thanks to *gradient back-propagation*.

Extensions: Model (1) can be replaced by more accurate ones upon linearization at a nominal loading scenario. Alternatively, the grid model can be substituted by a DNN already trained to solve the power flow task. Our design approach could also be used to design other (non)-incremental Volt/VAR rules. To move beyond piecewise-affine shapes, ReLUs could be replaced by other activation functions. As asserted by the universal approximation theorem, a DNN could in theory approximate any smooth nonlinear mapping. In general, the proposed approach could be of relevance to design nonlinear controllers beyond voltage regulation or electric grids.

IV. Deep Learning for ORD in Multiphase Feeders

In multiphase feeders, $\mathbf{X}$ is non-symmetric and has positive and negative entries. Hence, the rule analysis and design of Section III has to be revisited. For example, equilibrium setpoints cannot be found as the minimizers of an optimization as with (3). Moreover, increasing $\mathbf{q}$ does not necessarily mean all voltages increase. In multiphase feeders, the non-incremental rules of IEEE Std. 1547 remain globally exponentially stable as long as $\|\operatorname{dg}(\boldsymbol{\alpha})\mathbf{X}\|_2 < 1$ [13]. This is the same condition as in the single-phase setup. How about the stability and equilibrium of incremental rules in multiphase feeders? Recall that for single-phase rules, incremental rules were obtained as the PGD iterations solving (3). Lacking an equivalent inner optimization for multiphase feeders precludes a similar approach. Despite the rules of (7) not corresponding to PGD iterates anymore, they can still be shown to be stable for multiphase feeders.

Proposition 2. *Let* $\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$ *be the eigen-decomposition of matrix* $\mathbf{X}\mathbf{X}^\top$. *The incremental rules of* (7) *are globally exponentially stable for multiphase feeders if their step size is selected as* $\mu < \lambda_{\min}\left(\boldsymbol{\Lambda}^{-1/2}\mathbf{U}^\top\left(\mathbf{X}+\mathbf{X}^\top\right)\mathbf{U}\boldsymbol{\Lambda}^{-1/2}\right)$.

The claim follows readily by adopting the proof of Proposition 1: If μ is selected as above, then $\|\mathbf{I}-\mu\mathbf{X}\|_2 < 1$ follows from [4, Prop. 6]. Invoking a contraction argument, [4] also establishes the uniqueness of the equilibrium. As in single-phase, incremental rules in multiphase feeders allow us to enlarge feasible set $\mathcal{Z}$. Heed that different from the single-phase setting, incremental and non-incremental rules do not converge to the same equilibrium on multiphase feeders.

ORD for multiphase feeders can be posed as a deep learning task too, with some modifications. Firstly, $\mathbf{R}$ and $\mathbf{X}$ need to be altered. Secondly, the RNNs for multiphase feeders have $12N$ parameters, since each layer consists of $3N$ building modules corresponding to bus/phase pairs. Lastly, the step size has to be selected per Prop. 2. Adopting the proof of Prop. 1, we next find the minimum RNN depth in multiphase feeders.

Proposition 3. *Let the RNN of Fig. 4 implement the incremental rules of* (7) *on multiphase feeders with* μ *selected per Proposition 2. The RNN depth* T *ensuring voltage approximation error* $\|\Phi(\tilde{\mathbf{v}};\mathbf{z}) - \mathbf{v}_{\tilde{\mathbf{z}}}(\tilde{\mathbf{v}})\|_2 \leq \epsilon_1$ *for all* $\tilde{\mathbf{v}}$ *is*

$$T \geq \frac{\log \frac{\epsilon_1}{2\|\mathbf{X}\|_2\|\hat{\mathbf{q}}\|_2}}{\log\|\mathbf{I}-\mu\mathbf{X}\|_2}.$$

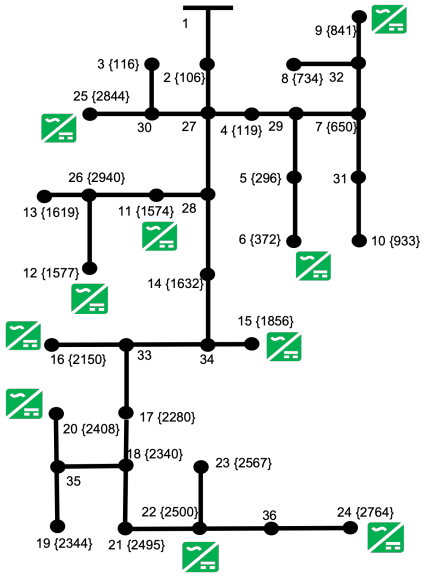

Fig. 5. The IEEE 37-bus feeder converted to single-phase. Node numbering follows the format `node number {panel ID}`. DERs marked green provide Volt/VAR control; the rest operate at unit power factor.

TABLE I
Incremental vs. non-incremental Volt/VAR control rules on the single-phase IEEE 37-bus feeder

Time	$\mathbf{q}=\mathbf{0}$	Non-incremental		Incremental	
	Obj. (p.u.)	Time (s)	Obj. (p.u.)	Time (s)	Obj. (p.u.)
1 pm	$3.01\cdot10^{-3}$	37.98	$3.68\cdot10^{-4}$	39.39	$3.66\cdot10^{-4}$
2 pm	$3.13\cdot10^{-3}$	42.93	$4.26\cdot10^{-4}$	37.91	$4.25\cdot10^{-4}$
3 pm	$4.24\cdot10^{-3}$	45.02	$8.59\cdot10^{-4}$	34.97	$8.50\cdot10^{-4}$
4 pm	$2.12\cdot10^{-3}$	48.30	$1.47\cdot10^{-4}$	38.52	$1.48\cdot10^{-4}$
5 pm	$8.53\cdot10^{-4}$	47.37	$9.70\cdot10^{-5}$	374.01	$6.90\cdot10^{-5}$

V. Numerical Tests

We benchmark RNN-based incremental rules against non-incremental rules from [13] on single- and multiphase feeders. Real-world data were sourced from the Smart* project on April 2, 2011 [22], as explained in [13]. The RNNs were trained using Pytorch. We first compare (non)-incremental rules, both designed via RNN training for the single-phase IEEE 37-bus feeder of Figure 5. Homes with IDs 20–369 were averaged 10 at a time and successively added as active loads to buses 2–26 as shown in Fig. 5. Active generation from solar panels was also added, as per the mapping in Fig. 5. Buses $\{6,9,11,12,15,16,20,22,24,25\}$ were assumed to host DERs with Volt/VAR control customized per bus.

Incremental rules were simulated in their accelerated rendition. Both sets of rules were trained over $S=80$ scenarios and 200 epochs with a learning rate of 0.001, using the Adam optimizer, and setting $\mu=1$ for incremental rules. To ensure repeatability, the results were repeated across several time periods between 1–6 PM, and are compiled in Table I. Incremental rules obtained marginally lower objectives than non-incremental rules across all periods, with a somewhat significant difference for the 5 PM period. This behavior is explained because incremental rules allow for a larger set $\mathcal{Z}$.

RNN-based incremental control rules were also contrasted with their non-incremental ones on the multiphase IEEE 13-

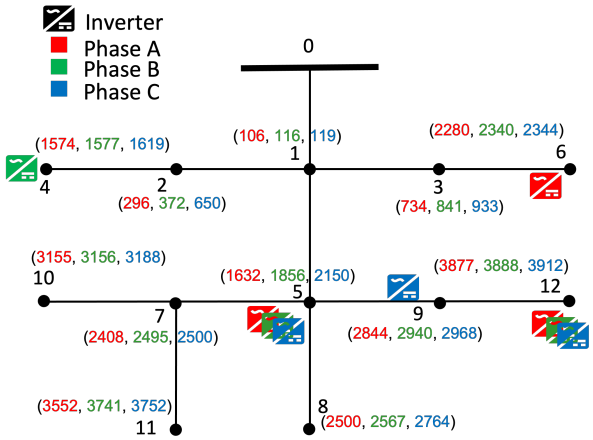

Fig. 6. Multiphase IEEE 13-bus distribution feeder.

TABLE II
INCREMENTAL VS. NON-INCREMENTAL VOLT/VAR CONTROL RULES ON THE MULTIPHASE IEEE 13-BUS FEEDER

Time	$\mathbf{q}=\mathbf{0}$	Non-incremental		Incremental	
	Obj. (p.u.)	Time (s)	Obj. (p.u.)	Time (s)	Obj. (p.u.)
1 pm	$2.51 \cdot 10^{-3}$	64.65	$1.15 \cdot 10^{-3}$	199.24	$4.11 \cdot 10^{-4}$
2 pm	$1.48 \cdot 10^{-3}$	66.60	$6.89 \cdot 10^{-4}$	209.92	$3.03 \cdot 10^{-4}$
3 pm	$6.89 \cdot 10^{-4}$	74.68	$4.94 \cdot 10^{-4}$	263.37	$2.16 \cdot 10^{-4}$
4 pm	$8.03 \cdot 10^{-4}$	68.32	$5.26 \cdot 10^{-4}$	126.81	$2.47 \cdot 10^{-4}$
5 pm	$5.51 \cdot 10^{-4}$	62.58	$4.11 \cdot 10^{-4}$	129.71	$1.95 \cdot 10^{-4}$

bus feeder, using the testing setup from [13]. Active loads were sampled 10 at a time from homes with IDs 20-379 and added to all three phases for the buses 1-12. Figure 6 also shows the solar panel assignments shown in Fig 6 for solar generation. Lastly, nine DERs with inverters were added across phases and bus indices as shown in Fig. 6. Learning rates for non-incremental and incremental RNNs were set as 0.1 and 0.001, respectively, with the design parameters $\mathbf{z} := [\bar{\mathbf{v}}; \boldsymbol{\delta}; \boldsymbol{\sigma}; \boldsymbol{\alpha}]$ initialized to feasible values $(0.95, 0.01, 0.3, 1.5)$. Table II compares the performance of the two rule types over multiple periods for $S = 80$. While incremental rules took longer time to train, they were successful in lowering cost $F(\mathbf{z})$ by more than 50%, thus yielding improved voltage profiles.

VI. CONCLUSIONS

A RNN-based ORD approach has been put forth for single- and multi-phase feeders. The key idea is to construct a RNN that emulates end-to-end the associated Volt/VAR dynamics. The RNN takes grid conditions as the input, the rule parameters as weights, and outputs the associated equilibrium voltages. Leveraging the convergence rates of the related optimization algorithms, we have provided bounds on the RNN depths to approximate equilibrium voltages within the desired accuracy. We have also established the stability of incremental control rules for multiphase feeders. Numerical tests have demonstrated that the designed control rules attain improved voltage profiles compared to their non-incremental alternatives. The improvement was found to be starker for mutiphase feeders, wherein (non)-incremental rules do not reach the same equilibrium. Our findings motivate further research to possibly characterize the equilibria of control rules for multiphase feeders; the convergence of accelerated incremental rules for multiphase feeders; and to deal with chance-constrained or ORD problems targeting phase imbalances.

REFERENCES

[1] *IEEE Standard for Interconnection and Interoperability of DERs with Associated Electric Power Systems Interfaces*, IEEE Std., 2018.
[2] M. Farivar, X. Zhou, and L. Chen, "Local voltage control in distribution systems: An incremental control algorithm," in *Proc. IEEE Intl. Conf. on Smart Grid Commun.*, Miami, FL, Nov. 2015.
[3] H. Zhu and H. J. Liu, "Fast local voltage control under limited reactive power: Optimality and stability analysis," *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 3794–3803, 2016.
[4] V. Kekatos, L. Zhang, G. B. Giannakis, and R. Baldick, "Voltage regulation algorithms for multiphase power distribution grids," *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 3913–3923, Sep. 2016.
[5] S. Bolognani, R. Carli, G. Cavraro, and S. Zampieri, "On the need for communication for voltage regulation of power distribution grids," *IEEE Trans. Control of Network Systems*, vol. 6, no. 3, pp. 1111–1123, 2019.
[6] M. Farivar, L. Chen, and S. Low, "Equilibrium and dynamics of local voltage control in distribution systems," in *Proc. IEEE Conf. on Decision and Control*, Florence, Italy, Dec. 2013, pp. 4329–4334.
[7] X. Zhou, M. Farivar, Z. Liu, L. Chen, and S. H. Low, "Reverse and forward engineering of local voltage control in distribution networks," *IEEE Trans. Autom. Contr.*, vol. 66, no. 3, pp. 1116–1128, 2021.
[8] A. Eggli, S. Karagiannopoulos, S. Bolognani, and G. Hug, "Stability analysis and design of local control schemes in active distribution grids," *IEEE Trans. Smart Grid*, vol. 36, no. 3, pp. 1900–1909, May 2021.
[9] V. Calderaro, G. Conio, V. Galdi, G. Massa, and A. Piccolo, "Optimal decentralized voltage control for distribution systems with inverter-based distributed generators," *IEEE Trans. Power Syst.*, vol. 29, no. 1, pp. 230–241, Jan. 2014.
[10] A. Samadi, R. Eriksson, L. Söder, B. G. Rawn, and J. C. Boemer, "Coordinated active power-dependent voltage regulation in distribution grids with PV systems," *IEEE Trans. Power Del.*, vol. 29, no. 3, pp. 1454–1464, Jun 2014.
[11] A. Singhal, V. Ajjarapu, J. Fuller, and J. Hansen, "Real-time local Volt/Var control under external disturbances with high PV penetration," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3849–3859, Jul. 2019.
[12] I. Murzakhanov, S. Gupta, S. Chatzivasileiadis, and V. Kekatos, "Optimal design of Volt/VAR control rules for inverter-interfaced distributed energy resources," *IEEE Trans. Smart Grid*, 2023, arxiv: 2210.12805.
[13] S. Gupta, S. Chatzivasileiadis, and V. Kekatos, "Deep learning for optimal Volt/VAR control using distributed energy resources," *IEEE Trans. Smart Grid*, 2023, arxiv: 2211.09557.
[14] W. Cui, J. Li, and B. Zhang, "Decentralized safe reinforcement learning for inverter-based voltage control," *Electric Power Systems Research*, vol. 211, p. 108609, 2022.
[15] Z. Yuan, G. Cavraro, M. K. Singh, and J. Cortes, "Learning provably stable local Volt/Var controllers for efficient network operation," *IEEE Trans. Power Syst.*, pp. 1–14, 2023.
[16] S. Karagiannopoulos, P. Aristidou, and G. Hug, "Data-driven local control design for active distribution grids using off-line optimal power flow and machine learning techniques," *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 6461–6471, 2019.
[17] W. Cui, Y. Jiang, and B. Zhang, "Reinforcement learning for optimal primary frequency control: A Lyapunov approach," *IEEE Trans. Power Syst.*, vol. 38, no. 2, pp. 1676–1688, Mar 2023.
[18] S. Taheri, M. Jalali, V. Kekatos, and L. Tong, "Fast probabilistic hosting capacity analysis for active distribution systems," *IEEE Trans. Smart Grid*, vol. 12, no. 3, pp. 2000–2012, May 2021.
[19] P. Jahangiri and D. C. Aliprantis, "Distributed volt/var control by pv inverters," *IEEE Trans. Power Syst.*, vol. 28, no. 3, pp. 3429–3439, Aug. 2013.
[20] G. Cavraro, S. Bolognani, R. Carli, and S. Zampieri, "The value of communication in the voltage regulation problem," in *Proc. IEEE Conf. on Decision and Control*, 2016, pp. 5781–5786.
[21] V. Kekatos, L. Zhang, G. B. Giannakis, and R. Baldick, "Accelerated localized voltage regulation in single-phase distribution grids," in *Proc. IEEE Intl. Conf. on Smart Grid Commun.*, Miami, FL, Nov. 2015.
[22] D. Chen, S. Iyengar, D. Irwin, and P. Shenoy, "Sunspot: Exposing the location of anonymous solar-powered homes," in *ACM Intl. Conf. on Systems for Energy-Efficient Built Environ.*, Palo Alto, CA, Nov. 2016.