Identification of Critical Times for Distribution System Time Series Analysis

Nicholas S. Coleman, *Student Member, IEEE,* and Karen N. Miu, *Member, IEEE*

***Abstract*—Online power distribution system time series analyses are typically divided into uniform intervals (e.g., market-driven) ranging from minutes to hours. This work presents a constraint-driven approach to refining an initial time window structure at critical times within an injection forecast. Implicit temporal load capability (ITLC), which is formalized in this paper, uses forecasted nodal injection characteristics to identify critical times when control actions are necessary in order to maintain feasibility. ITLC and control selection algorithms are presented. Quasi-static time series and ITLC simulation results are presented using a multi-phase, 2556-node distribution test circuit. With ITLC, a subset of the initial time windows are sub-divided at the analytically-selected critical times. This approach is more efficient than using a large number of short time windows that span the entire forecast horizon.**

***Index Terms*—Load capability, power distribution systems, power system control, time series analysis**

I. Introduction

TRADITIONAL distribution system control schemes are based on seasonal load averages, with distributed device switching triggered by time-of-day set-points and/or local sensing. In smart distribution systems, advanced metering infrastructure (AMI) provides real-time nodal data and two-way communication, which enables online control.

As distribution control shifts towards operational time frames, the need for online power flow solutions or static state estimates remains. Quasi-static time series (QSTS) analysis, which refers to a time series of steady-state power flow solutions, provides a suitable structure for this type of problem [1]–[5]. With QSTS analysis, long-term dynamics are captured, while short-term dynamics are assumed stable and represented as algebraic states [6]. It is assumed that no large disturbances occur, thus preserving long-term dynamical relationships. QSTS has been used to investigate time-varying distribution system injections (e.g., photovoltaics) [1], [2].

Balancing data and computational requirements is challenging in time series analysis; it is important to select proper time steps [2]. Evaluation times or control intervals are typically selected in advance. This works wells with classical optimization tools such as dynamic programming [7]–[9] and model predictive control [10], [11]. These approaches, however, restrict the analysis to a set of arbitrary or predetermined control intervals.

Submitted on 21-Nov-2016; revised 10-Mar-2017. This work was supported in part by NSF DUE #1226126 and Lockheed Martin Grant URI. Paper no. TPWRS-01728-2016.

The authors are with the Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA 19104 USA (e-mail: coleman@drexel.edu; miu@coe.drexel.edu).

This work introduces an analytical, constraint-driven approach to time window selection for online distribution system analysis. Forecasted nodal injection behavior is used to identify "critical times" when control is anticipated or is necessary to maintain feasibility. Analytically-selected critical times:

- indicate when the state is on a constraint boundary.
- sub-divide initial forecast/control intervals.
- can be naturally embedded into QSTS frameworks.
- help focus computational needs into select time windows.

The proposed method leverages distribution load capability (LC) metrics (e.g., [12]–[14]; analogues in transmission systems include [15]–[17]) to identify critical injection conditions. Then, a time-varying forecast is used to map critical conditions to critical times. The method, termed "implicit temporal load capability" (ITLC) extends initial studies presented in [5] to include a priori control schedules, legacy devices, and switching dead bands. These considerations reflect practical control procedures and preserve the validity of QSTS approximation.

Contributions of this paper include:

- a constraint-driven critical time identification model.
- a formalized implicit temporal load capability model.
- step-by-step solution algorithms.
- detailed day-ahead simulation results for a 2556-node distribution system.

This paper is organized as follows: Section II reviews load capability and introduces the extension to ITLC. Section III develops the ITLC model. Section IV presents the problem formulation. The ITLC solution algorithm and control selection sub-algorithms are presented in Section V. Sections VI and VII present simulation results and conclusions, respectively.

II. Background

Implicit temporal load capability extends traditional load capability to the operational time frame. This section reviews distribution LC and introduces the transition to ITLC.

A. Load Capability Review

Load capability is an analytical tool used to identify maximum feasible loading conditions [18]. Distribution LC is often formulated with respect to operating constraints [12]–[14]). Consider an n-node injection vector varying according to:

$$S_{\text{final}} = S_{\text{init}} + \lambda \hat{S} \tag{1}$$

where

S_{init}, $S_{\text{final}} \in \mathbb{C}^n$ are initial and final injection levels,
$\hat{S} \in \mathbb{C}^n$ is an injection variation direction,
$\lambda \in \mathbb{R}$ is a unitless variation factor or loading factor.

The LC problem is then formulated as the following nonlinear optimization:

$$\text{maximize} \quad \lambda \tag{2}$$

$$\text{such that} \quad F(x, \lambda, u) = 0 \tag{3}$$

$$G(x, \lambda, u) \leq 0 \tag{4}$$

where x is the system state u is the (constant) control setting. Eq. (3) represents the nonlinear power flow equations; (4) represents the operating constraints, including bus voltage magnitude limits, branch current/kVA ratings, and/or power factor constraints. LC literature presents methods for solving (2)-(4) with respect to different specific constraint sets [12]–[17].

B. Extension to Implicit Temporal Load Capability

While load capability is traditionally applied in planning studies, implicit temporal load capability is an operating tool. In ITLC, the injection variation vector $\hat{S}$ is obtained from a time-varying injection forecast, and has units of complex power per unit time. Accordingly, λ takes on units of time. The solution to (2)-(4) is then a "critical time" corresponding to a critical loading condition for which the power flow solution lies on the feasible region boundary.

Solving this problem repeatedly as the injection variation direction changes throughout the forecast horizon gives rise to the critical time identification model presented in this work.

III. Implicit Temporal Load Capability Model

ITLC requires an injection forecast divided into K intervals, or "time windows". The number of time windows and duration of each may rely on forecasted power injection characteristics [3], scheduled control actions, or regulatory metrics, for example. These characteristic-driven windows provide a good initial structure for distribution automation and planning studies.

ITLC conducts a constraint-driven refinement of these initial time windows. Distribution LC is applied to identify critical injection levels. These are mapped to critical times that subdivide the initial time windows. ITLC returns a sequence of analytically selected non-uniform control intervals, or "subwindows", and a control setting for each resulting sub-window.

This section first presents an injection forecast model and the corresponding ITLC inputs. Next, critical time identification and preventative control steps (i.e., time-window refinement) are presented. Finally, the ITLC outputs are described alongside an illustrative example.

A. Injection Forecast Model

First, consider uncontrollable injections (e.g., loads or photovoltaics) with $S_{\text{D}}(t_k) \in \mathbb{C}^n$, $k = 1, ..., K+1$ as their time series forecast. Let $S_{\text{init}} = S_{\text{D}}(t_1)$. $K+1$ injection levels yield K time windows, numbered $k = 1, ..., K$. The duration of time window k is:

$$\tau_k = t_{k+1} - t_k \tag{5}$$

For each time window, a variation vector $\hat{S}_k \in \mathbb{C}^n$ can be specified directly or estimated from the initial time series. In ITLC, $\hat{S}_k$ is the time-derivative of the complex power injections during time window k. This is a key differentiator between ITLC and previous LC works such as [12], [13], [17].

In this work, each $\hat{S}_k$ is estimated as a constant vector using linear interpolation (higher-order approximations are possible):

$$\hat{S}_k = \hat{P}_k + j\hat{Q}_k = \frac{S_{\text{D}}(t_{k+1}) - S_{\text{D}}(t_k)}{t_{k+1} - t_k} \tag{6}$$

Now consider the controllable injections. $S_{\text{C}}(u_k) \in \mathbb{C}^n$ represents constant injections as function of control setting u_k during time window k. u_k is a vector of discretely-valued device settings (e.g., capacitor or battery banks), each of which may be subject to its own constraints. The forecasted net injection at time t_k is initially:

$$S(t_k) = S_{\text{D}}(t_k) + S_{\text{C}}(u_k) \tag{7}$$

B. ITLC Inputs

The ITLC inputs include S_{init}, and the following sequences which describe a net injection forecast:

$$\hat{S} = (\hat{S}_1, \hat{S}_2, ..., \hat{S}_K) \tag{8}$$

$$T = (\tau_1, \tau_2, ..., \tau_K) \tag{9}$$

$$U = (u_1, u_2, ..., u_K) \tag{10}$$

C. Critical Time Identification

Within each time window, LC metrics are used to identify critical operating points. These are mapped to critical times when action is required to maintain feasibility with respect to one or more constraints. Critical times sub-divide a time window into sub-windows. Time window k will contain $L_k \geq 1$ sub-windows, denoted $(k, 1), ..., (k, L_k)$. L_k is determined as part of the solution.

Define $t_{k,l}$ as the start time of sub-window (k, l) (thus $t_k = t_{k,1}$). Define $\lambda_{k,l}$ as the duration of sub-window (k, l). $\lambda_{k,l}$ is the maximum non-negative scalar such that:

1) $S(t_{k,l}) + \lambda_{k,l}\hat{S}_k$ yields a state x that satisfies (3)-(4), and
2) $t_{k,l} + \lambda_{k,l} \leq t_{k+1}$.

The first condition is satisfied by finding constraint-limited upper bounds on $\lambda_{k,l}$. These bounds are LC estimates computed with (2)-(4) .

The second condition preserves the initial time window structure, and is satisfied by imposing the following non-negative upper bound on $\lambda_{k,l}$:

$$\lambda^e_{k,l} = \tau_k - \sum_{s=1}^{l-1} \lambda_{k,s} = t_{k+1} - t_{k,l} \tag{11}$$

If no non-negative scalar satisfies these conditions, then $\lambda_{k,l} = 0$. Therefore, the duration of sub-window (k, l) is:

$$\lambda_{k,l} = \max\{\min\{\lambda^V_{k,l}, \lambda^I_{k,l}, \lambda^S_{k,l}, \lambda^{PF}_{k,l}, ..., \lambda^e_{k,l}\}, 0\} \tag{12}$$

where

$\lambda^V_{k,l}, \lambda^I_{k,l}, \lambda^S_{k,l}, \lambda^{PF}_{k,l}$ are voltage, current, thermal, and PF constraint-limited LC estimates,

the ellipsis indicates that more LC estimators may be included, and

$\lambda^e_{k,l}$ is a bound imposed by the time window structure.

1) Remarks:

- If $\lambda_{k,1} = \lambda^e_{k,1}$, then time window k contains no critical times, and sub-window $(k, 1)$ spans the full time window; otherwise the time window is sub-divided.
- A negative constraint-limited LC estimate indicates that $S(t_{k,l})$ is infeasible. This yields a sub-window duration $\lambda_{k,l} = 0$, and requires corrective control before further variation. This is further discussed in Section V.

D. Unscheduled Control Actions

When a critical time is identified, an unscheduled control action is attempted in order to maintain feasibility. The following features apply:

- Specific control algorithms can be chosen by the user.
- Control rules (e.g., dead bands or switching frequency limits) may be included. In this work, control rules are modeled as inequality constraints.
- Actuation triggers (e.g., voltage set points for capacitor banks) may be included in the constraint set to account for automatic control via local sensing.

After the injection space is updated to account for a control action, a new sub-window begins and the ITLC continues to traverse the injection forecast.

If a critical time is identified and no available control action yields a feasible result, then ITLC will continue to traverse the forecast and identify sustained constraint violations. An operator may use this information to determine whether a violation ride-through is viable, or if an emergency condition exists within the forecast.

The ITLC outputs include the realized control setting $u^*_{k,l}$ for each sub-window. Note that because $\lambda_{k,l}$ can be zero, an unscheduled control action may occur at the beginning of a time window in order to relieve an initial constraint violation.

E. ITLC Outputs

Implicit temporal load capability returns a sequence of sub-window durations Λ, and an associated sequence of realized control settings U^*. The solution provides a time-stamped forecast of critical times and control actions, which accounts for operator preferences defined in U.

Distribution system operators may use ITLC results to:

- time-localize events that are typically identified in terms of the injection space, such as reverse power flows, line congestion, voltage problems, or PF problems.
- identify periods of greater operating risk, when closer monitoring is necessary.
- identify periods of lower operating risk, when computational or manpower resources can be reallocated.
- evaluate the ability to ride-through temporary operating constraint violations.

Fig. 1 is provided to illustrate the basic concept of ITLC. The upper table shows the input sequences (8)-(10) and S_{init} is marked on the plot. The initial forecast is divided into $K = 5$ uniform time windows (separated by solid vertical lines). In time window k, $\hat{S}_k$ is equal to the slope of the uncontrollable injection forecast. u_k represents scheduled control settings for

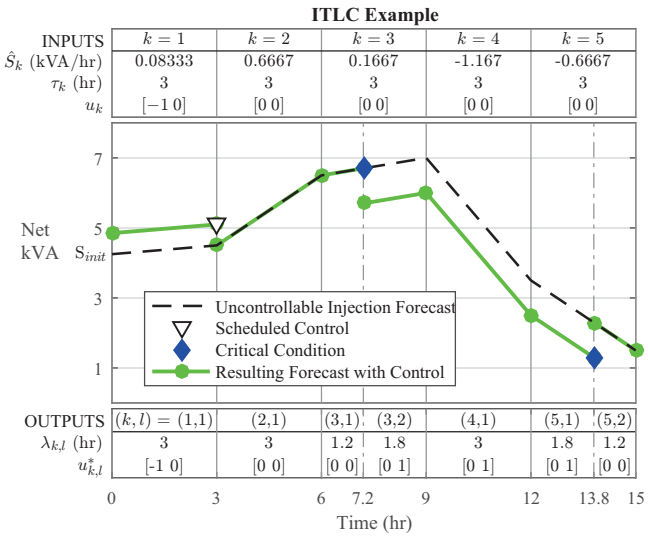

Fig. 1. Top: input sequences. Middle: initial and adjusted injection profiles. Bottom: output sequences. Please see Section III-E for a detailed description.

time window k; here, the two elements of the control vector are associated with a battery and a capacitor.

Scheduled control setting $u_1 = [-1\ 0]$ indicates that during time window 1, the battery is charging and the capacitor is not energized. At $t_2 = 3$, the battery is disconnected, and no more control actions are scheduled: $u_k = [0\ 0]$, $k = 2, ..., 5$.

In time window 3, a critical condition is mapped to critical time $t = 7.2$. Time window 3 is separated into sub-windows $(3, 1)$ and $(3, 2)$ (dash-dot line) and the capacitor is energized to prevent a violation. The revised control setting $u^*_{k,l} = [0\ 1]$ is realized in sub-windows $(3, 2)$, $(4, 1)$, and $(5, 1)$.

In time window 5, a critical time is identified at $t = 13.8$. This sub-divides time window 5; the capacitor is turned off to prevent a constraint violation: $u^*_{5,2} = [0\ 0]$. The output sequences are provided in the lower table within Fig. 1.

IV. Problem Formulation

Given S_{init} and sequences $\hat{S}$, T, and U, the goal of implicit temporal load capability is to find:

$$\Lambda = (\lambda_{1,1}, ..., \lambda_{1,L_1}, ..., \lambda_{K,1}, ..., \lambda_{K,L_K}) \quad (13)$$

$$U^* = (u^*_{1,1}, ..., u^*_{1,L_1}, ..., u^*_{K,1}, ..., u^*_{K,L_K}) \quad (14)$$

such that, $\forall \lambda_{k,l}, \forall u^*_{k,l}$,

$$F(x, \lambda, u^*) = 0 \quad (15)$$

$$G(x, \lambda, u^*) \leq 0 \quad (16)$$

where (15) represents the power flow equations, and (16) represent the electrical and operating constraints. The inequality constraints may include:

$$V_i^{\min} \leq |V_i| \leq V_i^{\max}, \qquad \forall i = 1, ..., n \quad (17)$$

$$|I_b| \leq I_b^{\max}, \qquad \forall b = 1, ..., n_b \quad (18)$$

$$|S_b| \leq S_b^{\max}, \qquad \forall b = 1, ..., n_b \quad (19)$$

$$\theta^L_{VI,i} \leq \theta_{VI,i} \leq \theta^U_{VI,i}, \qquad \forall i \in C_{PF} \quad (20)$$

$$H(x, \lambda, u^*) \leq 0 \quad (21)$$

where

V_i is the voltage at node i,
I_b is the current through branch b,
S_b is the complex power flow through branch b,
$\theta_{VI,i}$ is the PF angle of the injection at node i,
$V_i^{\min}$, $V_i^{\max}$, $I_b^{\max}$, $S_b^{\max}$ are voltage, current, and apparent power magnitude limits,
$\theta_{VI,i}^L$, $\theta_{VI,i}^U$ are lower and upper PF angle bounds [5],
n_b is the number of branches,
C_{PF} is the set of nodes with PF constraints,
$H(x, \lambda, u^*)$ represents the control constraints.

1) Remark on Λ: Recall that finding each element in (13) requires nonlinear optimization(s).

2) Remarks on Control Constraints: The control constraints represented by (21) vary depending on the number and type of controllable devices in the system. These may include:

- a system-wide dead band that blocks switching for a predefined duration, D, following any switching action.
- daily switching limits for individual devices.
- local actuation triggers for individual devices.

V. Solution Algorithm

This section presents the constraint-driven main ITLC solution algorithm and control sub-algorithms for handling scheduled and unscheduled control. The sub-algorithms use greedy heuristics to return feasible options when possible. Integrating optimal or robust control is of interest and is reserved for future work.

The algorithms presented here use the following rules:

- When a scheduled control action causes a constraint violation, it is simply rejected.
- When a constraint violation is encountered, an unscheduled control action is selected with a local optimization.
- When a constraint violation is unavoidable due to the dead band or a lack of switching options, a violation ride-through is attempted.

A. Main Algorithm: Implicit Temporal Load Capability

The main ITLC algorithm contains load capability, constraint checking steps, and calls to the control sub-algorithms as necessary. A dead band timer, $\bar{D} \in [0, D]$ (units of time), is used to enforce the dead band constraint. Additional remarks are presented below.

1) Remarks on the Main Algorithm:

- **Line 7:** if a control action is scheduled and $\bar{D} = 0$, Sub-Algorithm 1 is called to perform scheduled control actions. Details are provided in Section V-B.
- **Line 11:** distribution LC estimates are computed at the beginning of each sub-window. Each estimate requires iteratively solving (2)-(4) with respect to different operating constraints. LC estimation necessarily includes constraint checking steps, ensuring that constraint-limited sub-window durations are feasible.
- **Lines 12 - 13:** the minimum LC estimate is recorded in $\lambda_{k,l}^{\text{LC}}$, and the sub-window duration is saved in $\lambda_{k,l}$. These lines are equivalent to (12), with $\bar{D}$ replacing zero to enforce the dead band constraint.

Main Algorithm: Implicit Temporal Load Capability

Input: S_{init}, and sequences $\hat{S}_k$, T, U.
Output: Sequences Λ and U^*, subject to (15)-(16)

1: Initialize the dead band timer $\bar{D} = 0$
2: Initialize injections $S(t_{1,1}) = S_{\text{init}} + S_C(u_1)$ and state x (run multi-phase power flow)
Begin time window k
3: **for** $k = 1$ **to** K
4: Begin the first sub-window in this time window: $l = 1$
5: Initialize: $\lambda_{k,1} = 0$, $\lambda_{k,l}^e = \tau_k$
Check for a scheduled control action
6: **if** $k > 1$ **and** $u_k \neq u_{k-1}$ **and** $\bar{D} = 0$ **then**
7: Run **Sub-Alg. 1:** Scheduled Control: $u_{k,1}^* = \tilde{u}$
8: Update $S(t_{k,1}) = S(t_{k-1} + \tau_{k-1}) + S_{\text{C}}(u_{k,1}^*)$ and x
9: **end if**
Begin sub-window (k, l)
10: **while** $\lambda_{k,l} < \lambda_{k,l}^e$
11: Compute LC estimates $(\lambda_{k,l}^V, \lambda_{k,l}^I, \lambda_{k,l}^S, \lambda_{k,l}^{PF}, ...)$
12: $\lambda_{k,l}^{\text{LC}} = \min\{\lambda_{k,l}^V, \lambda_{k,l}^I, \lambda_{k,l}^S, \lambda_{k,l}^{PF}, ...\}$
13: $\lambda_{k,l} = \max\{\min\{\lambda_{k,l}^{\text{LC}}, \lambda_{k,l}^e\}, \bar{D}\}$
14: Update $S(t_{k,l} + \lambda_{k,l}) = S(t_{k,l}) + \lambda_{k,l}\hat{S}_k$ and x
15: **if** $\lambda_{k,l} = \lambda_{k,l}^e$ **then**
16: $\bar{D} = \max\{0, \bar{D} - \lambda_{k,l}\}$
17: **else if** $\lambda_{k,l} \geq \bar{D}$ **then**
18: $\bar{D} = 0$
19: Initialize: $l = l + 1$, $\lambda_{k,l} = 0$, $\lambda_{k,l}^e = t_{k+1} - t_{k,l}$
20: Run **Sub-Alg. 2:** Unscheduled Control: $u_{k,l}^* = \tilde{u}$
21: Update $S(t_{k,l}) = S(t_{k,l-1} + \lambda_{k,l-1}) + S_{\text{C}}(u_{k,l}^*)$ and x
22: **else**
23: $\bar{D} = \max\{0, \bar{D} - \lambda_{k,l}\}$
24: Initialize: $l = l + 1$, $\lambda_{k,l} = 0$, $\lambda_{k,l}^e = t_{k+1} - t_{k,l}$
25: **end if**
26: **end while**
27: **end for**

- **Line 20:** when a constraint violation is encountered before the end of a time window, Sub-Algorithm 2 is used to make unscheduled control actions. Details are provided in Section V-C.
- **Lines 2, 8, 14, and 21:** following any variation along $\hat{S}_k$ and/or discrete change(s) in the control setting, the net injection vector is updated. Then, the system state x is updated by solving the multi-phase power flow equations.

B. Sub-Algorithm 1: Scheduled Control

Sub-Algorithm 1 conducts switching scheduled in (10). Scheduled actions are ignored if: 1) $\bar{D} > 0$ at the switching time; 2) the scheduled action is infeasible; or 3) the scheduled action has no impact on the present control setting. In the first case, Sub-Algorithm 1 is unreachable. Other rules are possible; for example, a rule in [18] delays scheduled battery injections that introduce a low substation power factor (PF).

Sub-Algorithm 1: Scheduled Control

1: Choose $\tilde{u}$ to reflect the scheduled action.
2: **if** $\tilde{u}$ is feasible **and** requires switching **then**
3: Reset the dead band timer: $\bar{D} = D$.
4: **else**
5: Keep the previous control setting $\tilde{u} = u^*_{k-1,L^{k-1}}$.
6: **end if**

C. Sub-Algorithm 2: Unscheduled Control

Sub-Algorithm 2 switches power injection devices (e.g., battery or capacitor banks) to avoid violations (17)-(20). Users may choose a system-specific objective set (e.g., voltage spread reduction or loss minimization), search space, and/or solution technique. The simulations in Section VI use:

- constraint-dependent control objectives, which are listed in the steps of Sub-Algorithm 2.
- a local search space, which includes settings that are reachable with a single switching operation.
- a greedy selection strategy, which ranks local options based on the estimated ability to clear the violation, and selects the highest ranked option with a feasible result.

VI. Simulation Results

Simulations were performed using data from an actual multi-phase, radial power distribution circuit. The 2556-node test circuit contains 426 multi-phase loads and 6 three-phase, gang-operated capacitors. Fig. 2 provides a one-line diagram of the three-phase portion of the test circuit.

Nodal injection data was available at two loading levels: peak ($9191 + j3121$ kVA) and light ($3068 + j1018$ kVA). Under peak demand, the following constraints are violated:

- $|I_{3b}| \leq 462$ A.
- $\theta_{VI,1c} \leq 0.2003$ rad, corresponding to a power factor of no less than 0.98 lagging at the substation.

This PF constraint is also violated under light loading. AMI data and corresponding power flow solutions provide no information regarding the time or duration of these violations.

This section presents two simulations. First, a QSTS simulation is used to obtain steady-state operating conditions corresponding to a time series injection forecast. Then, an ITLC simulation is used to identify critical times and to select control actions to avoid constraint violations.

Sub-Algorithm 2: Unscheduled Control

1: **if** $\lambda^{\text{LC}}_{k,l} = \lambda^{V}_{k,l}$ **then**
2: Choose $\tilde{u} = \arg\min_u\{|V_i| - 0.5(V_i^{\max} + V_i^{\min})\}$, where i is the node with a voltage violation
3: **else if** $\lambda^{\text{LC}}_{k,l} = \lambda^{I}_{k,l}$ **then**
4: Choose $\tilde{u} = \arg\max_u\{I_b^{\max} - |I_b|\}$, where b is the branch with a current violation
5: **else if** $\lambda^{\text{LC}}_{k,l} = \lambda^{S}_{k,l}$ **then**
6: Choose $\tilde{u} = \arg\max_u\{S_b^{\max} - |S_b|\}$, where b is the branch with a thermal violation
7: **else if** $\lambda^{\text{LC}}_{k,l} = \lambda^{PF}_{k,l}$ **then**
8: Choose $\tilde{u} = \arg\min_u\{\theta_{VI,i} - 0.5(\theta^L_{VI,i} + \theta^U_{VI,i})\}$, where i is the node with a power factor violation
9: **end if**
10: **if** $\tilde{u}$ is feasible **then**
11: Reset the dead band timer: $\bar{D} = D$
12: **else**
13: Keep the previous control setting $\tilde{u} = u^*_{k,l-1}$ and attempt to ride-through the violation
14: **end if**

A. 24-Hour Quasi-Static Time Series Simulation

Typical demand curves were fitted around the known load levels to construct a 24-hour injection variation profile. Twenty-four hourly time windows were selected (i.e., $\tau_k = 1$ hr, $k = 1, ..., 24$), with light loading at $t_1 = 0$ hr (midnight) and peak loading at $t_{17} = 16$ hr (4 PM). QSTS results are obtained by computing the power flow solution at the beginning of each hourly time window and also at $t_{24}+\tau_{24}$ (at the end of the final time window).

Fig. 3 shows: 1) uncontrollable three-phase real and reactive demand forecasts at the substation; 2) the resulting time series of $|I_{3b}|$; and 3) the resulting time series of $\theta_{VI,1c}$. The QSTS simulation captured the following useful information that was unavailable from the AMI data sets alone:

- The substation, phase c PF violation is in place at every point in the time series but experiences relatively small variations throughout the day.
- The branch 3, phase b current rating is violated only under peak loading (4 PM). On the adjacent samples, (3 PM and 5 PM), $|I_{3b}|$ is within 3% of the 462 A rating.

QSTS simulation provides useful insights with hourly time

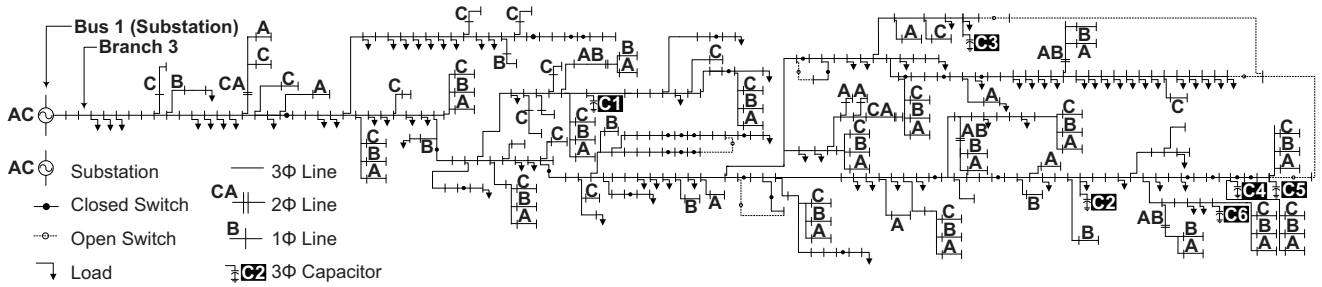

Fig. 2. One-line diagram of the three-phase portion of the 2556-node distribution test circuit. The circuit includes 426 multi-phase loads and 6 three-phase, gang-operated capacitors.

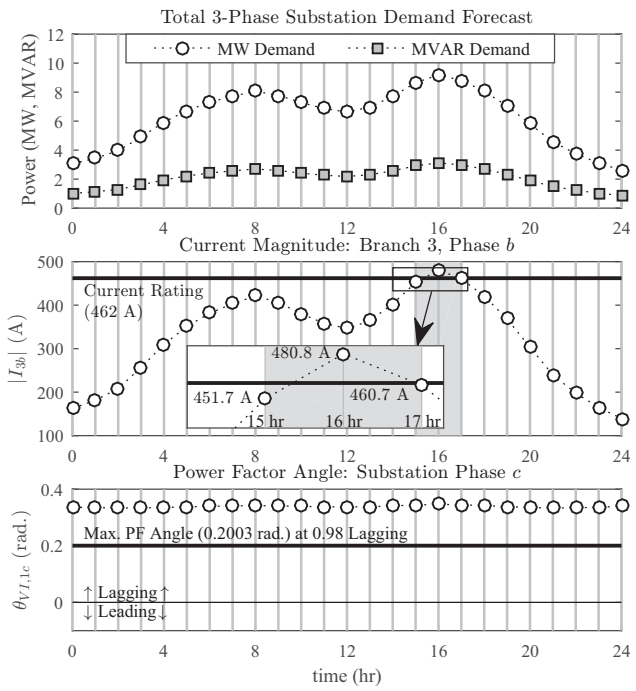

Fig. 3. Quasi-static time series simulation with assumed injection variation pattern. Top: uncontrollable real and reactive demand forecasts. Middle: current magnitude through branch 3, phase *b*. Bottom: power factor angle at the substation, phase *c*. Solid vertical lines denote time window divisions.

windows. Shorter windows could help better time-localize critical conditions, but at greatly increased computational cost.

Instead, ITLC uses the forecasted nodal injection characteristics to identify critical times in a significantly more computationally efficient manner. These critical times subdivide only those hourly time windows in which violations are anticipated, rather than requiring a large number of shorter time windows that span the entire forecast.

B. 24-Hour Implicit Temporal Load Capability Simulation

This simulation demonstrates critical time identification with ITLC. The six gang-operated capacitors are available. For illustrative purposes, it is assumed that the forecast is 100% accurate and that there is no delay between a critical time and the resulting operating time.

In each sub-window, distribution LC is computed with respect to all constraints (17)-(20). The following discussion focuses on constraints that were actually violated in the QSTS simulation: the current rating on branch 3, phase *b*, and the PF constraint at the substation, phase *c*.

The following subsections outline the steps of the solution algorithms with respect to the test circuit. Detailed descriptions are provided for the initialization steps and the first three time windows; an abbreviated description is presented for the remainder of the forecast.

1) ITLC Initialization: The initial injections S_{init} and the injection variation directions, $\{\hat{S}_k\}_{k=1}^{24}$, are real vectors of dimension 2556×1. Due to size constraints, their information is cumulatively represented in the total three-phase substation demand forecast in Fig. 3.

Each time window is one hour long: $\{\tau_k = 1\}_{k=1}^{24}$. Initially, all capacitors are off. Capacitor C2 (300 kVAR/phase) is scheduled to turn on at $t_{14} = 13$ hr (1 PM) and off at $t_{20} = 19$ hr (7 PM) in order to support the system under peak loading conditions. Therefore, initial control schedule U consists of:

$$u_k = \begin{cases} [0\,0\,0\,0\,0\,0], & k = 1, ..., 13, 20, ..., 24 \\ [0\,1\,0\,0\,0\,0], & k = 14, ..., 19 \end{cases}$$

where 0 (1) in the i^{th} position of the u_k vector indicates an off (on) status for gang-operated capacitor "C*i*" during time window k. See Fig. 2 for the capacitor locations.

A system-wide dead band parameter for capacitor switching $D = 0.1$ hr is included in the constraint set. The Main Algorithm begins by initializing the dead band timer at $\bar{D} = 0$ hr and initializing the nodal injection vector $S(t_{1,1} = 0)$ to reflect the net midnight injections.

2) Details for Time Window 1 *(Initial Violation):* Time window $k = 1$ is traversed in the first for-loop iteration. In sub-window $(1, 1)$, the estimators defined in [12], [14] yield load capability estimates of:

$$\lambda_{1,1}^{I} = 15.002 \quad \lambda_{1,1}^{PF} = -54.522$$

with units of hours, where the limiting current rating is on branch 3, phase *b* and the limiting PF constraint is on substation phase *c*. The maximum sub-window duration is:

$$\lambda_{1,1}^{e} = \tau_k = 1$$

The appropriate sub-window duration $\lambda_{1,1}$ is found with Main Algorithm lines 12-13:

$$\begin{array}{lll} \lambda_{1,1}^{\text{LC}} & = \min\{\lambda_{1,1}^{I}, \lambda_{1,1}^{PF}\} & = \lambda_{1,1}^{PF} \\ \lambda_{1,1} & = \max\{\min\{\lambda_{1,1}^{\text{LC}} \lambda_{1,1}^{e}\}, \bar{D}\} & = \bar{D} \quad = 0 \text{ hr} \end{array}$$

Since $\bar{D} \leq \lambda_{1,1} < \lambda_{1,1}^{e}$, a new sub-window $(1, 2)$ is initialized, and $\lambda_{1,2}^{e} = 1$. Sub-Algorithm 2 is entered, where the dead band timer is set to $\bar{D} = D = 0.1$. Sub-Algorithm 2 returns $u_{1,2}^{*} = [1\,0\,0\,0\,0\,0]$. This requires energizing capacitor C1, which nominally supplies 200 kVAR/phase.

$S(t_{1,2} = 0)$ is updated to reflect the corrective control and multi-phase power flow is run to update the state. Energizing C1 relieves the initial PF violation (see the "jump" in PF angle at $t = 0$ in Fig. 4).

The next while-loop iteration begins with the updated state and the initially feasible control setting $u_{1,2}^{*} = [1\,0\,0\,0\,0\,0]$. With these conditions, the load capability estimates are:

$$\lambda_{1,2}^{I} = 15.984 \quad \lambda_{1,2}^{PF} = 3.403$$

and the sub-window duration is computed as follows:

$$\begin{array}{lll} \lambda_{1,2}^{\text{LC}} & = \min\{\lambda_{1,2}^{I}, \lambda_{1,2}^{PF}\} & = \lambda_{1,2}^{PF} \\ \lambda_{1,2} & = \max\{\min\{\lambda_{1,2}^{\text{LC}} \lambda_{1,2}^{e}\}, \bar{D}\} & = \lambda_{1,2}^{e} \quad = 1 \text{ hr} \end{array}$$

$\lambda_{1,2} = \lambda_{1,2}^{e}$, and the dead band timer expires ($\bar{D} = 0$). Injections vary along $\hat{S}_1$ for a duration of $\lambda_{1,1} = 1$ hr:

$$S(1) = S(0) + 1 \times \hat{S}_1$$

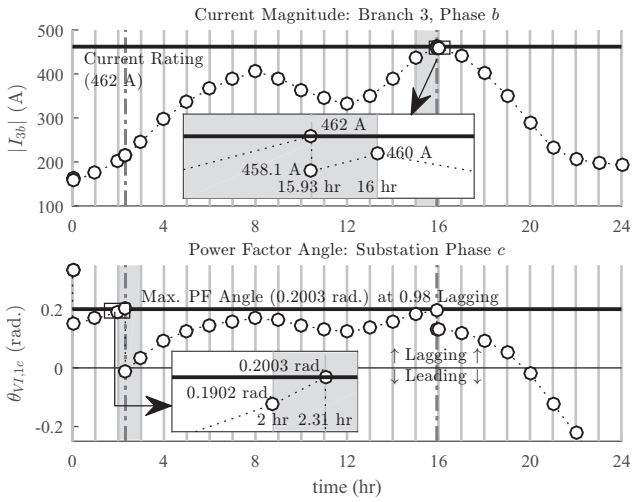

Fig. 4. Critical variable profiles with preventative control actions at the critical times. Top: current magnitude through branch 3, phase *b*. Bottom: power factor angle at the substation, phase *c*. Solid vertical lines denote time window divisions. Dashed vertical lines denote critical times/sub-window divisions.

and the updated state is computed using multi-phase power flow. Since $\lambda_{1,2} = \lambda^e_{1,2}$, the while-loop is exited, and the time window ends. Following the corrective control at $t = 0$, no more critical times were identified in $k = 1$.

3) Details for Time Window 2 *(No Violation):* No control actions are scheduled at t_2 (i.e., $u_2 = u_1$), and no critical times are identified within time window $k = 2$. Thus, no new sub-windows are created, no control actions take place, and $\bar{D}$ does not reset. At the end of $k = 2$, the following is recorded:

$$\begin{aligned} \lambda_{2,1} &= \lambda^e_{2,1} = 1 \\ u^*_{2,1} &= [1\,0\,0\,0\,0\,0] \\ S(2) &= S(1) + 1 \times \hat{S}_2 \end{aligned}$$

4) Details for Time Window 3 *(PF Violation):* No control actions are scheduled at t_3. $\lambda^e_{3,1} = \tau_3 = 1$ hr. The relevant load capability estimates are:

$$\lambda^I_{3,1} = 5.2615 \quad \lambda^{PF}_{3,1} = 0.3098$$

where the limiting current rating is on branch 3, phase *b* and the limiting PF constraint is on substation phase *c*. Therefore:

$$\begin{aligned} \lambda^{\text{LC}}_{3,1} &= \min\{\lambda^I_{3,1}, \lambda^{PF}_{3,1}\} & &= \lambda^{PF}_{3,1} \\ \lambda_{3,1} &= \max\{\min\{\lambda^{\text{LC}}_{3,1}\lambda^e_{3,1}\}, \bar{D}\} & &= \lambda^e_{3,1} \quad = 0.3098\,\text{hr} \end{aligned}$$

Since $\lambda_{3,1} < \lambda^e_{3,1}$, a critical time has been identified within $k = 3$. Injections vary along $\hat{S}_3$ until the critical time:

$$S(2.3098) = S(2) + 0.3098 \times \hat{S}_3$$

Following algorithm logic for $\bar{D} \leq \lambda_{3,1} < \lambda^e_{3,1}$, sub-window $(3, 2)$ is initialized and $\lambda^e_{3,2} = 0.6902$ hr. Sub-Algorithm 2 is entered, where the dead band timer is reset to $\bar{D} = D = 0.1$. The new control setting is $u^*_{3,2} = [1\,1\,0\,0\,0\,0]$, which requires energizing capacitor C2 (300 kVAR/phase). The injections $S(t_{3,2} = 2.3098)$ and the state are updated to reflect this change. This relieves the PF violation (see the "jump" in PF angle during $k = 3$ in Fig. 4). Here, ITLC has chosen to energize C2 earlier than originally scheduled.

The while-loop repeats, and no further critical times are encountered in $k = 3$. The dead band timer expires ($\bar{D} = 0$), and the following is recorded:

$$\begin{aligned} \lambda_{3,2} &= \lambda^e_{3,2} = 0.6902 \\ u^*_{3,2} &= [1\,1\,0\,0\,0\,0] \\ S(3) &= S(2.3098) + 0.6902 \times \hat{S}_3 \end{aligned}$$

5) Abbreviated Details for Time Windows 4 to 24 (Current Violation in $k = 16$*):* In $k = 4, ..., 13$, no control actions are scheduled and no critical times are identified. These windows proceed in a fashion similar to time window $k = 2$.

In $k = 14$, capacitor C2 is scheduled to turn on. Sub-Algorithm 1 is entered, but since C2 is already on, there is no change. Moving forward, no additional critical times occur until $k = 16$. The following outputs are recorded:

$$\left.\begin{aligned} \lambda_{k,1} = \lambda^e_{k,1} = 1 \\ u^*_{k,1} = [1\,1\,0\,0\,0\,0] \end{aligned}\right\}, k = 4, ..., 15$$

A critical time is identified within $k = 16$ at $t = 15.9322$ when the current rating at branch 3, phase *b* limits load capability. To prevent this violation, a new sub-window is initialized and Sub-Algorithm 2 returns a decision to energize capacitor C3 (200 kVAR/phase). The outputs for $k = 16$ are:

$$\begin{aligned} \lambda_{16,1} = \lambda^I_{16,1} = 0.9322 \qquad & u^*_{16,1} = [1\,1\,0\,0\,0\,0] \\ \lambda_{16,2} = \lambda^e_{16,2} = 0.0678 \qquad & u^*_{16,2} = [1\,1\,1\,0\,0\,0] \end{aligned}$$

When $k = 16$ ends, the unexpired dead band timer is at $\bar{D} = 0.0322$ hr. This carries over to $k = 17$, in which the dead band expires. No scheduled actions or critical times are encountered in time windows $k = 18, 19$.

At the beginning of $k = 20$, capacitor C2 is scheduled to turn off. Sub-Algorithm 1 is entered, where it is determined that this change is infeasible because it would reduce the PF at node $1c$ to below 0.98 lagging. This scheduled action is ignored. The control setting $u^*_{k,1} = [1\,1\,1\,0\,0\,0]$ is realized in sub-windows $(k, 1)$, $k = 17, ..., 24$.

6) ITLC Results Summary: Table I summarizes the ITLC simulation results. Sub-windows are listed along with their respective intervals, durations, limiting operating constraints (when applicable), and realized control settings.

ITLC identified critical times at 12:00 AM, 2:18 AM and 3:56 PM. Time windows $k = 1, 3, 16$ were each divided into two sub-windows, and Sub-Algorithm 2 made feasible capacitor switching decisions to avoid constraint violations.

ITLC also extracted temporal information that the QSTS simulation could not provide. For example, ITLC results show that the capacitor actuation at 2:18 AM frees enough branch capacity to delay the overcurrent violation until 3:56 PM, four minutes before forecasted peak demand. This type of information could allow operators to make more informed decisions when choosing whether to switch a capacitor or to ride-through the temporary overcurrent violation.

7) Remark on computation: : The analysis required 3,569 power flow solutions, less than 5% of the 86,400 solutions required for a day-ahead QSTS analysis with one second time steps (as in [4]). This computational demand level is comparable to QSTS with 24 second time steps.

TABLE I
ITLC SIMULATION RESULTS, ORGANIZED BY SUB-WINDOW.

Sub-window (k,l)	(1,1)†	(1,2)	(2,1)	(3,1)	(3,2)	(4,1)	...	(15,1)	(16,1)	(16,2)	(17,1)	...	(24,1)
$[t_{k,l}, t_{k,l}+\lambda_{k,l})$ (hr)	-	[0‡,1)	[1,2)	[2,2.3098)	[2.3098‡,3)	[3,4)	...	[14,15)	[15,15.9322)	[15.9322‡,16)	[16,17)	...	[23,24)
HH:MM AM/PM	-	[12‡, 1 A)	[1, 2 A)	[2, 2:18 A)	[2:18‡, 3 A)	[3, 4 A)	...	[2, 3 P)	[3, 3:56 P)	[3:56‡, 4 P)	[4, 5 P)	...	[11 P, 12 A)
Duration $\lambda_{k,l}$ (hr)	0	1	1	0.3098	0.6902	1	...	1	0.9322	0.0678	1	...	1
Limiting constraint	$\cos\theta_{VI,1c}$	-	-	$\cos\theta_{VI,1c}$	-	-	...	-	$\lvert I_{3b}\rvert$	-	-	...	-
Control setting $u^*_{k,l}$	$[0\,0\,0\,0\,0\,0]$	$[1\,0\,0\,0\,0\,0]$			$[1\,1\,0\,0\,0\,0]$					$[1\,1\,1\,0\,0\,0]$			

†PF violation caused by an initially infeasible control setting was corrected by energizing capacitor C1 before variation. This sub-window has a duration of 0 hr.
‡Critical time.

VII. CONCLUSION

This work presented the implicit temporal load capability (ITLC) model for critical time identification in smart distribution systems. ITLC identifies critical times using nodal injection characteristics and the sequential application of load capability. Critical times indicate when control is required to maintain feasibility with respect to static security constraints; this is a necessary step towards online optimal control.

A constraint-driven ITLC algorithm and control selection sub-algorithms were presented. Step-by-step simulation results illustrated how ITLC can be used to analytically refine an initial time window structure. In particular, it can be used to time-localize events that are typically identified in terms of the injection space such as reverse power flows, line congestion, or voltage problems.

Critical time identification can improve the benefit of time series studies relative to their computational cost. With traditional QSTS studies, time windows of arbitrarily short duration are required to pinpoint critical times. ITLC allows for the use of longer time windows and does not increase computational burden under low-risk conditions. In both ITLC and QSTS, one must assume that fast control dynamics have settled between steady state solutions [6]. A dead band (on the order of tens of seconds) can be used to ensure the validity of this assumption.

When considering practical obstacles (e.g., forecast error and control delays), analytically identified critical times can be used as a guide for targeting when to increase time-resolution within a time series analysis. For example, a simple rule may require an additional AMI data pull 10 minutes before an identified critical time. This would allow for additional processing and re-forecasting in order to refine the critical time and/or confirm the need for preventative control.

Several technical extensions are possible as well. For example, different control sub-algorithms may incorporate energy storage systems, demand response, and/or network reconfiguration. Various control cost functions or optimization horizons may also be explored. Since ITLC addresses time-varying injections, other time-varying parameters such as energy pricing and weather may be considered as well.

REFERENCES

[1] B. A. Mather, "Quasi-Static Time-Series Test Feeder for PV Integration Analysis on Distribution Systems," in *2012 IEEE Power and Energy Soc. General Meeting*, San Diego, CA, USA, 2012.

[2] D. Paradis, F. Katiraei and B. Mather, "Comparative analysis of time-series studies and transient simulations for impact assessment of PV integration on reduced IEEE 8500 node feeder," in *2013 IEEE Power and Energy Soc. General Meeting*, Vancouver, BC, Canada, 2013.

[3] M. R. Kleinberg, "Online Optimization of Capacitor Switching in Electric Power Distribution Systems," Ph.D. dissertation, Dept. of Elect. and Comput. Eng., Drexel Univ., Philadelphia, PA, USA, 2015.

[4] R. J. Broderick, J. E. Quiroz, M. J. Reno, A. Ellis, J. Smith, and R. Dugan. "Time Series Power Flow Analysis for Distribution Connected PV Generation," Sandia National Laboratories, Albequerque, NM, USA, Rep. SAND2013-0537, 2013.

[5] N. S. Coleman and K N. Miu, "A Study of Time Window Selection for Electric Power Distribution System Analysis," in *Proc. 2015 IEEE Int. Symp. Circuits and Systems*, Lisbon, Portugal, 2015, pp. 1891-1894.

[6] P. Rousseaux and T. Van Custem, "Quasi steady-state simulation diagnosis using Newton method with optimal multiplier," in *2006 IEEE Power and Energy Soc. General Meeting*, Montreal, QC, Canada, 2006.

[7] R. Perez-Guerrero, G. T. Heydt, N. J. Jack, B. K. Keel, and A. R. Castelhano Jr., "Optimal Restoration of Distribution Systems Using Dynamic Programming", *IEEE Trans. Power Syst.*, vol. 23, no. 3, Jul. 2008.

[8] M. Kleinberg, J. Harrison, and N. Mirhosseini, "Using Energy Storage to Mitigate PV Impacts on Distribution Feeders," in *2014 IEEE Power and Energy Soc. Innovative Smart Grid Technologies Conference*, Washington, DC, USA, 2014.

[9] Z. Hu, X. Wang, H. Chen, and G. A. Taylor, "Volt/VAr control in distribution systems using a time-interval based approach," *IEE Proc. on Generation, Transmission and Distribution*, vol. 150, no. 5, pp. 548-553, Sep. 2003.

[10] G. Ferrari-Trecati, E. Gallesty, P. Letizia, M. Spedicato, M .Morari, and M. Antoine, "Modeling and Control of Co-Generation Power Plants: A Hybrid System Approach," *IEEE Trans. Control Syst. Technol.*, vol. 12, no. 5, pp. 694-705, Sep. 2004.

[11] T. Geyer, M. Larssony, and M. Morari, "Hybrid emergency voltage control in power systems," in *Proc. of the European Control Conference*, Cambridge, UK, 2003, pp. 1893-1898.

[12] K. N. Miu and H.-D. Chiang, "Electric distribution system load capability: problem formulation, solution algorithm, and numerical results," *IEEE Trans. Power Del.*, vol. 15, no. 1, pp. 436-422, Jan. 2000.

[13] H. Liu, Z. Li, K. Yu and X. Chen, "The Real-Time Assessment of Electric Distribution Network Load Capability," in *2009 Asia-Pacific Power and Energy Eng. Conf.*, Wuhan, Mar. 2009.

[14] N. S. Coleman and K. N. Miu, "Distribution Load Capability with Nodal Power Factor Constraints," *IEEE Trans. Power Syst.*, to be published.

[15] P. W. Sauer, R. J. Evans Jr., and M. A. Pai, "Maximum unconstrained loadability of power systems," in *Proc. 1990 IEEE Int. Symp. Circuits Syst.*, New Orleans, LA, USA, 1990, pp. 1818-1821.

[16] G. D. Irisarri, X. Wang, J. Tong, and S. Mokhtari, "Maximum loadability of power systems using interior point nonlinear optimization method," *IEEE Trans. Power Syst.*, vol. 12, no. 1, pp. 162-172, Feb. 1997.

[17] C. D. Vournas, M. Karystianos, and N. G. Maratos, "Bifurcation points and loadability limits as solutions of constrained optimization problems," in *Proc. IEEE Power Eng. Soc. Summer Meeting*, Seattle, WA, USA, 2000, pp. 1883-1888.

[18] N. S. Coleman, "Load Capability for Smart Distribution Systems, M.S. thesis. Dept. of Electrical and Computer Engineering, Drexel Univ., Philadelphia, PA, 2013.

Nicholas S. Coleman (S' 2012) received B.S. and M.S. degrees in electrical engineering from Drexel University, Philadelphia, PA, USA. He is a Ph.D. candidate at the Drexel University Center for Electric Power Engineering.

Mr. Coleman was an inaugural recipient of the IEEE PES Scholar awards, and was selected as the John W. Estey Outstanding PES Scholar for IEEE Region II in 2012-13.

Karen N. Miu (M' 1998) received the B.S., M.S., and Ph.D.degrees in electrical engineering from Cornell University, Ithaca, NY, USA. She is currently a Professor in the Electrical and Computer Engineering Department, Drexel University, Philadelphia, PA, USA. Her research interests include distribution system analysis, distribution automation, and optimization techniques applied to power systems.