Online PMU Missing Value Replacement via Event-Participation Decomposition

Brandon Foggo, *Member, IEEE,* Nanpeng Yu, *Senior Member, IEEE,*

***Abstract*—We introduce a new method for online Phasor Measurement Unit (PMU) missing value replacement. Our approach allows us to decompose PMU event responses into a non-dynamic component (denoted the participation factor) that can be inferred directly from the past and a dynamic component that can be inferred directly from all other PMUs (denoted the event strength). When missing values occur, we can use these two components, which do not rely on the missing index, to estimate the correct value. The method is extremely fast and can easily be used for online applications. Furthermore, extensive testing on real power system event data reveals that our approach achieves state-of-the-art performance in terms of Mean Absolute Percent Errors (MAPEs) for PMU data dropped during event periods. The method also yields an interpretable and simplified view of events for further analysis and applications. The method relies only on PMU data and does not take outside information such as network topology.**

***Index Terms*—Missing value replacement, data imputation, Phasor Measurement Unit, power system event.**

NOMENCLATURE

α	Hyper-parameter set.
β	Learning rate.
$.^{\dagger}$	Pseudo-inverse.
$\mathcal{N}$	Multivariate Gaussian distribution.
σ_1	Baseline noise standard deviation.
σ_2	Event noise standard deviation.
Θ	Parameter set.
θ	Laplace distribution width hyper-parameter.
θ_d	Temporal sparsity parameter
d	Disturbance value.
F	Frobenius norm.
F_S	Nominal system frequency.
f_t	N-dimensional time series of frequency data.
I_N	$N \times N$ Identity matrix.
j	PMU superscript index.
N	Number of PMUs.
O_r	Outlier removal algorithm.
q	Number of simultaneous events.
R	Sampling frequency.
r	Number of simultaneously occurring events.
t	Time subscript index.
u	ADMM auxiliary dual variable.
v	Participation factor.
X_t	N-dimensional time series of any non-frequency data.
y	Deviation from baseline.
z	ADMM auxiliary variable.

B. Foggo and N. Yu are with the Department of Electrical and Computer Engineering, University of California, Riverside, CA 92501 USA. E-mail: nyu@ece.ucr.edu.

I. INTRODUCTION

PHASOR Measurement Units (PMUs) measure electrical quantities on the power system using a common time source for synchronization, which greatly improves the situational awareness for the system operators. The gathering of high-frequency snapshots of local voltage and current phasors at the control center enables developing a wide range of data-driven power system management, control, and protection applications. The amount of PMU data collected by system operators worldwide has seen a gigantic increase over the last decade. In the United States alone, more than 2000 PMUs [1] have been installed in the transmission grids.

However, synchrophasor systems often have several data quality issues, which can be problematic for critical data-driven applications that depend on streaming PMU data. For instance, if someone were to use PMU data for power system event detection and classification, then, while missing data during normal periods of operation would be unproblematic, any data unreported during the event period could severely impact the quality of that task. In practice, failures in PMUs, phasor data concentrators (PDCs), and communication links could lead to missing PMU data [2].

Our goal is to improve upon the methods of missing value replacement for PMU data. Since our primary motivation is to enhance the performance of downstream applications, most of which are online, we will need our method of missing value replacement to be online as well. Furthermore, our interest is in improving missing value replacement algorithms that directly relate to power system events. We will thus focus only on online missing value replacement strategies for PMU data, and our method of analysis will focus heavily on data that is dropped during event periods.

Online missing value replacement has been studied before, mostly in machine learning literature. In fields where the missing values are less critical, simple methods such as nearest neighbors [3] or mean imputation [4] are typically chosen. Such methods can be improved by graph-structured data (when that graph is known) [5]. Using Generative Adversarial Networks to generate missing value replacements has also been proposed [6] [7]. For streaming data, low-rank matrix factorization [8] [9] [10] [11] and matrix completions [12] have become quite popular. For PMU data, in particular, streaming linear regressions on past observations have shown to be fruitful [13]. Techniques in PMU data processing have also tilted towards low rank tensor factorizations coupled with subspace selection/tracking [14], leading to the state of the art algorithm in the field denoted the *OnLine Algorithm for PMU Data Processing* (OLAP) [15], which was further tested in reference [16]. Two variants of OLAP exist - one spatial [15], and one temporal [17]. The spatial version works by primarily uses information from other PMUs, while the temporal version mostly uses information from the past. Our method is able to beat these two methods by using both of these sources of information simultaneously and equally.

Our proposed approach, denoted SPIKE-P, is inspired by the theoretical response of the system states in small-signal stability analysis, where the system response is given by a linear combination of various dynamic modes [18]. We develop a model, which we call the *Event-Participation Decomposition Model*. This model allows us to decompose event responses into a non-dynamic component that can be inferred directly from the past and a dynamic component that can be inferred from all non-missing data at the current timestamp. The non-dynamic component represents the amount of participation of PMUs have in various disturbances, which mimics the participation matrix derived from the modal matrix of the dynamic system. The dynamic component represents the magnitudes of various disturbances.

Our main contributions are as follows:

- We create a model for PMU missing data imputation, which takes simultaneous but independent advantage of temporal information and spatial information without knowledge of the electrical grid topology.
- We develop an algorithm for the inference of parameters in that model, which is extremely fast - fast enough to run in real-time.
- We show that missing data imputation under this model results in state-of-the-art performance in terms of Mean Absolute Percentage Error for missing values dropped during system event periods.
- We discuss the implications of the learned model parameters related to other problems in the field.

The remainder of the paper is organized as follows. Section II provides the formulation of the missing value replacement problem with the event-participation model. The parameter inference techniques are presented in Section III. Section IV shows the experimental results with a real-world, large-scale PMU dataset. The conclusions are stated in Section V.

II. Problem Formulation and Modeling

A. Problem Setup

Let N be the number of PMUs from which we receive streaming data. Then at each timestamp, sampled every $1/R$ seconds, we receive five N-dimensional vectors - one for voltage magnitude data, one for voltage angle data, one for current magnitude data, one for current angle data, and one for frequency data. Since we consider each data type independently from the others (other than an explicit relationship between frequency data and angular data), we will denote any of the first four of these vectors as X_t where t indexes time samples. When context on data type is necessary, we will make the distinction explicit. We will denote the vector of incoming frequencies as f_t. We will use superscripts to denote PMU indices, typically with the letter 'j'. Furthermore, when the time stamp is dropped from the data vector, i.e., when the symbol X is used alone, we refer to the entire time series. We will also index all data up until timestamp t via the notation $X_{:t}$.

We perform online missing value replacement via inference on dynamic generative models. Each model is parametric with parameter set Θ and hyper-parameter set α, giving a dynamic distribution which we will denote as $p(X_{:t}, f_{:t}, \Theta; \alpha)$. Parameters and missing values are estimated in two stages. In the first stage, we estimate parameters via maximum apriori estimation.

$$\Theta^* = \text{argmax } p(\Theta | X_{:t}^{\text{obs}}; f_{:t}^{obs}, \alpha) \tag{1}$$

Where the superscript 'obs' refers to all data that is non-missing. In the second stage, we use the estimated parameters and previously estimated missing values to estimate the new (incoming) missing PMU values, again through maximum likelihood.

$$X_t^*, f_t^* = \text{argmax } p(X_t^{-obs} | X_{:t}^{\text{obs}}, f_{:t}^{obs}, X_{:t-1}^*, f_{:t-1}^*, \Theta^*; \alpha) \tag{2}$$

Where the superscript '-obs' refers to missing data indices.

B. Normal Behavior Dynamic Model

We begin by describing a baseline generative model of the dynamics of X_t under normal operating conditions. Under these conditions, the baseline model considers all magnitude data to remain constant. On the other hand, Angular data changes according to deviations of the measured frequency at the PMU to the nominal system frequency F_S (e.g., $60Hz$ is the US, and $50Hz$ in the EU). These simple dynamics are described in the following equations, letting $\mathcal{N}(\mu; \Sigma)$ denote the multivariate Gaussian distribution with mean μ and covariance Σ.

$$\begin{aligned}
base(X_t) \quad &= \begin{cases} X_t, & \text{Magnitude Data} \\ X_t + \frac{360}{R}(f_t - F_S \mathbf{1}_N), & \text{Angle Data} \end{cases} \\
base(f_t) \quad &= f_t \\
\epsilon_t \quad &\sim \mathcal{N}(0; \sigma_1 I_N) \quad (\mathbb{R} \ni \sigma_1 > 0) \\
X_{t+1} \quad &= base(X_t) + \epsilon_t
\end{aligned} \tag{3}$$

Where R is the PMU sampling rate, I_N is the $N \times N$ identity matrix, and $\mathbf{1}_N$ is the N-dimensional vector containing all 1s. The $360/R$ term transforms the frequency deviation data into a per-timestamp angle update.

Inference in the baseline model is rather trivial. There are no parameters, and maximum likelihood estimation of missing values yields $X_t^* = base(X_{t-1}^*)$ (note that X^* is a mix of observed and estimated data).

C. The Event-Participation Model

The normal behavior dynamic model only describes PMU data under normal operating conditions. But many applications utilizing PMUs are designed specifically to extract information under abnormal operational conditions. As such, a model that considers disturbances that impact a large area of the system is necessary.

Critical to our disturbance model is the idea of decomposing events into a non-dynamic component that can be inferred directly from the past and a dynamic component that can be directly inferred from all of the non-missing data at the current timestamp.

As such, we write such disturbances as a product of factors $v^j d_t$, where v^j is an unchanging latent variable specific to the j^{th} PMU, while d_t is a dynamic component that is constant across PMUs. The intuition for this model, aside from being convenient for missing value replacement, is that deviations in dynamics from the baseline model during an event tend to have a scaled symmetry - all PMUs participating in the disturbance will have deviations with very similar shapes, but with different values of how much the disturbance effects each PMU. Thus, we can interpret the sequence d_t as the shape of the disturbance, and v^j as the amount of participation that the j^{th} PMU has in that disturbance. As such, we will call v^j the participation factor of the j^{th} PMU, and d_t as the event value at time t. We call this model the 'event-participation decomposition model' and describe it in the following equation block.

$$\begin{aligned} v &\sim \prod_j Laplace(0; \frac{1}{\theta}) \\ \eta &\sim \mathcal{N}(0; \sigma_2 I) \\ X_{t+1}^j &= base(X_t^j) + v^j d_t + \eta_t^j \end{aligned} \tag{4}$$

Of course, the idea of event-participation decomposition is agnostic to the prior placed on the latent participation vector v. We have chosen a Laplacian prior here to induce participation sparsity. This is because many types of events are known only to impact a limited number of PMUs significantly.

This model can be generalized to multiple disturbances occurring simultaneously by letting v be an $N \times r$ matrix and d an r-dimensional vector where r is the number of simultaneously occurring disturbances.

Letting y_t denote $X_t - base(X_{t-1})$, the joint probability distribution can be written as follows. The variables v and d are considered latent, while θ and σ_2 are considered hyper-parameters.

$$\begin{aligned} p(y_{:t}, v | d_{:t}; \alpha) &= \frac{1}{Z(\alpha)} exp\{-\frac{\|y_{:t} - v d_{:t}^T\|_F^2}{2\sigma_2^2} - \theta \sum_j \|v^j\|_1\} \\ Z(\alpha) &= \frac{2}{\theta}\sqrt{2\pi\sigma_2^2} \end{aligned} \tag{5}$$

Where F denotes the Frobenius norm.

III. Parameter Inference

Inference under the event-participation decomposition model is non-trivial. Maximization of the log-probability conditioned on the observed data yields the objective function given in the following equation block (ignoring terms that do not depend on the parameters),

$$\begin{aligned} \mathcal{L} &= -\frac{\|y_{:t} - v d_{:t}^T\|_F^2}{2\sigma_2^2} - g(v) \\ g(v) &= \theta \sum_j \|v^j\|_1, \end{aligned} \tag{6}$$

which has the form of a regularized low-rank matrix approximation. While online low-rank matrix approximation itself is well studied, regularized versions have seen significantly less development.

To begin describing our estimation algorithm, we first note that when v is fixed, the solution to d is just a projection. The derivation of this fact is included in the next equation block.

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial d_{:t}} &= \frac{1}{2\sigma_2^2}(-y_{:t}^T v + d v^T v) \\ \implies d_{:t}^*(v) &= y_{:t}^T v (v^T v)^{-1} \end{aligned} \tag{7}$$

In other words, $d_{:t}^T$ is the projection of $y_{:t}$ onto the vectors spanned by v. We can thus rewrite our objective function as given in the following equation:

$$\mathcal{L} = -\frac{\|y_{:t} - v v^\dagger y_{:t}\|_F^2}{2\sigma_2^2} - g(v) \tag{8}$$

Where $v^\dagger = (v^T v)^{-1} v^T$ is the pseudo-inverse of v. We can then find the regularized optimal v for this equation, and then use $d_{:t}^T = v^\dagger y_{:t}$ to find the shape of the disturbances.

A. Proximal Stochastic Implicit Krasulina Updates

To optimize objective function (8), we develop a proximal variant of the stochastic implicit Krasulina updates [19]. Let $v^{(k)}$ denote the k^{th} iteration value of v. Let $d_{:t}^{(k)T}$ denote $v^{(k)\dagger} y_{:t}$. Then the classic implicit Krasulina updates (to solve this objective without the regularization term) take the form:

$$v^{(k+1)} = \underset{\tilde{v}}{\text{argmin}} \frac{1}{2\beta}\|v^{(k)} - \tilde{v}\|_F^2 + \frac{1}{2}\|y_{:t} - \tilde{v} d^{(k)T}\|_F^2 \tag{9}$$

where $\beta > 0$ is a learning rate.

In the case where $y_{:t}$ is a column vector, the solution to this minimization problem can be written nicely as:

$$v^{(k+1)} = v^{(k)} - \frac{\beta}{1 + \beta \|d^{(k)T}\|^2}(v^{(k)} d_{:t}^{(k)T} - y_{:t}) d_{:t}^{(k)} \tag{10}$$

(Note that $d_{:t}^{(k)T}$ is just a column vector here). In this case, the update equation is known as the *stochastic* implicit Krasulina update.

The stochastic implicit Krasulina updates are a way of performing online matrix decomposition. But they alone do not solve the regularized form. However, they can be generalized to a method that *does* solve the regularized form. To see this, note that the the Krasulina update can be written as a proximal operator update step coupled with a standard minimization step in the attempt to solve $\text{argmin}_{v,d}\ f(v,d)$ (where $f = \|y_{:t} - vd^T\|_F^2$) via fixed points of v and d:

$$\begin{aligned} d^{(k+1)T} &= \text{argmin}_d\ f(v^{(k)}, d) \\ v^{(k+1)} &= Prox_{\beta f(\cdot,d)}\left(v^{(k)}\right) \end{aligned} \tag{11}$$

where $prox_{\beta f}$ maps v to $argmin_{\tilde{v}} \frac{1}{2}\|v - \tilde{v}\|^2 + \beta f(\tilde{v})$. The second step in this iteration, the proximal operator step, is called the *method of multipliers* update to the problem $\text{argmin}_v f(v,d)$. This step works, as our goal is to obtain $0 \in \partial f(v)$ (for optimality), while the proximal operator gives us a value in $v - \tilde{v} \in \partial f(\tilde{v})$. After several iterations, this set of equations will converge to a fixed point v^*, giving us $0 = v^* - v^* \in \partial f(v^*)$.

We can continue in this vein to solve the optimization problem $\text{argmin}_{\tilde{v}}\ f(\tilde{v},d) + \frac{1}{2\beta}\|v^{(k)} - \tilde{v}\|_F^2 + g(\tilde{v})$ by augmenting the method of multipliers step into an *alternating direction method of multipliers* step [20]. To do so, we transform the unconstrained optimization $\text{argmin}_{\tilde{v}} f(\tilde{v},d) + \frac{1}{2\beta}\|v - \tilde{v}\|_F^2 + g(\tilde{v})$ problem into a constrained one as follows:

$$\begin{aligned} &\text{argmin}_{\tilde{v}} f(\tilde{v},d) + \frac{1}{2\beta}\|v - \tilde{v}\|_F^2 + g(z) \\ &\text{subject to } z = \tilde{v} \end{aligned} \tag{12}$$

We have introduced a dummy variable z and constrained it to be equal to $\tilde{v}$. Next, we introduce a new variable u, associated with the Lagrange multiplier of that constraint. We then have the augmented lagrangian given by the following equation.

$$\mathcal{L} = f(\tilde{v},d) + Tr(u^T(z - \tilde{v})) + \frac{1}{2\beta}\|v - \tilde{v}\|_F^2 + \frac{1}{2\beta}\|\tilde{v} - z\|_F^2 + g(z) \tag{13}$$

We have used the same parameter β for the constraint penalty and the stochastic update penalty.

Finally, we minimize $\mathcal{L}$ with dual ascent via the following update equations:

$$\begin{aligned} d^{(k+1)T} &= \text{argmin}_d\ f(v^{(k)}, d) \\ v^{(k+1)} &= Prox_{\frac{\beta}{2} f(\cdot,d)}\ \left(z^{(k)} - \frac{1}{2}(v^{(k)} + u^{(k)})\right) \\ z^{(k+1)} &= Prox_{\beta g(\cdot)}\ \left(v^{(k)} + u^{(k)}\right) \\ u^{(k+1)} &= u^{(k)} + \beta\left(z^{(k+1)} - v^{(k+1)}\right) \end{aligned} \tag{14}$$

Where $Prox_{\beta g(\cdot)}$ maps z to $argmin_{\tilde{z}} \frac{1}{2}\|z - \tilde{z}\|^2 + \beta g(\tilde{z})$.

The argument that this set of equations yields an optimal solution is similar to the argument given for the equations in (11). Again, our goal is $0 \in \partial f(v) + \partial g(v)$. The second update equation gives us $z - u - \tilde{z} \in \partial f(\tilde{z})$. The third gives $v + u - \tilde{v} \in \partial g(\tilde{v})$. Fixed point updates of these equations have u, z, and v converge to fixed points u^*, z^*, and v^*, in which case we get from the third equation that $z^* = v^*$. We then get from the second equation that $-u^* \in \partial f(v^*)$, and from the third that $u^* \in \partial g(v^*)$. Adding these together yields the result.

In the stochastic case, $Prox_{\beta f(\cdot,d)}\ \left(z^{(k)} - \frac{1}{2}(v^{(k)} + u^{(k)})\right)$ still has a very similar form to the stochastic implicit Krasulina update equation, and $Prox_{\beta g(\cdot)}$ is the shrinkage function shown in the second equation of the following equation block.

$$\begin{aligned} v^{(k+1)} &= q^{(k)} - \frac{\beta}{1 + \beta\|d^{(k)T}\|^2}(q^{(k)}d^{(k)T} - y_{:t})d^{(k)} \\ &\qquad (q^{(k)} = z^{(k)} - \frac{1}{2}(v^{(k)} + u^{(k)})) \\ z_{ij}^{(k+1)} &= \begin{cases} v_{ij}^{(k)} + u_{ij}^{(k)} - \beta\theta, & v_{ij}^{(k)} + u_{ij}^{(k)} > \beta\theta \\ v_{ij}^{(k)} + u_{ij}^{(k)} + \beta\theta, & v_{ij}^{(k)} + u_{ij}^{(k)} < -\beta\theta \\ 0, & \text{else} \end{cases}, \end{aligned} \tag{15}$$

We call these update equations the *proximal stochastic implicit Krasulina updates*.

B. Denoising

We want to guard against the possibility of non-event data heavily influencing the learned values in d^T (and, as a result, the learned values in v). As it currently stands, a PMU whose behavior significantly deviates from the baseline simply due to noise may have such an effect. We can deal with this by slightly altering the update equations in block (14).

The third update equation in the proximal stochastic implicit Krasulina updates, $z^{(k+1)} = Prox_{\beta g(\cdot)}\ \left(v^{(k)} + u^{(k)}\right)$, can be expanded upon quite generally. First note that this equation is the solution to the optimization problem given in the following equation block.

$$\underset{\tilde{z} \in \mathbb{R}^N}{\text{argmin}}\ e^{-\frac{1}{2\beta}\|v^{(k)} + u^{(k)} - \tilde{z}\|_F^2} e^{-g(\tilde{z})} \tag{16}$$

which is the maximum likelihood solution to the estimation of $v^{(k)} + u^{(k)}$ under the corruption of Gaussian noise with a prior distribution given by $e^{-g(v^{(k)} + u^{(k)})}$.

Reference [21] proposed that we may replace this proximal update, which is fundamentally a denoising step, with any denoising algorithm of our choice - regardless of whether or not that algorithm corresponds to an actual prior distribution. In that reference, the authors chose to use standard image denoising algorithms instead of a proximal update corresponding to an explicit prior. However, in our case, we do want the sparsity induced by the exponential prior - we just *also* want our v vector to not have any ridiculously high values corrupting our results. Thus, we will keep the original proximal operator of this update step but *compose* it with an outlier removal algorithm. More explicitly, we will replace the third update step with the equation

$$z^{(k+1)} = O_r \circ Prox_{\beta g(\cdot)}\ \left(v^{(k)} + u^{(k)}\right) \tag{17}$$

Where O_r refers to an outlier removal algorithm.

Our outlier removal algorithm of choice will be straightforward. If v (after shrinkage operation) has less than q nonzero entries, all nonzero entries are considered noise and set to zero. The reasoning behind such a simple outlier removal algorithm is that the effect of outliers on our parameter estimation is most significant when there is no event occurring; taking outliers into account during these non-event periods would force d to be nonzero, which is undesirable. Since events tend to involve several PMUs simultaneously, the sparsity of any column in the v matrix can be used to infer whether or not an event is occurring. If that column is too sparse, hence no event, we can safely remove the values in the column that do exist and deem them outliers. In our experiments, we set the threshold q to be 5% of the number of all PMUs.

C. Zero-Initialization and Kick-starting

For our parameters to be interpretable, we ideally want both v and d to be zero during non-event periods. However, this is a bit of a problem for the alternating minimization of the objective function $\|y - vd\|_F^2 + g(v)$. Particularly, the optimal solution to this objective, conditioned on $v = 0$, is $d = 0$, and visa-versa. Thus the parameter estimation algorithm above cannot escape this zero-zero state. To combat this, we modify the algorithm such that, when $v = 0$, we attempt to kick-start the parameter estimation out of this state every time new data is received.

Kick-starting occurs when a new data column comes in and consists first of checking if z is zero. If z is indeed zero, then we set d to the average absolute value of y, $d \leftarrow \mathbb{E}_{nm}\left[|y|\right]$ where $\mathbb{E}_{nm}$ is the sample expectation over the non-missing data at the new timestamp. The first column of v is then set to y/d, and the algorithm proceeds normally. However, if z is *still* zero after the algorithm iterates for this timestamp, then d is set back to zero, and a new kick-start will be attempted at the next data reception.

When data is missing at the time instance of a successful kick-start, we set the v values of those missing indices to the median value of the corresponding column of the v values that we have estimated during the kick-start.

D. Warm Up and Escaping

Also, for parameter interpretation purposes, we want v, when it is nonzero, to change rather slowly. This means that we need to have a pretty good value of v before receiving the next data point at the kick-start time of an event. For this purpose, we use more iterations than usual during, and only during, the kick-start. Our experiments will use 20 iterations for the kick-starts and just 1 iteration for every other time instance. However, this means that all non-event periods use many iterations (since all non-event periods attempt a kick-start), which would slow the algorithm down significantly. We can deal with this by exiting the iterations (and setting $d = 0$) whenever $z = 0$ at the end of one of them.

E. Temporal Sparsity, Event Clearing, Static Participation, and Median Slope Regression

We also use the shrinkage operator on d [the second equation in equation block (15)]. Every time d is computed, we apply shrinkage with a separate parameter θ_d substituting the original $\beta\theta$ terms. This allows for better temporal sparsity. Furthermore, if d is ever zero for two iterations in a row, we consider the event to be cleared and set all parameters (v, z, and u) back to zero.

The updates (14) are only computed during the kick-start. After a successful kick-start, the participation factors are held constant, and only d is updated. To avoid the case of a small number of noisy outliers continuing the event beyond its true timespan, we estimate d during these periods as the median slope estimate over PMUs with nonzero participation factors. That is, during event continuation periods, we estimate d via the following equation.

$$d = \mathbb{M}\mathrm{edian}\left(y/z \mid |z| > 0\right) \tag{18}$$

This can be viewed as a slight simplification to the Theil-Sen estimator [22].

F. Summary / Organization of Submodules

The organization of these SPIKE-P submodules can be summarized as follows: 1) Initialize all variables to zero. 2) Receive a new data vector. 3) Attempt kick-start via iterations of (14) coupled with denoising; escape on failure, mark event=True if successful. If event is True: 4) Receive more data. Estimate d via median filtering. 5) Check for event clearing; mark event=False if cleared. Data is replaced at each retrieval of new data.

IV. Experiments

A. Comparison Methods

1) OLAP: OLAP, short for 'OnLine Algorithm for PMU data processing' [15], may be considered state of the art in PMU missing value replacement literature. At its core, OLAP is a Singular Value Decomposition (SVD) based approach that relies on the assumption that a streaming window of vector PMU data will form a low-rank matrix. At each new data retrieval, SVD is performed, and all but the first few singular vectors are dropped. The remaining right singular vectors can then be used on the observed data to obtain a linear combination of left singular vectors that best fit those observations. We can fill in the unobserved data with the corresponding values in that linear combination. OLAP also features an event-detection submodule, making it quite useful in practice. However, this event detection scheme is irrelevant to performance comparisons on missing value replacement, so we will not cover it here. OLAP features a few hyperparameters - most importantly, the number of retained singular values and the size of the rolling window of data. We have found that using both a window size of one and just one singular vector yields the best trade-off of performance vs. time through experimentation. These hyper-parameters also ensure that OLAP and SPIKE-P have very similar computation times. Thus these are the hyperparameters that we will use for comparison against SPIKE-P.

2) Ensemble of Correlation Predictors: This is a method in which the correlation (in terms of deviations from the baseline model) between every pair of PMUs is kept track of. We will denote the correlation between PMUs i and j as ρ_{ij}. Each PMU with existing data is then an estimator for the data at the PMUs with missing data. For example, if the PMU with index i has missing data, we can approximate $X_i - X_{i-1} \approx \rho_{ij}(X_j - X_{j-1})$. Averaging over all PMUs with observed data then yields the *Ensemble of Correlation Predictors*. This is quite similar to the method proposed in reference [13].

B. Experimental Setup

Our dataset consists of two years of real data from hundreds of PMUs and over one thousand event labels of different types in the Eastern Interconnection of the U.S. power transmission grid. All data reported by the PMUs are positive sequence values. The dataset contains about 1000 labeled events with the exact timing of that event available. Events are broadly classified as Bus, Generator, Line, Oscillation, and Transformer events.

For each event in our dataset, we gathered 2 seconds of data with our best estimation of the event-timing set to the middle of the window. The tested algorithms do not see this entire two-second window at once. They see the first timestamp, then the second, and so on, simulating online processing. At the event's peak, we drop all data from 10% of the participating PMUs for 5 time instances (one-sixth of a second). We apply our proposed SPIKE-P and comparison methods to estimate and replace the missing values from the streaming PMU dataset. The mean average percent error (MAPE) is then collected over these dropped and estimated data points. The average MAPE over the PMUs is calculated for each event. Since the number of events is on the order of one thousand, we further averaged these reported MAPE values across the different event labels, which we report in Tables I-VI. The hyper-parameters for SPIKE-P used in this experimental are as follows: $\beta = 1.0$, $\theta = 0.01$, $\theta_d = 0.1$.

For each method tested, each data type is considered independently. Thus missing values in voltage magnitude data, for example, do not effect the performance of missing value replacement on current magnitude data.

C. Numerical Results

Results of our large-scale test, covering every event in our dataset, can be seen in Tables I-VI. All of these methods are excellent on normal data (see Table I). SPIKE-P and Baseline are, of course, identical on such data, as the former reduces to the latter when no kickoff is ever successful. EnCorr is often equivalent to baseline in these cases since the estimator learns its deviation from baseline as an average over a bunch of uncorrelated errors from the other PMUs, which will average to very near zero deviation. OLAP performs worse during normal data than these other methods, especially in missing value replacement for reactive power.

For Bus Events (Table II), SPIKE-P has a clear performance advantage over all other methods, seeing substantially lower errors. OLAP is a distant second here. A similar result occurs for Oscillation Events (Table V) and Transformer Events (Table VI). On Generator Events (Table III) and Line Events (Table IV), SPIKE-P performs best on current magnitude data, real power data, and reactive power data. However, OLAP takes a slide lead in terms of voltage magnitude performance, with SPIKE-P following closely. Both SPIKE-P and OLAP achieve fantastic results on voltage magnitude data for these types of events. On a final note, we see that voltage magnitude data does not appear to participate in our available Oscillation Event data (Table V) and appears to act as it would during normal data periods - thus, the results in this row mirror the results in the corresponding row of the Normal Period table (Table I) closely.

We would also like to note that all of these methods work quite well on voltage data for every event. As past literature on missing value recovery has mostly focused on this data type, this is not surprising. However, SPIKE-P achieves roughly the same accuracy as these other methods on voltage magnitude, which is significantly better than those on the other data types.

The computation times that we report in all of these tables is the time it took to perform missing value replacement for the entire 2 second event window (60 samples) for all four data variables. Thus times that are reported on the order of 0.01 seconds is equivalent to about 200 microseconds per iteration. The computational complexity of the baseline estimator, OLAP, and SPIKE-P are all $O(N)$ where N is the number of PMUs. All three of these are extremely suitable for online use. The computational complexity of EnCorr is worse, as it requires $O(N^2)$ computations. Note that this is specific to the rank-1 version of OLAP that we used here. Other implementations of OLAP (i.e., implementations using a higher rank and window size) have worse computation times.

We have also plotted several samples of event-participation decomposition values and some plots of the values that SPIKE-P is replacing. In the lefthand column of Figure 1, we took one of our bus events and plotted the time-differenced current magnitude values, the event-strength time-series found by SPIKE-P, and the participation factors found by SPIKE-P. We see that SPIKE-P has indeed learned the inherent properties of this event, as the shape of the event-strength time-series very closely outlines what we would expect from seeing the time differenced pure values. Similarly, we can see that the corresponding participation factors have captured the relative strength of the event under each PMU. The righthand column of the same figure shows the same set of learned values for the voltage magnitude data over the same event. A similar analysis shows the efficacy of this method for voltage magnitude. In Figure 2, we show a sample of the true (non-differenced) values of each variable alongside the values SPIKE-P replaced them with when they were dropped at the event's peak. For voltage magnitude, real power, and reactive power, the replaced values are extremely close to the true values. The current magnitude performs a little worse for this event, but the results are still good. This figure, as well as Figure 4, are labelled with anonymized PMU identification numbers, so, for example, the line labelled True (id: 1) corresponds to the line labelled Replaced (id: 1). The PMU identification numbers are unique to each figure. Unfortunately, we are not authorized to provide non-anonymized versions of these PMU IDs. Figures 3 - 4 show the same plots as figures 1 - 2, but for a Generator Event Sample. Again, the event-strength time series closely captures what we would expect of the event. The participation factors closely match the relative level that each PMU participates in this event. Furthermore, the replaced values are extremely close to the true ones for all plotted data types here (current magnitude, voltage magnitude, and real

NORMAL	Base	EnCorr	OLAP	SPIKE-P
Time (s)	0.006	0.143	0.01	0.006
VM (%)	0.550	0.550	0.733	0.550
IM (%)	2.557	2.557	3.538	2.557
P (%)	1.379	1.434	2.032	1.379
Q (%)	3.351	3.351	30.597	3.351

TABLE I: Average MAPEs over Normal Period Tests.

BUS	Base	EnCorr	OLAP	SPIKE-P
Time (s)	**0.005**	0.303	0.067	0.038
VM (%)	2.449	2.449	1.309	**1.082**
IM (%)	68.792	56.531	21.393	**18.995**
P (%)	27.413	22.692	7.888	**7.225**
Q (%)	28.594	24.442	17.580	**10.937**

TABLE II: Average MAPEs over Bus Event data.

power; this particular event had no apparent effect on reactive power).

Finally, Figure 5 shows how SPIKE-P performs when one event closely follows another. Indeed, SPIKE-P can infer between these events that the first event has cleared and can correctly capture the relevant variables for the second event.

The above experiments cover short term data drops occurring at the beginning of an event and lasting a few timestamps. Since most events are very short, this set of experiments covers the vast majority of real cases. However, some types of events, like oscillation events, are more prolonged than the others. We thus provide another experiment to cover these cases. For this new experiment, we have taken all of our oscillation events, and dropped PMUs randomly (at random starting times) throughout their duration for an entire two-thirds of a second (20 samples). Resulting MAPEs are averaged across the events and the data types and summarized in Table VII. Only SPIKE-P shows significant improvement over the Baseline model under this scenario.

One weakness inherent to every method of online missing value replacement for PMU data is that at least one datapoint of the event must be observed *before* the datapoint we are attempting to replace. This means that if any PMU were to have missing data for an entire event, no method would have good accuracy at replacing the data of that PMU. This

GEN	Base	EnCorr	OLAP	SPIKE-P
Time (s)	**0.006**	0.212	0.008	0.016
VM (%)	3.066	3.066	**1.853**	2.082
IM (%)	11.500	9.876	7.532	**6.507**
P (%)	7.894	7.128	6.790	**5.942**
Q (%)	7.740	7.028	6.867	**6.006**

TABLE III: Average MAPEs over Generator Event data.

LINE	Base	EnCorr	OLAP	SPIKE-P
Time (s)	**0.003**	0.123	0.004	0.012
VM (%)	1.664	1.664	**1.057**	1.340
IM (%)	11.683	11.361	11.074	**10.389**
P (%)	9.171	9.049	8.855	**8.678**
Q (%)	9.399	9.274	9.103	**8.946**

TABLE IV: Average MAPEs over Line Event data.

OSC	Base	EnCorr	OLAP	SPIKE-P
Time (s)	**0.004**	0.302	0.052	0.049
VM (%)	**1.160**	**1.160**	1.391	**1.160**
IM (%)	4.885	4.884	7.944	**2.710**
P (%)	4.899	4.898	7.978	**2.712**
Q (%)	4.816	4.815	7.881	**2.705**

TABLE V: Average MAPEs over Oscillation Event data.

TRANS	Base	EnCorr	OLAP	SPIKE-P
Time (s)	**0.004**	0.235	0.013	0.016
VM (%)	1.213	1.213	0.936	**0.258**
IM (%)	5.826	5.485	6.246	**4.632**
P (%)	5.212	4.872	5.807	**4.104**
Q (%)	5.287	4.947	5.867	**4.125**

TABLE VI: Average MAPEs over Transformer Event data.

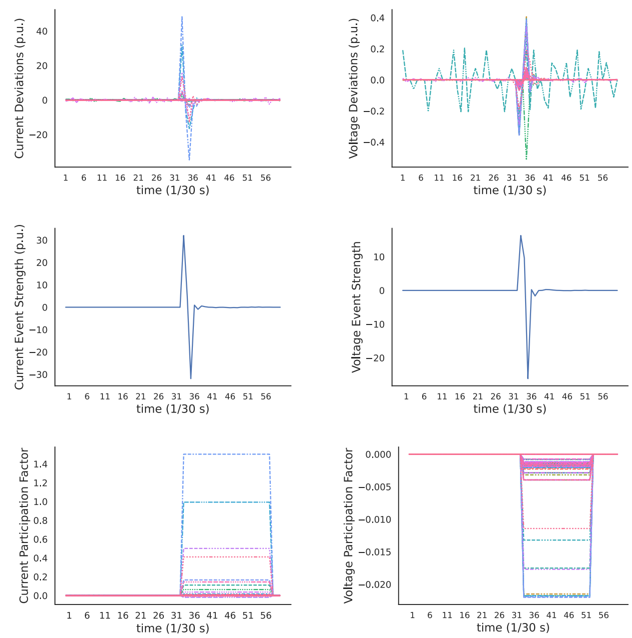


Fig. 1: Bus event sample decomposition

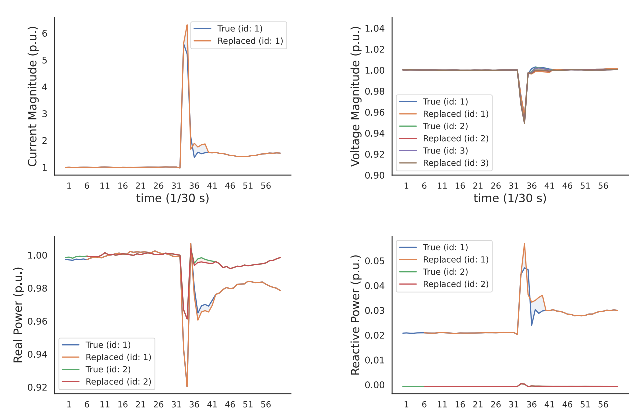


Fig. 2: SPIKE-P replacement on a Bus Event sample.

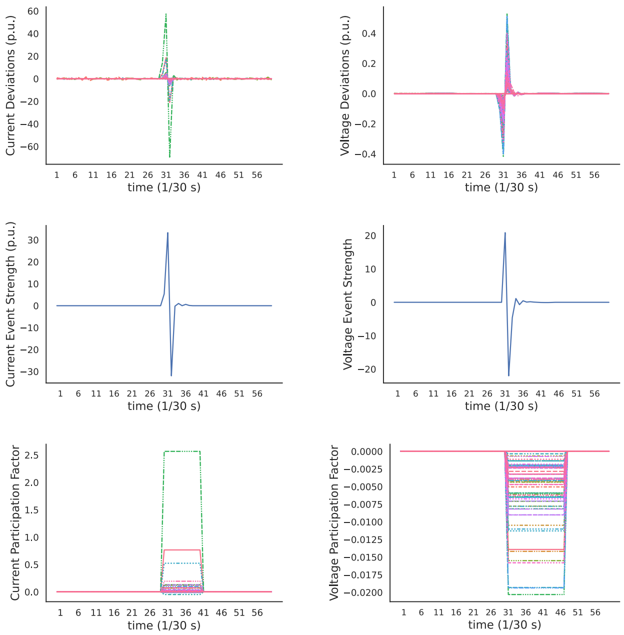


Fig. 3: Generator Event Sample Decomposition

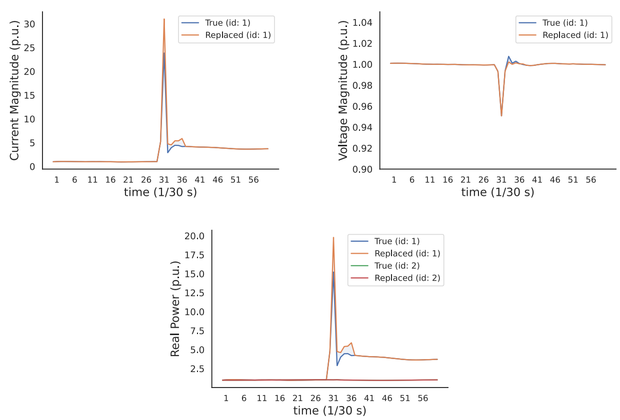


Fig. 4: SPIKE-P replacement on a Generator Event Sample.

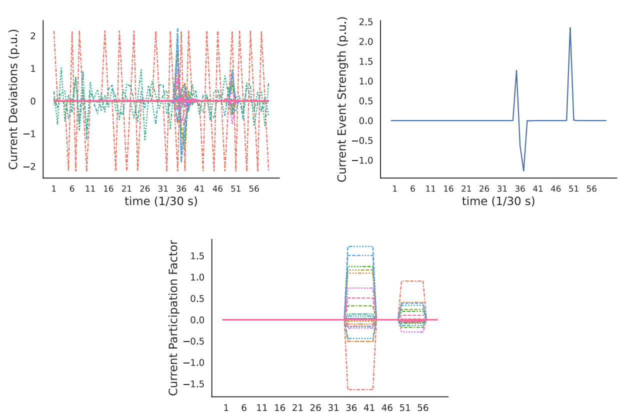


Fig. 5: Line Event Sample Decomposition, Current Magnitude. In this sample, a second event closely follows the first.

LONG	Base	EnCorr	OLAP	SPIKE-P
Time (s)	**0.004**	0.235	0.013	0.023
Average (%)	5.20	5.10	5.40	**3.41**

TABLE VII: Average MAPEs over long term drops on Oscillation Events.

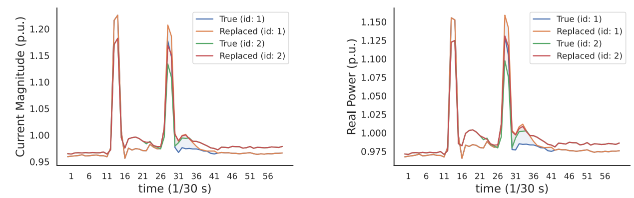


Fig. 6: SPIKE-P replacement on an event with *all* data dropped using only the participation factors of a nearby event.

weakness is present in our method as well, but there is a workaround - we can take the participation factors of a nearby event (temporally) and use those as a guess of the participation factors for any PMU that doesn't have data at the start of the event. To demonstrate this, we took one of our windows which happened to have two events occur within its time-span. We then estimated participation factors for the first event, and dropped *all* data for 2 randomly selected participating PMUs for the second event. Using only the participation factors of the first event, the algorithm still only falters by less than 2.5% MAPE over the completely missing data of the second event. This results of this experiment are visualized in Figure 6.

D. Discussion

Aside from imputing missing data, SPIKE-P has the potential to drive solutions to other problems. Since the event-participation decomposition, taken from SPIKE-P as z and d (generally as vector time-series), we obtain a simple view of the event, which is quite interpretable. The resulting values of this decomposition can be used for other purposes. For example, the event strength could be used as a highly informative engineered feature for event classification, or we could use SPIKE-P for data augmentation. Data augmentation refers to a popular scheme of generating fake data points by perturbing existing data and often helps fill out the space of inputs needed to train machine learning algorithms. The use of such techniques is widespread in machine learning solutions for other domains. SPIKE-P can be used to generate augmented PMU data spanning over a system event by taking the decomposition of the original data, slightly perturbing the participation factors and event strengths, and then re-integrating those perturbed decomposition variables back into the original time series.

To improve upon this method, future work will focus on using a more specialized event detection scheme rather than the one that is baked into the SPIKE-P estimation steps (with the kickstart essentially acting as an event detection scheme).

V. Conclusion

We introduce a new method for online Phasor Measurement Unit (PMU) missing value replacement. The approach is denoted Stochastic Implicit Krasulina Event-Participation Decomposition (SPIKE-P). It decomposes PMU event responses into a dynamic component capturing the essence of the event strength and a static component capturing each PMU's participation in that event. When missing values occur, we can use these two components, which do not rely on the missing index, to estimate the correct value. Critically, we used this decomposition on data derived as the deviation from a baseline model, which considered magnitude data to be constant and angular data to follow predictably to its frequency data. Extensive testing on real power system event data showed that our approach achieved state-of-the-art performance in terms of Mean Absolute Percent Errors (MAPEs) for PMUs dropped during event periods and that the method was easily fast enough to be performed online.

Acknowledgment

This material is based upon work supported by the Department of Energy under Award Number DE-OE0000916.

References

[1] A. G. Phadke and T. Bi, "Phasor measurement units, WAMS, and their applications in protection and control of power systems," *Journal of Modern Power Systems and Clean Energy*, vol. 6, no. 4, pp. 619–629, Jul. 2018.

[2] C. Huang, F. Li, D. Zhou, J. Guo, Z. Pan, Y. Liu, and Y. Liu, "Data quality issues for synchrophasor applications part I: a review," *Journal of Modern Power Systems and Clean Energy*, vol. 4, no. 3, pp. 342–352, 2016.

[3] S. Faisal and G. Tutz, "Nearest neighbor imputation for categorical data by weighting of attributes," *arXiv preprint arXiv:1710.01011*, 2017.

[4] A. R. T. Donders, G. J. Van Der Heijden, T. Stijnen, and K. G. Moons, "A gentle introduction to imputation of missing values," *Journal of clinical epidemiology*, vol. 59, no. 10, pp. 1087–1091, 2006.

[5] X. Yao, Y. Gao, D. Zhu, E. Manley, J. Wang, and Y. Liu, "Spatial origin-destination flow imputation using graph convolutional networks," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2020.

[6] I. Spinelli, S. Scardapane, and A. Uncini, "Missing data imputation with adversarially-trained graph convolutional networks," *Neural Networks*, vol. 129, pp. 249–260, Sept. 2020.

[7] A. Madapu, S. Segarra, S. P. Chepuri, and A. G. Marques, "Generative adversarial networks for graph data imputation from signed observations," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 9085–9089.

[8] M. Mardani, G. Mateos, and G. B. Giannakis, "Subspace learning and imputation for streaming big data matrices and tensors," *IEEE Transactions on Signal Processing*, vol. 63, no. 10, pp. 2663–2677, 2015.

[9] ——, "Imputation of streaming low-rank tensor data," in *2014 IEEE 8th Sensor Array and Multichannel Signal Processing Workshop (SAM)*. IEEE, 2014, pp. 433–436.

[10] J. Vinagre, A. M. Jorge, and J. Gama, "Fast incremental matrix factorization for recommendation with positive-only feedback," in *International Conference on User Modeling, Adaptation, and Personalization*. Springer, 2014, pp. 459–470.

[11] D. Osipov and J. H. Chow, "PMU missing data recovery using tensor decomposition," *IEEE Transactions on Power Systems*, vol. 35, no. 6, pp. 4554–4563, Nov. 2020.

[12] M. Liao, D. Shi, Z. Yu, W. Zhu, Z. Wang, and Y. Xiang, "Estimate the lost phasor measurement unit data using alternating direction multipliers method," in *2018 IEEE/PES Transmission and Distribution Conference and Exposition (T&D)*. IEEE, 2018, pp. 1–9.

[13] N. T. Le and W. Benjapolakul, "A data imputation model in phasor measurement units based on bagged averaging of multiple linear regression," *IEEE Access*, vol. 6, pp. 39 324–39 333, 2018.

[14] K. Chatterjee, K. Mahapatra, and N. R. Chaudhuri, "Robust recovery of PMU signals with outlier characterization and stochastic subspace selection," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3346–3358, Jul. 2020.

[15] P. Gao, M. Wang, S. G. Ghiocel, J. H. Chow, B. Fardanesh, and G. Stefopoulos, "Missing data recovery by exploiting low-dimensionality in power system synchrophasor measurements," *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1006–1013, 2015.

[16] G. M. De Mijolla, S. Konstantinopoulos, P. Gao, J. H. Chow, and M. Wang, "An evaluation of algorithms for synchrophasor missing data recovery," in *2018 Power Systems Computation Conference (PSCC)*. IEEE, 2018, pp. 1–6.

[17] S. Konstantinopoulos, G. M. De Mijolla, J. H. Chow, H. Lev-Ari, and M. Wang, "Synchrophasor missing data recovery via data-driven filtering," *IEEE Transactions on Smart Grid*, vol. 11, no. 5, pp. 4321–4330, Sept. 2020.

[18] P. Kundur, N. J. Balu, and M. G. Lauby, *Power system stability and control*. McGraw-hill New York, 1994, vol. 7.

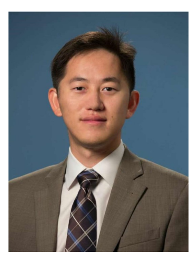
[19] E. Amid and M. K. Warmuth, "An implicit form of Krasulina's k-PCA update without the orthonormality constraint," *arXiv preprint arXiv:1909.04803*, 2019.

[20] S. Boyd, N. Parikh, and E. Chu, *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[21] T. Meinhardt, M. Moller, C. Hazirbas, and D. Cremers, "Learning proximal operators: Using denoising networks for regularizing inverse imaging problems," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1781–1790.

[22] P. K. Sen, "Estimates of the regression coefficient based on Kendall's tau," *Journal of the American statistical association*, vol. 63, no. 324, pp. 1379–1389, 1968.

Brandon Foggo (M'19) received the B.S. degree in Electrical Engineering from the University of California, Los Angeles, in 2015, and the Ph.D. degree in Electrical and Computer Engineering from the University of California, Riverside, in 2019. His research interests lie in statistical learning theory and information theory, particularly in their merging, as well as applications to cyber physical systems with an emphasis on power distribution systems.



Nanpeng Yu (M'11-SM'16) received his B.S. in Electrical Engineering from Tsinghua University, Beijing, China, in 2006. Dr. Yu also received his M.S. and Ph.D. degree in Electrical Engineering from Iowa State University, Ames, IA, USA in 2007 and 2010 respectively. He is currently an Associate Professor in the department of Electrical and Computer Engineering at University of California, Riverside, CA, USA. His current research interests include machine learning in smart grid, electricity market design and optimization, and smart energy communities. Dr. Yu is an Editor of IEEE Transactions on Smart Grid and IEEE Transactions on Sustainable Energy.