



Resilient Scheduling of Control Software Updates in Radial Power Distribution Systems

Kin Cheong Sou , Member, IEEE, and Henrik Sandberg , Fellow, IEEE

Abstract—In response to newly found security vulnerabilities, or as part of a moving target defense, a fast and safe control software update scheme for networked control systems is highly desirable. We here develop such a scheme for intelligent electronic devices (IEDs) in power distribution systems, which is a solution to the so-called software update rollout problem. This problem seeks to minimize the makespan of the software rollout, while guaranteeing safety in voltage and current at all buses and lines despite possible worst case update failure, where malfunctioning IEDs may inject harmful amounts of power into the system. Based on the nonlinear DistFlow equations, we derive linear relations relating software update decisions to the worst case voltages and currents, leading to a decision model both tractable and more accurate than previous models based on the popular linearized DistFlow equations. Under reasonable protection assumptions, the rollout problem can be formulated as a vector bin-packing problem, and instances can be built and solved using scalable computations. Using realistic benchmarks including one with 10,476 buses, we demonstrate that the proposed method can generate safe and effective rollout schedules in real time.

Index Terms—Cyber-physical systems, fault tolerant control, power system analysis computing, power system security, software algorithms.

I. INTRODUCTION

CRITICAL infrastructures, such as power and water distribution networks and cyber-physical systems (CPSs) in general, have become targets of cyberattacks in the past decade [1], [2]. Simultaneously, cybersecurity features in these systems are lacking. Control systems have been designed to meet safety requirements, often meaning a preference for low-complexity solutions that minimize time delay and prevent the

adoption of many standard computer security features. It has now become urgent to develop security solutions that take the special safety requirements of CPSs into account.

An essential difference between CPS and regular computer security is that “software patching and frequent updates are not well suited for control systems” [3]. Software updates are problematic in that they may require a complex reboot, which may come with operational costs and safety risks. It is not uncommon that the control software is updated infrequently, and systems may run for a long time even with known vulnerabilities. A practically relevant research problem is to develop control algorithms that can be safely updated in real time to patch newly discovered security weaknesses or as part of a moving target or software rejuvenation defense [4], [5], [6].

This article considers a control software update rollout problem for power distribution systems introduced in [7] and extends the results in [8]. The rollout problem seeks to arrange the software updates of intelligent electronic devices (IEDs) (e.g., smart inverters) into a minimum-time schedule, while guaranteeing power system operational safety despite worst case update failure. The problem is not solvable using traditional maintenance planning techniques [9] due to the intricate cyber-physical relationships involved. A central challenge here is an accurate yet tractable relation between the software update decisions and their worst case consequences to the system states (e.g., voltages and currents). We derive the desired relation using the *nonlinear* DistFlow equations [10], [11] as opposed to the popular linearized DistFlow equations adopted in [7]. The nonlinear DistFlow equations more accurately describe voltages and explicitly model line currents absent in the linearized equations. We numerically demonstrate the ramifications of the DistFlow models in maintaining safety standards. Compared with [8], this article presents major upgrades to the current and voltage safety limit constraints to reduce the conservatism of the rollout problem. Our numerical case studies indicate that the rollout schedules obtained in this article are effective over a wide variety of test cases. In addition, this article features streamlined computational procedures to set up and solve the rollout problem in real time. These include fixed-point iterations to compute voltage and current bounds (required to build a rollout problem instance) and a modified heuristic algorithm to solve the associated bin-packing problem. These innovations eliminate the need for mixed-integer (bi)linear programming in [8], enabling practical real-time large-scale rollout scheduling. For instance, our procedure can find a nontrivial schedule in a 10,476-bus

Manuscript received 14 July 2023; revised 19 July 2023 and 29 September 2023; accepted 17 October 2023. Date of publication 1 December 2023; date of current version 19 September 2024. The work of Kin Cheong Sou was supported in part by the National Science and Technology Council of Taiwan under Grant NSTC 112-2221-E-110-007. The work of Henrik Sandberg was supported in part by the Swedish Energy Agency and ERA-Net Smart Energy Systems under grant agreement No 883973 and in part by the Swedish Civil Contingencies Agency under project CERCES2. Recommended by Associate Editor Joshua A. Taylor. (Corresponding author: Kin Cheong Sou.)

Kin Cheong Sou is with the Department of Electrical Engineering, National Sun Yat-sen University, Kaohsiung 80424, Taiwan (e-mail: sou12@mail.nsysu.edu.tw).

Henrik Sandberg is with the Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden (e-mail: hsan@kth.se).

Digital Object Identifier 10.1109/TCNS.2023.3338254

case in less than 3 s. This was impossible in [8]. Our novel voltage and current bounds are potentially useful in robust and contingency-based ac optimal power flow problems, which have received significant interest in the past few years, see, e.g., [12], [13], [14], [15], [16], and [17].

Smart distribution grids play an essential role in society's transition into a net-zero energy system and thereby achieve highly set climate goals. The transition requires the integration of a wide variety of controllable energy devices, both for consumption and for supply. The interface between power supply (grid) and demand (consumer) is sometimes referred to as the *grid edge* [18]. The grid edge will turn distribution power grids, traditionally not very automated, into complex distributed control systems. Many local control loops in the grid edge simultaneously present an increased attack surface and will require resilient and systematic controller update/patching schemes to handle newly discovered vulnerabilities or to respond to contingencies [19], [20], [21], while accounting for possible adverse effects of failed control updates.

The rest of this article is organized as follows. Section II introduces notation, distribution system model, and its operational requirements. In Section III, the software update rollout problem is defined. The exact mathematical model is unfit for real-time applications. Thus, a tractable approximation is derived. Section IV summarizes the solution procedure for the rollout problem. Section V presents case studies to demonstrate that the proposed procedure is time efficient and is able to deliver effective and safe rollout schedules for large systems. Finally, Section VI concludes this article.

II. SYSTEM MODELING

A. Mathematical Notation

We define the following notation: the vector $\mathbf{1}$ (respectively, $\mathbf{0}$) is the all-one (respectively, all-zero) vector, and e_i for $i = 1, 2, \dots$ is the i th unit vector. By default, the symbol j means $\sqrt{-1}$. For a set of positive integers \mathcal{N} (details in Section II-B) and $\mathcal{I} \subseteq \mathcal{N}$, the symbol $\mathbf{1}_{\mathcal{I}} \in \{0, 1\}^{|\mathcal{N}|}$ is the 0–1 binary indicator vector with support \mathcal{I} . For any vector v , $\text{diag}(v)$ or D_v is the diagonal matrix with the diagonal entries defined by v . For any vector v and scalar a (positive or negative), the symbol v^a denotes the entrywise exponentiation (i.e., the i th entry of v^a is $(v_i)^a$). For any two vectors x and y of the same length, the symbol $x \odot y$ denotes the entrywise (Hadamard) product of x and y . For any two matrices A and B , the symbol $A \otimes B$ denotes the Kronecker product of A and B .

B. Power Distribution System Modeling

We consider single-phase radial power distribution systems. This can potentially be interpreted as the positive-sequence approximation of a three-phase system. We assume that the system operates in steady state and, hence, all the electrical quantities (e.g., current and voltage) can be represented using per unit phasors. We use the following notation to describe the system.

- 1) N denotes the number of nonreference buses (also number of lines).
- 2) $\mathcal{N} = \{1, \dots, N\}$ is the set of all nonreference buses. Bus 0 is the reference bus (also called the slack bus).
- 3) $\mathcal{L} = \{1, \dots, N\}$ is the set of all lines. Each line has a reference direction for line current and line power flow. Reference direction points away from bus 0. A line is labeled by the bus that it points to (e.g., line n points to bus n and no other line points to n due to radiality).
- 4) $\pi_n \in \mathcal{N} \cup \{0\}$ is the “parent” bus of $n \in \mathcal{N}$ (so that line n goes from π_n to n).
- 5) $d(n) \subseteq \mathcal{N}$: For any $n \in \mathcal{N}$, $d(n)$ denotes the set of descendants of n including n itself. That is, $m \in d(n)$ if and only if the (only) path from bus 0 to m traverses n .
- 6) \tilde{A} denotes the $N \times (N + 1)$ line–bus incidence matrix. For $(m, n) \in \mathcal{L} \times (\mathcal{N} \cup \{0\})$, $\tilde{A}_{mn} = 1$ if line m leaves bus n , $\tilde{A}_{mn} = -1$ if line m enters bus n , and $\tilde{A}_{mn} = 0$ otherwise.
- 7) A denotes the $N \times N$ submatrix of \tilde{A} with the first column a_0 removed (i.e., $\tilde{A} = [a_0 \ A]$).
- 8) \mathcal{D} denotes the $\{0, 1\}^{N \times N}$ descendant matrix so that $\mathcal{D}_{nm} = 1$ if and only if $m \in d(n)$. It can be verified that $\mathcal{D}^\top = -A^{-1}$ when row k of A corresponds to line k for all $k \in \mathcal{L}$.
- 9) All the shunt elements are ignored, and $z = r + jx$ is an N -vector of line series impedances; $r > \mathbf{0}$ is a vector of series resistances, and $x > \mathbf{0}$ is a vector of series reactances.
- 10) D_r and D_x denote $N \times N$ positive diagonal matrices with $D_r = \text{diag}(r)$ and $D_x = \text{diag}(x)$, respectively. Also, $D_z := D_r + jD_x$.
- 11) P , Q , and S are N -vectors of active, reactive, and apparent power flows for all the lines, respectively. P , Q , and S are defined at the sending ends of the lines (e.g., P_n is defined at bus π_n).
- 12) p , q , and s are N -vectors of net injections of active power, reactive power, and apparent power for nonreference buses, respectively.
- 13) $p = p^G - p^L$ and $q = q^G - q^L$, where p^G , q^G , p^L , and q^L are vectors of active power generation, reactive power generation, active power load, and reactive power load for nonreference buses, respectively.
- 14) ℓ denotes the N -vector of squared magnitude line currents.
- 15) v denotes the N -vector of magnitude voltage of nonreference buses.
- 16) ν denotes the N -vector of squared magnitude voltage (i.e., $\nu = v^2$). At bus 0, squared magnitude voltage ν_0 is constant.

For all the aforementioned vectors, subscript n means the n th entry of the vector. For instance, p_n^G is the active power generation injection to bus $n \in \mathcal{N}$, while ℓ_n is the squared current for line $n \in \mathcal{L}$. Fig. 1 illustrates a segment of the distribution system with the relevant electrical quantities.

For a radial distribution system with shunt elements ignored, the electrical quantities ν, ℓ, P, Q, p , and q are related by the

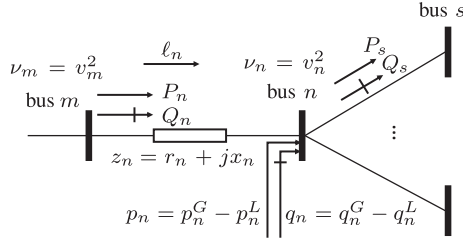


Fig. 1. Segment of distribution system with the relevant quantities.

(nonlinear) DistFlow equation [10], [11] as follows:

$$p = A^\top P + D_r \ell \quad (1a)$$

$$q = A^\top Q + D_x \ell \quad (1b)$$

$$A\nu + \nu_0 a_0 = 2D_r P + 2D_x Q - (D_r^2 + D_x^2)\ell \quad (1c)$$

$$\ell_n = \frac{P_n^2 + Q_n^2}{\nu_{\pi_n}} \quad \forall n = 1, 2, \dots, N. \quad (1d)$$

Note that $A^{-1} = -D^\top$ and $A^{-1}a_0 = -1$ (since $\tilde{A}\mathbf{1} = \mathbf{0}$). The variables P and Q can be eliminated from (1) to obtain

$$\nu = \nu_0 \mathbf{1} + 2R p + 2X q + M \ell \quad (2a)$$

$$\ell = (\nu)^{-1} \odot |D(p + jq) - (D - I)D_z \ell|^2 \quad (2b)$$

where

$$R = D^\top D_r D$$

$$X = D^\top D_x D$$

$$M = D^\top D_r (I - 2D) D_r + D^\top D_x (I - 2D) D_x. \quad (3)$$

Equation (2) implies that ν and ℓ are functions of net power injections (p, q) , motivating the shorthand $\nu = \nu(p, q)$ and $\ell = \ell(p, q)$. Also, since $D \in \{0, 1\}^{N \times N}$ with unit diagonal entries and r and x are positive, (3) implies that all the entries of R and X are nonnegative, while all the entries of M are nonpositive.

C. Distribution System Setting and Grid Code

We assume that every nonreference bus in the distribution system is equipped with a smart inverter to control and monitor its distributed generation (DG). The inverter-interfaced distributed generation is $p_n^G \geq 0$ for bus $n \in \mathcal{N}$. In addition, the inverter can provide reactive power support with reactive power injection q_n^G , which is not sign restricted. We assume that the rating of the inverter at bus n is C_n restricting its apparent power injection. If bus n has no generation, then $C_n = 0$. The vector of all the inverter ratings is denoted by $C \in \mathbb{R}^N$. Thus, the active and reactive power injections of the inverters satisfy

$$p_n^G \geq 0, \quad |p_n^G + jq_n^G| \leq C_n \quad \forall n \in \mathcal{N}. \quad (4)$$

It is assumed that the operator knows the inverter generation set points denoted by \hat{p}_n^G and \hat{q}_n^G and the load estimates denoted by \hat{p}_n^L and \hat{q}_n^L . In normal operating conditions, inverter set points \hat{p}_n^G and \hat{q}_n^G are chosen so that the grid code is satisfied. In this article, the grid code specifies that the steady-state ν and ℓ should

satisfy the following safety limits:

$$\underline{\nu} \leq \nu \leq \bar{\nu}, \quad \text{with given } \underline{\nu} \in \mathbb{R}^N \text{ and given } \bar{\nu} \in \mathbb{R}^N \quad (5a)$$

$$0 \leq \ell \leq \bar{\ell}, \quad \text{with given } \bar{\ell} \in \mathbb{R}^N. \quad (5b)$$

We note that $\ell \geq 0$ is not explicitly enforced as it is implied by (5a) and (1d). Typical values of the safety limits are: 0.9 p.u. for $(\underline{\nu})^{1/2}$, 1.1 p.u. for $(\bar{\nu})^{1/2}$, and a few hundred amperes (e.g., 600 A) for $(\bar{\ell})^{1/2}$.

III. PROBLEM STATEMENT AND FORMULATION

A. Rollout Problem Setting and Statement

As explained in the Section I, we envision a scenario where, owing to security or operational considerations, the operator needs to remotely update the software or firmware of all the inverters through some communication networks. To minimize possible disruption, the operator wishes to schedule the updates to finish as soon as possible (e.g., all at once if possible). However, some or all updates may fail. Software update failure for the inverter at a bus is modeled by the condition that, instead of following its power set point command, the inverter may inject an uncontrollable amount of power subjected to its physical limit in (4). We assume that each bus is equipped with a relay to promptly detect out-of-range voltage or current (usually within a few milliseconds). Then, the breaker or other protection will act to isolate the faulty inverter from the rest of the system, and the update for the inverter is rolled back. We assume that $\Delta\tau$ is the universal fault clearing time for the system. In other words, if an inverter experiences software update failure, it may inject uncontrolled amount of power for up to $\Delta\tau$ s, and then, its power output is back to normal.

It is not desirable to schedule too many updates at the same time, lest the updates fail simultaneously and inject a dangerous level of uncontrolled power causing power outage or even risk to human lives. Thus, the operator is faced with an update scheduling problem, where he/she seeks to minimize the makespan of the entire process while guaranteeing the safe operation of the system despite the worst case update failure due to the schedule. This problem is referred to as the *software update rollout problem* or the *rollout problem* for short.

Safe operation means that the steady-state squared voltage ν and steady-state squared line current ℓ satisfy the grid code constraints in (5). In this article, we do not consider transient safety. Power systems possess sufficient inertia so that transient voltage and current violations, though possibly larger than their steady-state values, dwindle rapidly. Inverters are designed with ride through capability so that relays will not trip even if violation is large for a short period of time.

Formally, the safety requirement of the rollout problem can be stated as follows: for any $\mathcal{I} \subseteq \mathcal{N}$, define $\mathcal{S}(\mathcal{I})$ by

$$\begin{aligned} \mathcal{S}(\mathcal{I}) := \{ & (p, q) \mid p = p^G - \hat{p}^L, \quad q = q^G - \hat{q}^L \\ & |p_n^G + jq_n^G| \leq C_n, \quad p_n^G \geq 0, \quad \text{if } n \in \mathcal{I} \\ & p_n^G = \hat{p}_n^G, \quad q_n^G = \hat{q}_n^G, \quad \text{if } n \in \mathcal{N} \setminus \mathcal{I} \} \end{aligned} \quad (6)$$

Algorithm 1: Fixed-Point Iterations to Bound ν^L and ℓ^U .

Input: Matrices \mathcal{D} and D_z in Section II-B, vectors \hat{p}^L, \hat{q}^L , and C in Section II-C, R, X , and M in (3), and tolerance $\epsilon > 0$

Output: $\hat{\nu}^L \leq \nu^L$ and $\hat{\ell}^U \geq \ell^U$

1: Denote $a := \mathcal{D}\hat{p}^L, b := \mathcal{D}\hat{q}^L, r := \mathcal{D}C, \theta \in \mathbb{R}^N$ s.t.

$$\theta_n := \begin{cases} \tan^{-1}(b_n/a_n), & \text{if } a_n \neq 0 \text{ or } b_n \neq 0 \\ 0, & \text{if } a_n = b_n = 0 \end{cases}$$

2: Define

$$\bar{S} := \max \{|a+j(b+r)|, |a+j(b-r)|, |a+jb-r \odot e^{j\theta}|\}$$

where “max” is interpreted rowwise

3: Initialize $v^0 \in \mathbb{R}^N$ and $i^0 \in \mathbb{R}^N$ such that $v^0 \leq (\nu^L)^{1/2}$ and $i^0 \geq (i^U)^{1/2}$ and $k \leftarrow 0$

4: **while** $\|v^k - v^{k-1}\| > \epsilon$ or $\|i^k - i^{k-1}\| > \epsilon$ **do**

5: $v^{k+1} = (\nu_0 \mathbf{1} - 2R\hat{p}^L - 2X(\hat{q}^L + C) + M(i^k)^2)^{1/2}$

6: $i^{k+1} = (v^k)^{-1} \odot (\bar{S} + |(\mathcal{D} - I)D_z(i^k)^2|)$

7: $k \leftarrow k + 1$

8: **end while**

9: Return $\hat{\nu}^L = (v^k)^2$ and $\hat{\ell}^U = (i^k)^2$

as the set of all possible (p, q) due to update failures at buses in $\mathcal{I} \subseteq \mathcal{N}$. Then, it is required that all the solutions $\nu(p, q)$ and $\ell(p, q)$ of (1) due to $(p, q) \in \mathcal{S}(\mathcal{I})$ satisfy the safety limits in (5). Subsequently, we derive linear relationships between $\mathbf{1}_{\mathcal{I}}$ (i.e., the decision indicator vector of the rollout problem) and the worst case $\nu(p, q)$ and $\ell(p, q)$ for $(p, q) \in \mathcal{S}(\mathcal{I})$ (used to describe the safety constraints in the rollout problem).

B. Universal Voltage Lower Bound and Current Upper Bound

To derive the safety constraints of the rollout problem, we need bounds of ν and ℓ as simple functions of the decision vector. Since $\ell \geq 0$ and M has nonpositive entries in (2), an immediate upper bound of ν arises when the term $M\ell$ in (2a) is ignored. This will be discussed in Section III-D. On the other hand, the lower bound of ν and the upper bound of ℓ are intertwined. To obtain the bounds, for any $n \in \mathcal{N}$, we denote the squared voltage lower bound ν_n^L and the squared current upper bound ℓ_n^U as

$$\nu_n^L := \min_{p, q, P, Q, \nu, \ell} \nu_n \quad \text{subject to (1) and (4)} \quad (7a)$$

$$\ell_n^U := \max_{p, q, P, Q, \nu, \ell} \ell_n \quad \text{subject to (1) and (4)}. \quad (7b)$$

Then, Sections III-C and III-E discuss how ν^L and ℓ^U can be used in (2) to obtain voltage and current bounds with respect to the decision vector. Problem (7) can be solved as a nonconvex bilinear program [due to (1d)]. However, this may not be suitable for real-time large-scale applications. Instead, we propose to quickly estimate lower bounds of ν^L and upper bounds of ℓ^U using the fixed-point iterations in Algorithm 1.

The following three questions pertain to Algorithm 1.

a) Does Algorithm 1 converge?

b) Is it true that $\hat{\nu}^L \approx \nu^L$ and $\hat{\ell}^U \approx \ell^U$?

c) Is it true that $\hat{\nu}^L \leq \nu^L$ and $\hat{\ell}^U \geq \ell^U$?

Question (a) regarding convergence can be verified online. Furthermore, our extensive simulations indicate quick convergence as long as the initial guess i^0 is not too large (so v^k stays real valued). As explained in Section V-B, question (b) regarding approximation quality can also be checked online. However, question (c) regarding bounding property cannot be verified numerically, and it must be proven analytically. Next, we provide an intuition to justify (b) and (c). Then, a statement is presented to prove (c).

Suppose that the while loop of lines 4–8 in Algorithm 1 converges with limits $\ell^* := (\lim_{k \rightarrow \infty} i^k)^2$ and $\nu^* := (\lim_{k \rightarrow \infty} v^k)^2$. Then, ℓ^* and ν^* are characterized by

$$\nu^* = \nu_0 \mathbf{1} - 2R\hat{p}^L - 2X(\hat{q}^L + C) + M\ell^* \quad (8a)$$

$$\ell^* = (\nu^*)^{-1} \odot (\bar{S} + |(\mathcal{D} - I)D_z\ell^*|)^2 \quad (8b)$$

which resembles (2), except that the right-hand sides (RHSs) of (8) are optimized with respect to (p, q) subject to (4). Indeed, owing to the fact that R and X in (3) have nonnegative entries and Lemma 1 in Appendix A, the following holds: for any $n \in \mathcal{N}$, entries ν_n^* in (8a) and ℓ_n^* in (8b) can be rewritten as

$$\nu_n^* = \min_{(p, q) \text{ in (4)}} e_n^\top (\nu_0 \mathbf{1} + 2Rp + 2Xq + M\ell^*)$$

$$\begin{aligned} \ell_n^* &= \max_{(p, q) \text{ in (4)}} e_n^\top \left((\nu^*)^{-1} \odot (|\mathcal{D}(p + jq)| + |(\mathcal{D} - I)D_z\ell^*|)^2 \right) \\ &\geq \max_{(p, q) \text{ in (4)}} e_n^\top \left((\nu^*)^{-1} \odot |\mathcal{D}(p + jq) - (\mathcal{D} - I)D_z\ell^*|^2 \right) \end{aligned} \quad (9)$$

where “ (p, q) in (4)” means (p, q) satisfying (4) in (9). The relations in (9) have two implications. First, the resemblance of (9) to (2) [equivalent to (1)] and the optimization suggest that $\nu^* \approx \nu^L$ and $\ell^* \approx \ell^U$ (hence, $\hat{\nu}^L \approx \nu^L$ and $\hat{\ell}^U \approx \ell^U$). Second, the optimization in (9) suggests the inequalities $\nu^* \leq \nu^L$ and $\ell^* \geq \ell^U$, and hence, $\hat{\nu}^L \leq \nu^L$ and $\hat{\ell}^U \geq \ell^U$. Indeed, this can be verified by the following statement, whose proof can be found in Appendix B.

Proposition 1: For Algorithm 1, it holds that $v^k \leq (\nu^L)^{1/2}$ and $i^k \geq (\ell^U)^{1/2}$ for all $k \geq 0$.

In particular, Proposition 1 specifies that $(\hat{\nu}^L, \hat{\ell}^U)$ as returned by Algorithm 1 satisfy the desired bounding inequalities. Proposition 1 also guarantees that, in case Algorithm 1 does not converge (though we never encounter this in our extensive simulations), any premature solution from Algorithm 1 still satisfies the bounding inequalities. In practice, Algorithm 1 converges quickly even if the initialization conditions $v^0 \leq (\nu^L)^{1/2}$ and $i^0 \geq (\ell^U)^{1/2}$ are not followed. As is standard in load flow analysis, we adopt a “flat start” of $v^0 = \mathbf{1}$ and $i^0 = \mathbf{0}$ in here. For convenience, the shorthand $\hat{\nu}^L := (\hat{\nu}^L)^{1/2}$ and $\hat{i}^U := (\hat{\ell}^U)^{1/2}$ will be used subsequently.

C. Expressions Related to Line Current Upper Limit

This section develops two separate results: 1) an upper bound of line current affinely dependent on $\mathbf{1}_{\mathcal{I}}$ and 2) given any line

current upper limit $\bar{\ell} \in \mathbb{R}^N$ a set of linear inequalities on $\mathbf{1}_{\mathcal{I}}$ to ensure that $\ell(p, q) \leq \bar{\ell}$ for all $(p, q) \in \mathcal{S}(\mathcal{I})$.

1) Affine Current Upper Bound: Let $\ell = \ell(p, q)$ from (2). Then, $\ell^{1/2}$ is the vector of magnitude line currents and, hence, by Kirchhoff's current law, we have

$$\sqrt{\ell_n} = \left| \sum_{m \in d(n)} i'_m \right| \leq \sum_{m \in d(n)} |i'_m| \leq \sum_{m \in d(n)} \frac{|p_m + jq_m|}{\hat{v}_m^L} \quad \forall n \in \mathcal{L} \quad (10)$$

where i'_m is the net current injection at bus m , $d(n)$ is the set of (n -inclusive) descendants of n defined in Section II-B, and \hat{v}^L is the voltage lower bound by Algorithm 1 [so that $\hat{v}^L \leq \nu^L$ in (7a)]. In (10), the last inequality holds because $|p_m + jq_m| = |v_m| |i'_m|$, and hence, $|p_m + jq_m| \geq \hat{v}_m^L |i'_m|$. Given power injection (p, q) , (10) provides an upper bound on $\ell(p, q)^{1/2}$. Let $\mathcal{I} \subseteq \mathcal{N}$ denote the set of buses with software update failure implying that $(p, q) \in \mathcal{S}(\mathcal{I})$ in (6). Then, the maximum line current $\sqrt{\ell_n}$ subject to \mathcal{I} is upper bounded by

$$\begin{aligned} & \max_{(p,q) \in \mathcal{S}(\mathcal{I})} \sum_{m \in d(n)} \frac{|p_m + jq_m|}{\hat{v}_m^L} \\ &= \sum_{m \in d(n) \setminus \mathcal{I}} \frac{|(\hat{p}_m^G - \hat{p}_m^L) + j(\hat{q}_m^G - \hat{q}_m^L)|}{\hat{v}_m^L} + \sum_{m \in d(n) \cap \mathcal{I}} \frac{|s^*|_m}{\hat{v}_m^L} \end{aligned} \quad (11)$$

where

$$\begin{aligned} |s^*|_m &= \max_{p_m^G, q_m^G} |(\hat{p}_m^L + j\hat{q}_m^L) - (p_m^G + jq_m^G)| \\ \text{s.t. } p_m^G &\geq 0, \quad |p_m^G + jq_m^G| \leq C_m. \end{aligned} \quad (12)$$

By Lemma 1 in Appendix A, $|s^*|_m$ in (12) is equal to

$$|s^*|_m = \max \{ |\hat{p}_m^L + j(\hat{q}_m^L + C_m)|, |\hat{p}_m^L + j(\hat{q}_m^L - C_m)|, |\hat{p}_m^L + j\hat{q}_m^L - C_m(\cos(\theta_{sm}) + j\sin(\theta_{sm}))| \} \quad (13)$$

where $\theta_{sm} = \tan^{-1}(\hat{q}_m^L/\hat{p}_m^L)$ if at least one of \hat{p}_m^L and \hat{q}_m^L is nonzero, and $\theta_{sm} = 0$ otherwise. Define

$$\begin{aligned} |s^*| &\in \mathbb{R}^N \text{ with entries specified in (13)} \\ |s^{\text{nom}}| &:= |(\hat{p}^G - \hat{p}^L) + j(\hat{q}^G - \hat{q}^L)|. \end{aligned} \quad (14)$$

Then, the current upper bound described by (11), (13), and (14) can be written compactly as

$$\begin{aligned} \ell^{1/2} &\leq \mathcal{D}((\hat{v}^L)^{-1} \odot |s^{\text{nom}}| + \text{diag}((\hat{v}^L)^{-1} \odot (|s^*| - |s^{\text{nom}}|)) \mathbf{1}_{\mathcal{I}}) \\ &:= I^{\text{iub1}} + W^{\text{iub1}} \mathbf{1}_{\mathcal{I}}. \end{aligned} \quad (15)$$

In summary, if $\mathcal{I} \subseteq \mathcal{N}$ denotes the set of bus(es) with software update(s), then despite the worst case update failure, the line current vector is upper bounded by the expression in (15).

2) Linear Inequalities Guaranteeing Current Upper Limit: For any net power injections (p, q) , (2b) specifies that

$$\begin{aligned} \sqrt{\ell_n} &= (\nu_n)^{-1/2} e_n^\top |\mathcal{D}(p + jq) - (\mathcal{D} - I)D_z \ell| \\ &\leq (\hat{v}_n^L)^{-1} e_n^\top \left(|\mathcal{D}(p + jq)| + |(\mathcal{D} - I)D_z(\hat{i}^U \odot \ell^{1/2})| \right) \end{aligned}$$

where \mathcal{D} and D_z are defined in Section II-B, and $\hat{v}^L = (\hat{\nu}^L)^{1/2}$ and $\hat{i}^U = (\hat{\ell}^U)^{1/2}$ from Algorithm 1. The aforementioned inequality is due to the following four facts: 1) $\hat{v}^L \leq (\nu(p, q))^{1/2}$; 2) triangular inequality; 3) $\mathbf{0} \leq \ell(p, q) = \ell^{1/2} \odot \ell^{1/2} \leq \hat{i}^U \odot \ell^{1/2}$; and 4) entries of $(\mathcal{D} - I)$, D_r , and D_x are nonnegative. Let $\mathcal{I} \subseteq \mathcal{N}$ denote the set of buses with software update failure implying that $(p, q) \in \mathcal{S}(\mathcal{I})$ in (6). Then, for any $n \in \mathcal{N}$

$$\begin{aligned} \max_{(p,q) \in \mathcal{S}(\mathcal{I})} \sqrt{\ell_n} &\leq (\hat{v}_n^L)^{-1} \max_{(p,q) \in \mathcal{S}(\mathcal{I})} e_n^\top |\mathcal{D}(p + jq)| \\ &\quad + (\hat{v}_n^L)^{-1} e_n^\top |(\mathcal{D} - I)D_z(\hat{i}^U \odot \ell^{1/2})|. \end{aligned} \quad (16)$$

The two terms on the RHS of (16) can be analyzed separately. By Lemma 2 in Appendix A, the first term is $(\hat{v}_n^L)^{-1} \max\{|a + j(b + r)|, |a + j(b - r)|, |a + jb - re^{j\theta}|\}$, with a , b , r , and θ defined in (49) in Lemma 2. Owing to θ , the expression is a nontrivial function of $\mathbf{1}_{\mathcal{I}}$, and hence, we consider its simplifying upper bound $(\hat{v}_n^L)^{-1}(|a + jb| + r)$, which is written as

$$\begin{aligned} & (\hat{v}_n^L)^{-1} \left| e_n^\top \mathcal{D}(\hat{p}^L - \hat{p}^G + j(\hat{q}^L - \hat{q}^G)) + e_n^\top \mathcal{D}D_{(\hat{p}^G + j\hat{q}^G)} \mathbf{1}_{\mathcal{I}} \right| \\ &+ (\hat{v}_n^L)^{-1} e_n^\top \mathcal{D}D_C \mathbf{1}_{\mathcal{I}}. \end{aligned} \quad (17)$$

Because of the absolute value, (17) is not affine with respect to $\mathbf{1}_{\mathcal{I}}$. Thus, it is further upper bounded by polygonal approximation. In particular, for any $x + jy \in \mathbb{C}$, we have

$$\begin{aligned} |x + jy| &\leq \frac{1}{\cos(\pi/N')} \\ &\quad \times \max_{1 \leq k \leq N'} \left\{ \cos\left(\frac{(2k-1)\pi}{N'}\right) x + \sin\left(\frac{(2k-1)\pi}{N'}\right) y \right\} \end{aligned} \quad (18)$$

for any integer $N' \geq 4$. For example, if $N' = 4$, then

$$|x + jy| \leq \max\{x + y, x - y, -x + y, -x - y\} = |x| + |y|.$$

Therefore

$$\begin{aligned} (17) &\leq (\hat{v}_n^L)^{-1} \frac{1}{\cos(\pi/N')} \\ &\quad \times \max_{1 \leq k \leq N'} \left\{ \cos\left(\frac{(2k-1)\pi}{N'}\right) (e_n^\top \mathcal{D}(\hat{p}^L - \hat{p}^G) \right. \\ &\quad \left. + e_n^\top \mathcal{D}D_{\hat{p}^G} \mathbf{1}_{\mathcal{I}}) + \sin\left(\frac{(2k-1)\pi}{N'}\right) (e_n^\top \mathcal{D}(\hat{q}^L - \hat{q}^G) \right. \\ &\quad \left. + e_n^\top \mathcal{D}D_{\hat{q}^G} \mathbf{1}_{\mathcal{I}}) \right\} + (\hat{v}_n^L)^{-1} e_n^\top \mathcal{D}D_C \mathbf{1}_{\mathcal{I}}. \end{aligned} \quad (19)$$

Now, we consider the second term on the RHS of (16). Using (15) and the fact that all the entries of $(\mathcal{D} - I)$, D_r , and D_x are nonnegative, the term can be upper bounded by

$$(\hat{v}_n^L)^{-1} \left| e_n^\top (\mathcal{D} - I) \text{diag}(z \odot \hat{i}^U) (I^{\text{iub1}} + W^{\text{iub1}} \mathbf{1}_{\mathcal{I}}) \right|. \quad (20)$$

Using polygonal approximation in (18) with integer $N'' \geq 4$, we obtain

$$\begin{aligned} (20) &\leq (\hat{v}_n^L)^{-1} \frac{1}{\cos(\pi/N'')} \\ &\quad \times \max_{1 \leq k \leq N''} \left\{ \cos\left(\frac{(2k-1)\pi}{N''}\right) e_n^\top (\mathcal{D} - I) \right\} \end{aligned}$$

$$\begin{aligned} & \times \text{diag}(r \odot \hat{i}^U)(I^{\text{iub1}} + W^{\text{iub1}} \mathbf{1}_{\mathcal{I}}) + \sin\left(\frac{(2k-1)\pi}{N''}\right) \\ & \times e_n^\top (\mathcal{D} - I) \text{diag}(x \odot \hat{i}^U)(I^{\text{iub1}} + W^{\text{iub1}} \mathbf{1}_{\mathcal{I}}) \}. \quad (21) \end{aligned}$$

Hence, the sum of the RHS of (19) and (21) is an upper bound of $\sqrt{\ell_n(p, q)}$ for all $(p, q) \in \mathcal{S}(\mathcal{I})$. Thus, $\forall (p, q) \in \mathcal{S}(\mathcal{I})$

$$\text{RHS of (19)} + \text{RHS of (21)} \leq (\bar{\ell}_n)^{1/2} \Rightarrow \sqrt{\ell_n} \leq (\bar{\ell}_n)^{1/2}. \quad (22)$$

Utilizing the fact that for any $x \in \mathbb{R}^{N'}$, $y \in \mathbb{R}^{N''}$, and $z \in \mathbb{R}$, we have

$$\max_i x_i + \max_j y_j \leq z \iff \mathbf{1}_{N''} \otimes x + y \otimes \mathbf{1}_{N'} \leq z \mathbf{1}_{N'N''}$$

with $\mathbf{1}_{N'}$, $\mathbf{1}_{N''}$, and $\mathbf{1}_{N'N''}$ denoting all-one vectors of appropriate dimensions, the condition in (22) can be “vectorized” as a set of linear constraints with respect to $\mathbf{1}_{\mathcal{I}}$ to enforce $\sqrt{\ell_n} \leq (\bar{\ell}_n)^{1/2}$ for a specific $n \in \mathcal{N}$. By “vectorizing” the conditions in (22) for all n , the desired linear constraints can be summarized as

$$I^{\text{iub2}} + W^{\text{iub2}} \mathbf{1}_{\mathcal{I}} \leq \mathbf{1}_{N''} \otimes (\mathbf{1}_{N'} \otimes (\bar{\ell})^{1/2}) \quad (23)$$

where

$$\begin{aligned} I^{\text{iub2}} &= \mathbf{1}_{N''} \otimes (c' \otimes ((\hat{v}^L)^{-1} \odot \mathcal{D}(\hat{p}^L - \hat{p}^G)) \\ &+ s' \otimes ((\hat{v}^L)^{-1} \odot \mathcal{D}(\hat{q}^L - \hat{q}^G))) \\ &+ c'' \otimes (\mathbf{1}_{N'} \otimes ((\hat{v}^L)^{-1} \odot (\mathcal{D} - I) \text{diag}(r \odot \hat{i}^U) I^{\text{iub1}})) \\ &+ s'' \otimes (\mathbf{1}_{N'} \otimes ((\hat{v}^L)^{-1} \odot (\mathcal{D} - I) \text{diag}(x \odot \hat{i}^U) I^{\text{iub1}})) \\ W^{\text{iub2}} &= \mathbf{1}_{N''} \otimes (c' \otimes D_{\hat{v}^L}^{-1} \mathcal{D} \mathcal{D}_{\hat{p}^G} + s' \otimes D_{\hat{v}^L}^{-1} \mathcal{D} \mathcal{D}_{\hat{q}^G} \\ &+ \mathbf{1}_{N'} \otimes D_{\hat{v}^L}^{-1} \mathcal{D} \mathcal{D}_C) \\ &+ c'' \otimes (\mathbf{1}_{N'} \otimes (D_{\hat{v}^L}^{-1} (\mathcal{D} - I) \text{diag}(r \odot \hat{i}^U) W^{\text{iub1}})) \\ &+ s'' \otimes (\mathbf{1}_{N'} \otimes (D_{\hat{v}^L}^{-1} (\mathcal{D} - I) \text{diag}(x \odot \hat{i}^U) W^{\text{iub1}})) \end{aligned} \quad (24)$$

and $c', s' \in \mathbb{R}^{N'}$, and $c'', s'' \in \mathbb{R}^{N''}$ with entries defined by

$$\begin{aligned} c'_k &= \frac{\cos((2k-1)\pi/N')}{\cos(\pi/N')} \quad \forall k \in \{1, \dots, N'\} \\ s'_k &= \frac{\sin((2k-1)\pi/N')}{\cos(\pi/N')} \quad \forall k \in \{1, \dots, N'\} \\ c''_k &= \frac{\cos((2k-1)\pi/N'')}{\cos(\pi/N'')} \quad \forall k \in \{1, \dots, N''\} \\ s''_k &= \frac{\sin((2k-1)\pi/N'')}{\cos(\pi/N'')} \quad \forall k \in \{1, \dots, N''\}. \end{aligned} \quad (25)$$

In summary, if $\mathcal{I} \subseteq \mathcal{N}$ denotes the set of bus(es) with software update(s), then despite all possible worst case update failure, if (23) holds, then the (squared) line currents are no greater than the upper limits specified by $\bar{\ell}$.

We note that (15) can also be used to construct line current upper limit constraints by imposing $I^{\text{iub1}} + W^{\text{iub1}} \mathbf{1}_{\mathcal{I}} \leq (\bar{\ell})^{1/2}$.

This is in fact adopted by our previous work in [8], requiring N instead of $NN'N''$ linear constraints in (23) used in this article. However, (15) is derived by replacing the current phasors with their magnitudes in (10). This tends to be conservative especially when there is significant reverse power flow in the system. Our numerical experiment (not shown in this article) indeed indicates that (23) is generally less conservative than (15). In addition, the total time to construct the constraints in (23) and to solve the corresponding rollout problem remains modest (see Section V for more detail). This motivates our proposed current upper limit constraints using (23). Nevertheless, (15) retains its distinct advantage that it is an upper bound of $(\ell)^{1/2}$ linearly dependent on $\mathbf{1}_{\mathcal{I}}$. Usually for $\mathcal{I} \subseteq \mathcal{N}$, $I^{\text{iub1}} + W^{\text{iub1}} \mathbf{1}_{\mathcal{I}} \leq \hat{i}^U$ because \hat{i}^U assumes update failure at all the buses in \mathcal{N} . This observation is utilized to obtain a less conservative second term in (16) [as in (20)].

D. Voltage Upper Limit Constraint

Note that $M\ell \leq 0$ in (2a) because all the entries of M in (3) are nonpositive. Therefore, for any (p, q) , $\nu(p, q) \leq \nu_0 \mathbf{1} + 2Rp + 2Xq$. Let $\mathcal{I} \subseteq \mathcal{N}$ denote the set of buses with software updates implying that $(p, q) \in \mathcal{S}(\mathcal{I})$ in (6). Then, for any $n \in \mathcal{N}$

$$\begin{aligned} \max_{(p,q) \in \mathcal{S}(\mathcal{I})} \nu_n &\leq \max_{(p,q) \in \mathcal{S}(\mathcal{I})} \{ \nu_0 + 2e_n^\top R p + 2e_n^\top X q \} \\ &= e_n^\top (\nu_0 \mathbf{1} + 2R(\hat{p}^G - \hat{p}^L) + 2X(\hat{q}^G - \hat{q}^L)) \\ &+ \sum_{i \in \mathcal{I}} \max_{p_i^G \geq 0, q_i^G} \{ 2R_{ni} p_i^G + 2X_{ni} q_i^G \mid |p_i^G + j q_i^G| \leq C_i \} \\ &- \sum_{i \in \mathcal{I}} (2R_{ni} \hat{p}_i^G + 2X_{ni} \hat{q}_i^G) \end{aligned} \quad (26)$$

where R_{ni} and X_{ni} denote the (n, i) entry of R and X , respectively. Since $R_{ni} \geq 0$ and $X_{ni} \geq 0$, in (26), we have

$$\begin{aligned} \max_{p_i^G \geq 0, q_i^G} \{ 2R_{ni} p_i^G + 2X_{ni} q_i^G \mid |p_i^G + j q_i^G| \leq C_i \} \\ = 2C_i \sqrt{R_{ni}^2 + X_{ni}^2} \\ := 2C_i Z_{ni} \end{aligned} \quad (27)$$

with

$$Z \in \mathbb{R}^{N \times N}, \quad Z_{ni} := \sqrt{R_{ni}^2 + X_{ni}^2} \quad \forall (n, i). \quad (28)$$

Note that in (26) (for the last term), we have

$$\sum_{i \in \mathcal{I}} (2R_{ni} \hat{p}_i^G + 2X_{ni} \hat{q}_i^G) = e_n^\top (2RD_{\hat{p}^G} + 2XD_{\hat{q}^G}) \mathbf{1}_{\mathcal{I}}. \quad (29)$$

Thus, (27) and (29) imply that the inequalities in (26) can be assembled for all n as

$$\nu(p, q) \leq \hat{\nu}^{\text{nom}} + W^{\text{vub}} \mathbf{1}_{\mathcal{I}} \quad \forall (p, q) \in \mathcal{S}(\mathcal{I}) \quad (30)$$

where

$$\begin{aligned} \hat{\nu}^{\text{nom}} &:= \nu_0 \mathbf{1} + 2R(\hat{p}^G - \hat{p}^L) + 2X(\hat{q}^G - \hat{q}^L) \\ W^{\text{vub}} &:= 2ZD_C - (2RD_{\hat{p}^G} + 2XD_{\hat{q}^G}). \end{aligned} \quad (31)$$

In summary, if \mathcal{I} denotes the set of buses with software updates, the maximum squared voltages are upper bounded by $\hat{\nu}^{\text{nom}} + W^{\text{vub}} \mathbf{1}_{\mathcal{I}}$ as specified in (30). Therefore, if the constraints

$$\hat{\nu}^{\text{nom}} + W^{\text{vub}} \mathbf{1}_{\mathcal{I}} \leq \bar{\nu} \quad (32)$$

are imposed on $\mathbf{1}_{\mathcal{I}}$, then the worst case possible squared voltages due to software update failure associated with inverters at buses in \mathcal{I} cannot exceed $\bar{\nu}$.

We remark that $\hat{\nu}^{\text{nom}}$ in (31) is the vector of nominal squared voltage (i.e., no software update failure) of the *linearized* Dist-Flow equations (i.e., (2a) with the loss term $M\ell$ ignored).

E. Voltage Lower Limit Constraint

Let $\mathcal{I} \subseteq \mathcal{N}$ denote the set of bus(es) with software update(s) implying that $(p, q) \in \mathcal{S}(\mathcal{I})$. Then, for any $n \in \mathcal{N}$, (2a) specifies that

$$\begin{aligned} \nu_n &= \nu_0 + 2e_n^\top R p + 2e_n^\top X q + e_n^\top M \ell \\ &\geq \nu_0 + e_n^\top \left(2R p + 2X q + M \left(\hat{i}^U \odot (I^{\text{iub1}} + W^{\text{iub1}} \mathbf{1}_{\mathcal{I}}) \right) \right) \end{aligned}$$

where the inequality holds because of the following four facts: 1) all the entries of M are nonpositive; 2) $\ell = \ell^{1/2} \odot \ell^{1/2}$; 3) for all (p, q) , it holds that $\ell^{1/2} \leq \hat{i}^U$ by (7b); and 4) for all $(p, q) \in \mathcal{S}(\mathcal{I})$, it holds that $\ell^{1/2} \leq I^{\text{iub1}} + W^{\text{iub1}} \mathbf{1}_{\mathcal{I}}$. Therefore, the minimum ν_n due to $(p, q) \in \mathcal{S}(\mathcal{I})$ satisfies

$$\begin{aligned} \min_{(p,q) \in \mathcal{S}(\mathcal{I})} \nu_n &\geq \nu_0 + \min_{(p,q) \in \mathcal{S}(\mathcal{I})} \{ 2e_n^\top R p + 2e_n^\top X q \} \\ &+ e_n^\top M \left(\hat{i}^U \odot (I^{\text{iub1}} + W^{\text{iub1}} \mathbf{1}_{\mathcal{I}}) \right) \\ &= e_n^\top (\nu_0 \mathbf{1}_{\mathcal{I}} + 2R(\hat{p}^G - \hat{p}^L) + 2X(\hat{q}^G - \hat{q}^L)) \\ &+ \sum_{i \in \mathcal{I}} \min_{p_i^G \geq 0, q_i^G} \{ 2R_{ni} p_i^G + 2X_{ni} q_i^G \mid |p_i^G + j q_i^G| \leq C_i \} \\ &- \sum_{i \in \mathcal{I}} (2R_{ni} \hat{p}_i^G + 2X_{ni} \hat{q}_i^G) + e_n^\top M \left(\hat{i}^U \odot (I^{\text{iub1}} + W^{\text{iub1}} \mathbf{1}_{\mathcal{I}}) \right). \end{aligned} \quad (33)$$

Since $R_{ni} \geq 0$ and $X_{ni} \geq 0$, in (33), we have

$$\begin{aligned} \min_{p_i^G \geq 0, q_i^G} \{ 2R_{ni} p_i^G + 2X_{ni} q_i^G \mid |p_i^G + j q_i^G| \leq C_i \} \\ &= - \sum_{i \in \mathcal{I}} 2X_{ni} C_i \\ &= -2e_n^\top X D_C \mathbf{1}_{\mathcal{I}}. \end{aligned} \quad (34)$$

Thus, (29), (31), and (34) imply that the inequalities in (33) can be assembled for all n as

$$\nu(p, q) \geq \hat{\nu}^{\text{nom}} + M(\hat{i}^U \odot I^{\text{iub1}}) - W^{\text{vlb}} \mathbf{1}_{\mathcal{I}} \quad \forall (p, q) \in \mathcal{S}(\mathcal{I}) \quad (35)$$

where

$$W^{\text{vlb}} := 2X D_C + 2R D_{\hat{p}^G} + 2X D_{\hat{q}^G} - M D_{\hat{i}^U} W^{\text{iub1}}. \quad (36)$$

In summary, if \mathcal{I} denotes the set of buses with software updates, then the minimum squared voltages are lower bounded by $\hat{\nu}^{\text{nom}} + M(\hat{i}^U \odot I^{\text{iub1}}) - W^{\text{vlb}} \mathbf{1}_{\mathcal{I}}$ as in (35). Therefore, if the

constraints

$$\hat{\nu}^{\text{nom}} + M(\hat{i}^U \odot I^{\text{iub1}}) - W^{\text{vlb}} \mathbf{1}_{\mathcal{I}} \geq \underline{\nu} \quad (37)$$

are imposed on $\mathbf{1}_{\mathcal{I}}$, then the worst case squared voltages due to software update failure associated with inverters at buses in \mathcal{I} cannot be lower than $\underline{\nu}$.

F. Software Update Rollout Problem Formulation

The decision variables of the rollout problem are the software update instants $t_i \geq 0$ for bus $i \in \mathcal{N}$. Since the updates should be scheduled to cause the least interruption to normal operation, the makespan (i.e., the time when the first update starts to the time when the last update finishes) should be minimized. Furthermore, the safety constraints should be satisfied. For given t_1, t_2, \dots, t_N and t , we define

$$\mathcal{I}(t) := \{i \in \mathcal{N} \mid t \in [t_i, t_i + \Delta\tau]\} \quad (38)$$

as the set of buses such that time interval $[t_i, t_i + \Delta\tau]$ contains t . Recall that $\Delta\tau$ is the fault clearing time, so $\mathcal{I}(t)$ is the set of buses, whose adverse effect of software update failure can be seen at time t (for given t_1, t_2, \dots). Then, to maintain the grid code in (5) in the worst case scenario (all the scheduled updates fail, and the corresponding nodes suffer the worst possible power injections), constraints (23), (32), and (37) should hold for $\mathcal{I}(t)$ at all t . Hence, the rollout problem is summarized as

$$\begin{aligned} &\underset{t_i}{\text{minimize}} \quad \max_{i \in \mathcal{N}} t_i \\ &\text{s.t.} \quad \mathbf{H} \mathbf{1}_{\mathcal{I}(t)} \leq \mathbf{b} \quad \forall t \\ &\quad \mathcal{I}(t) \text{ satisfies (38)} \end{aligned} \quad (39)$$

where

$$\mathbf{H} := \begin{bmatrix} W^{\text{iub2}} \\ W^{\text{vub}} \\ W^{\text{vlb}} \end{bmatrix}, \quad \mathbf{b} := \begin{bmatrix} \mathbf{1}_{N''} \otimes (\mathbf{1}_{N'} \otimes (\bar{\ell})^{1/2}) - I^{\text{iub2}} \\ \bar{\nu} - \hat{\nu}^{\text{nom}} \\ \hat{\nu}^{\text{nom}} + M(\hat{i}^U \odot I^{\text{iub1}}) - \underline{\nu} \end{bmatrix}. \quad (40)$$

Owing to the indicator function $\mathcal{I}(t)$, the problem (39) is non-linear and nonconvex. However, as shown in [7, Proposition 1], t_i in (39) can in fact be restricted to nonnegative integer multiples of $\Delta\tau$. Then, (39) can be reinterpreted as follows. The time axis can be divided into intervals called time slots of the form $[k\Delta\tau, (k+1)\Delta\tau]$ with $0 \leq k \leq N-1$. The decision to make is the assignment of updates to the time slots, so that each update is assigned to exactly one time slot. For the shortest update schedule, the number of time slots used is minimized. Furthermore, instead of imposing the constraint in (39) for all t , it suffices to consider only integer multiples of $\Delta\tau$. By defining the “discrete-time” version of \mathcal{I} in (38) with time slot index starting from $j = 1$

$$\mathcal{I}_j := \mathcal{I}((j-1)\Delta\tau), \quad j = 1, 2, \dots, N$$

problem (39) can be reformulated as follows:

$$\begin{aligned} &\underset{T}{\text{minimize}} \quad T \\ &\text{s.t.} \quad \mathcal{I}_1, \mathcal{I}_2, \dots, \mathcal{I}_T \text{ partition } \mathcal{N} \end{aligned}$$

$$\mathbf{H}\mathbf{1}_{\mathcal{I}_j} \leq \mathbf{b}, \quad j \in \{1, \dots, T\} \quad (41)$$

with \mathbf{H} and \mathbf{b} defined in (40). Problem (41) is known as the *vector bin-packing problem*. It can be modeled as an integer program (see, e.g., [7]) or approximately solved using greedy algorithms to be detailed in the next section.

IV. SOLVING SOFTWARE UPDATE ROLLOUT PROBLEM

To quickly solve large instances of (41), we adopt the best fit decreasing greedy algorithm. First, we check if it is possible to let each time slot hold exactly one update, since the instance is feasible if and only if this is possible. If feasible, the updates are sorted by their “sizes” specified by the row vector $\mathbf{1}^\top \mathbf{H}$ (large to small). Then, the sorted updates are considered one by one. For each update, we first try to assign it to the time slot with the minimum “sum-up residual capacity (to be defined in Algorithm 2),” among the used slots with some update already assigned. Only if no used slot can hold the update, will a new slot be appended to the schedule.

Algorithm 2 summarizes our customized best fit decreasing algorithm.

Assume that the instance is feasible. The assignment $\mathbf{C}(:, N_y) \leftarrow \mathbf{b}$ in line 16 specifies that whenever a new time slot is used, its capacity vector is initialized to be \mathbf{b} . Whenever an update is assigned to the time slot, the modification $\mathbf{C}(:, s^*) \leftarrow \mathbf{H}(:, \mathbf{u}_i)$ in line 21 ensures that the residual capacity vector of the slot is properly deducted. Then, the condition $\mathbf{C}(:, s_j) \geq \mathbf{H}(:, \mathbf{u}_i)$ in line 12 specifies that the time slot can accept an update only if its residual capacity vector remains nonnegative after the assignment. Together, the assignment returned by the algorithm satisfies the second constraints in (41). In addition, the for loop over i implies that each update must be assigned to exactly one time slot (a new slot if necessary). Hence, the first constraint of (41) is satisfied by the algorithm output. Therefore, the algorithm returns a feasible rollout, whenever the instance is feasible. Though the returned rollout is not necessarily optimal, it typically requires very few time slots (see Table II in Section V for detail).

The main loop over i is executed N times. Each iteration of the inner loop over j requires $O(K)$ operations. The remaining steps in an iteration of the main loop require $O(N + K)$ operations. Hence, the time complexity of Algorithm 2 is $O(N^2 K)$. In Section V, we demonstrate that Algorithm 2 as stated can be executed very efficiently even in MATLAB. For example, running Algorithm 2 for the 10 476-bus instance from [22] requires less than 2 s. The overall software rollout scheduling procedure is summarized in Algorithm 3.

V. NUMERICAL DEMONSTRATION

We demonstrate the practical performance of the proposed software update rollout scheduling procedure with distribution system benchmarks. All computation is performed on a Mac Studio (2022) with M1 Ultra CPU and 128 GB of RAM, running MATLAB through the Rosetta 2 translation environment.

Algorithm 2: Best Fit Decreasing Algorithm to Solve (41).

Input: $\mathbf{H} \in \mathbb{R}^{K \times N}$, $\mathbf{b} \in \mathbb{R}^K$
Output: $\mathbf{x} \in \{0, 1\}^{N \times N}$, $\mathbf{y} \in \{0, 1\}^N$ if instance is feasible; $\mathbf{x}_{ij} = 1$ iff update i is assigned to time slot j , and $\mathbf{y}_j = 1$ iff time slot j is needed in the rollout schedule

- 1: **if** $\mathbf{b} \not\geq \mathbf{H}(:, i)$ for any $i \in \mathcal{N}$ **then** \triangleright feasibility check
- 2: report infeasibility, QUIT
- 3: **end if**
- 4: Define $\mathbf{s} = \mathbf{1}^\top \mathbf{H}$ \triangleright “size” of each update
- 5: Define update indices $\mathbf{u}_1, \dots, \mathbf{u}_N: \mathbf{s}(\mathbf{u}_1) \geq \mathbf{s}(\mathbf{u}_2) \geq \dots$
- 6: Initialize $\mathbf{x} \leftarrow \mathbf{0}_{N \times N}$, $\mathbf{y} \leftarrow \mathbf{0}_N$
- 7: Initialize $N_y \leftarrow 0$ \triangleright number of time slots used
- 8: Initialize $\mathbf{C} \leftarrow \mathbf{0}_{K \times N}$ (residual capacity vectors for all time slots), $\mathbf{c} \leftarrow (\mathbf{b}^\top \mathbf{1}) \mathbf{1}^\top$ (sum-up residual capacity)
- 9: Initialize time slot indices $s_1 \leftarrow 1, \dots, s_N \leftarrow N$
- 10: **for** $i = 1, \dots, N$ **do** \triangleright loop over sorted updates
- 11: **for** $j = 1, \dots, N$ **do** \triangleright loop over sorted time slots
- 12: **if** $\mathbf{C}(:, s_j) \geq \mathbf{H}(:, \mathbf{u}_i)$ **then** \triangleright check used slots
- 13: $s^* \leftarrow s_j$, quit for loop over j
- 14: **end if**
- 15: **if** $j > N_y$ **then** \triangleright open a new slot
- 16: $s^* \leftarrow j$, N_y++ , $\mathbf{C}(:, N_y) \leftarrow \mathbf{b}$, $\mathbf{y}(N_y) \leftarrow 1$
- 17: quit for loop over j
- 18: **end if**
- 19: **end for**
- 20: Update $\mathbf{x}(\mathbf{u}_i, s^*) \leftarrow 1$
- 21: Update $\mathbf{C}(:, s^*) \leftarrow \mathbf{H}(:, \mathbf{u}_i)$, $\mathbf{c}(s^*) \leftarrow \mathbf{s}(\mathbf{u}_i)$
- 22: **while** $j \geq 2$ **do** \triangleright update used time slots ordering
- 23: **if** $\mathbf{c}(s_j) < \mathbf{c}(s_{j-1})$ **then**
- 24: Swap s_j with s_{j-1}
- 25: **end if**
- 26: $j--$
- 27: **end while**
- 28: **end for**

A. Distribution System Benchmarks

The benchmarks considered in this article include: CIGRE 44-bus low-voltage (LV) radial network from [23], radial networks from Matpower 7.1 [24], and the REDS repository [22]. The CIGRE network contains one substation busbar, three feeder heads, and 40 nonreference buses. In our study, the substation and the feeder heads are merged into one slack bus with squared voltage fixed at $v_0 = 1$ p.u. The descriptions (topology, line parameters, load, etc.) of all other benchmarks can be found in [22] and [24]. We assume that for each benchmark, all the nonreference buses are equipped with smart inverters with the same (benchmark dependent) power rating.

B. Quality of Universal Voltage and Current Bounds

We numerically demonstrate the quality of the voltage lower bound \hat{v}^L and current upper bound \hat{i}^U obtained by Algorithm 1. Owing to (7) and Proposition 1, for any (p, q) , it holds that

Algorithm 3: Software Update Rollout Scheduling Procedure.

- Input:** (a) DistFlow equation (1), (b) load (\hat{p}^L, \hat{q}^L) and nominal generation (\hat{p}^G, \hat{q}^G) , (c) inverter capacity C in (4), and (d) safety limits $\underline{\nu}$, $\bar{\nu}$, and $\bar{\ell}$ in (5)
- 1: Compute universal voltage lower bound $\hat{\nu}^L = (\hat{v}^L)^2$ and current upper bound $\hat{\ell}^U = (\hat{i}^U)^2$ according to Algorithm 1
 - 2: Compute $W^{\text{iub}2}$ and $I^{\text{iub}2}$ according to (15), (24), and (25)
 - 3: Compute W^{vub} and $\hat{\nu}^{\text{nom}}$ according to (3), (28), and (31)
 - 4: Compute W^{vub} , M , and $I^{\text{iub}1}$ using (3), (15), and (36)
 - 5: Assemble matrix \mathbf{H} and vector \mathbf{b} according to (40)
 - 6: Apply Algorithm 2 to decide if instance (\mathbf{H}, \mathbf{b}) is feasible. If feasible, rollout schedule is defined by output \mathbf{x}

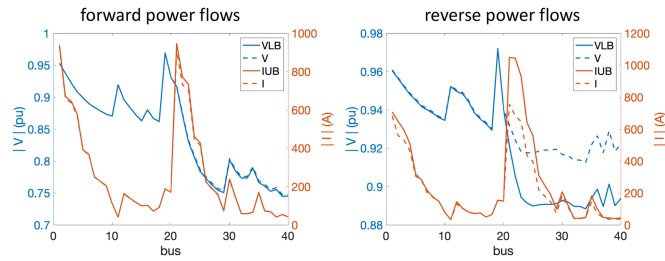


Fig. 2. Accuracy demonstration of universal voltage and current bounds with the CIGRE LV network. Solid blue lines are $\hat{\nu}^L$ obtained by Algorithm 1. Dotted blue lines are $\nu(\bar{p}, \bar{q})$, upper bounds of $\hat{\nu}^L$. Solid orange lines are $\hat{\ell}^U$ obtained by Algorithm 1. Dotted orange lines are $\ell(\bar{p}, \bar{q})$, lower bounds of $\hat{\ell}^U$. Left: typical load with forward power flows. The average relative error in voltage is 0.17% and the relative error in current is 1.1%. Right: zero net active power load with significant reverse power flows. The average relative error in voltage is 1.4%, and the relative error in current is 15%.

$\nu(p, q) \geq \nu^L \geq \hat{\nu}^L$ and $\ell(p, q) \leq \ell^U \leq \hat{\ell}^U$. Hence, if the difference $\nu(p, q) - \hat{\nu}^L$ is small, then the error $\nu^L - \hat{\nu}^L$ must be small. A similar relation holds for $\ell(p, q)$, ℓ^U , and $\hat{\ell}^U$. A suitable choice of (p, q) is $(\bar{p}, \bar{q}) = (-\hat{p}^L, -\hat{q}^L - C)$ because (\bar{p}, \bar{q}) minimizes ν in (2a) if the loss term $M\ell$ is ignored (also making ℓ large in (2b) consequently). For the CIGRE LV benchmark, we compare $\hat{\nu}^L$ and $\nu(\bar{p}, \bar{q})$ (respectively, $\hat{\ell}^U$ and $\ell(\bar{p}, \bar{q})$) in two cases. In both the cases, the total inverter power rating is the total active power load. However, for the first case, the default load is considered resulting in typical forward power flows. On the other hand, for the second case, “background” distributed generation is added so that the total net active power load is zero. This is a situation with significant distributed generation and reverse power flows. Fig. 2 indicates that $\hat{\nu}^L$ and $\hat{\ell}^U$ obtained by Algorithm 1 are of acceptable quality for the CIGRE benchmark.

The aforementioned experiment is repeated for selected benchmarks from [22] and [24]. Table I suggests similar conclusions. Except for the approximately 10% average relative error for current upper bound in scenarios with significant reverse power flows, generally, the error is negligible. We note that, in the aforementioned experiment, on average, Algorithm 1 converges in 6.5 fixed-point iterations, and the maximum number

TABLE I
AVERAGE (OVER BUSES OR LINES) RELATIVE ERROR OF UNIVERSAL VOLTAGE LOWER BOUNDS AND CURRENT UPPER BOUNDS

case name	forward power flows		reverse power flows	
	rel err ν	rel err ℓ	rel err ν	rel err ℓ
Matpower 33bw	0.006%	0.21%	0.059%	5.6%
Matpower 85	0.050%	0.38%	0.064%	1.3%
Matpower 118zh	0.0050%	0.22%	0.017%	11%
Matpower 141	0.0018%	0.047%	0.023%	3.9%
REDS 135+8	≈ 0	0.022%	0.0070%	7.3%
REDS 201+3	≈ 0	0.012%	≈ 0	4.5%
REDS 873+7	≈ 0	0.011%	≈ 0	4.0%
REDS 10476+84	≈ 0	0.010%	≈ 0	4.2%

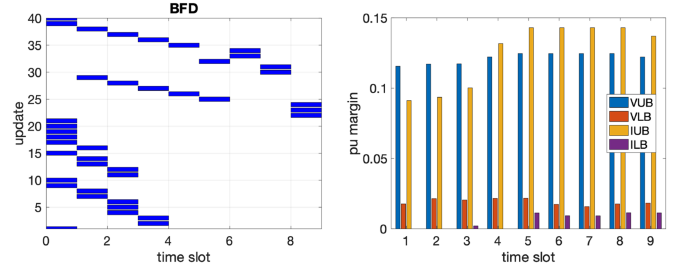


Fig. 3. Results of CIGRE heavy load scenario case study concerning (41). Left: Gantt diagram. Right: voltage and current margins.

of iterations is nine over all the cases. The relative tolerance in Algorithm 1 is 10^{-6} .

C. CIGRE 44-Bus LV Distribution Grid—Heavy Load

In this case study, we assume that the power rating of each smart inverter is $0.7/N = 0.015$ p.u. (the system’s total inverter rating is 0.7 p.u.). The (linear) voltage limits in (5) are 0.85 and 1.1 p.u. for all the buses. For each line, the (linear) current upper limit is the maximum of 600 A and 1.5 times the nominal current of the line in the default load setting.

To simulate the scenario with heavy load, the nominal distributed generation is zero (i.e., $\hat{p}^G = 0$ and $\hat{q}^G = 0$), and the load is 130% of the default load (both for \hat{p}^L and \hat{q}^L) so that the total active power load is 0.89 p.u. With all relevant data specified, the rollout problem in (41) is set up and solved using Algorithm 2. The Gantt diagram of the obtained update schedule is shown in Fig. 3 (left), utilizing nine time slots. Fig. 3 (right) also shows the voltage and current margins for each of the time slots defined as follows: for any time slot with assigned update $\mathcal{I} \subseteq \mathcal{N}$, the following problems are solved:

$$\nu_n^L(\mathcal{I}) := \min_{p,q,P,Q,\nu,\ell} \nu_n \quad \text{s.t. (1), } (p, q) \in \mathcal{S}(\mathcal{I}) \text{ in (6)} \quad (42a)$$

$$\nu_n^U(\mathcal{I}) := \max_{p,q,P,Q,\nu,\ell} \nu_n \quad \text{s.t. (1), } (p, q) \in \mathcal{S}(\mathcal{I}) \text{ in (6)} \quad (42b)$$

$$\ell_n^U(\mathcal{I}) := \max_{p,q,P,Q,\nu,\ell} \ell_n \quad \text{s.t. (1), } (p, q) \in \mathcal{S}(\mathcal{I}) \text{ in (6)} \quad (42c)$$

to obtain the worst case voltage for each nonreference bus $n \in \mathcal{N}$ and the worst case current for each line $n \in \mathcal{L}$ for \mathcal{I} associated with the time slot. Then, the worst case quantities from (42) are compared with the safety limits in (5) to obtain the relevant margins (nonnegative values mean limit satisfaction). It can be seen that the rollout schedule obtained by solving the proposed

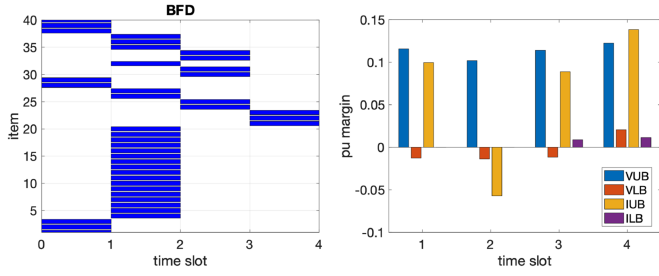


Fig. 4. Results of CIGRE heavy load scenario case study concerning (43). Left: Gantt diagram. Right: voltage and current margins.

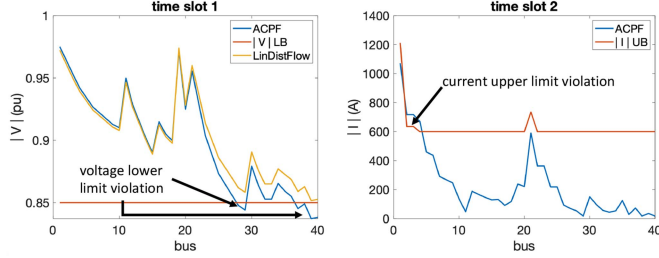


Fig. 5. Voltage and current profiles showing safety limit violations of the rollout schedule due to (43). Left: voltage profile of time slot 1. Right: current profile of time slot 2. In both the subfigures, ACPF denotes the voltage and current profiles obtained from ac power flow simulation using Matpower [24] with some power injections $(p, q) \in \mathcal{S}(\mathcal{I})$ for \mathcal{I} conforming to the time slot. In the left, LinDistFlow denotes the minimum voltage due to linearized DistFlow equation (i.e., $(\hat{v}^{\text{nom}} - \tilde{W}^{\text{vub}} \mathbf{1}_T)^{1/2}$), which suggests false limit satisfaction.

problem in (41) with Algorithm 2 indeed satisfies all the safety limits.

As a comparison, the linearized DistFlow version of (41) is considered. The following problem (i.e., an extension of the linearized DistFlow rollout problem in [7]) is solved to obtain the corresponding rollout schedule:

$$\begin{aligned} & \text{minimize} \quad T \\ & \text{s.t.} \quad \mathcal{I}_1, \mathcal{I}_2, \dots, \mathcal{I}_T \text{ partition } \mathcal{N} \\ & \quad \tilde{\mathbf{H}} \mathbf{1}_{\mathcal{I}_j} \leq \tilde{\mathbf{b}}, \quad j \in \{1, \dots, T\} \end{aligned} \quad (43)$$

where

$$\tilde{\mathbf{H}} = \begin{bmatrix} W^{\text{vub}} \\ \tilde{W}^{\text{vub}} \end{bmatrix}, \quad \tilde{\mathbf{b}} = \begin{bmatrix} \bar{v} - \hat{v}^{\text{nom}} \\ \hat{v}^{\text{nom}} - \underline{v} \end{bmatrix}$$

with W^{vub} and \hat{v}^{nom} defined in (31) and $\tilde{W}^{\text{vub}} := 2XD_C + 2RD_{\hat{p}^G} + 2XD_{\hat{q}^G}$ (i.e., the matrix in (36) with the last term removed). Fig. 4 shows the Gantt diagram and the margins of the rollout schedule obtained by solving (43) using Algorithm 2. The linearized DistFlow schedule requires four versus nine time slots required by the proposed problem in (41). However, Fig. 4 shows negative voltage lower limit margins for the first three time slots, as well as negative current upper limit margin for the second time slot. This indicates voltage and current limit violations, as illustrated in Fig. 5. This justifies the apparently more complicated safety constraints proposed in (40) for the rollout scheduling problem, as opposed to the simpler linearized DistFlow version introduced earlier in [7].

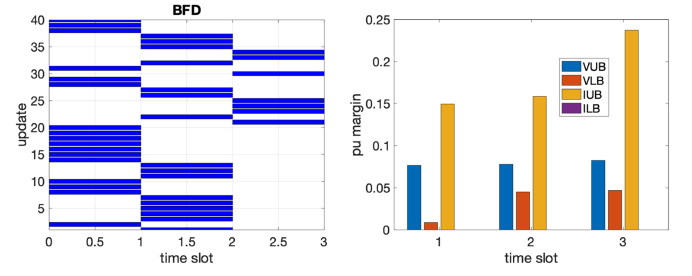


Fig. 6. Results of CIGRE significant DG scenario case study concerning (41). Left: Gantt diagram. Right: voltage and current margins.

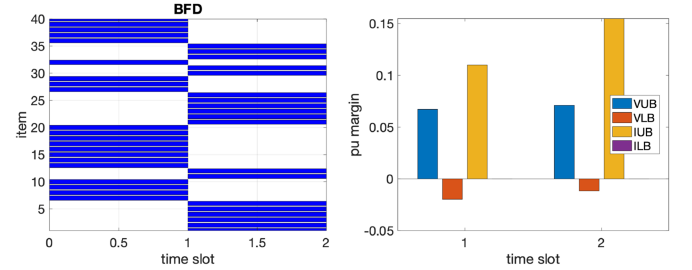


Fig. 7. Results of CIGRE significant DG scenario case study concerning (43). Left: Gantt diagram. Right: voltage and current margins.

D. CIGRE 44-Bus LV Distribution Grid—Significant distributed generation (DG)

Opposite to Section V-C, this section considers the scenario with significant distributed generation and reverse power flows. In this scenario, the loads \hat{p}^L and \hat{q}^L retain their default values. The total inverter power rating is increased to 0.9 p.u. (versus total active power load of 0.69 p.u.). In addition, all the inverters nominally generate active power according to their ratings (i.e., $\hat{p}^G = 0.9/N \mathbf{1}$ p.u. and $\hat{q}^G = \mathbf{0}$ p.u.). The voltage lower limits, current limits, and current upper limits are all uniform over their elements. They are, respectively, 0.9 p.u., 1.2 p.u., and 800 A. Fig. 6 shows the Gantt diagram and the voltage and current margins for the rollout schedule obtained by the proposed method. In comparison, Fig. 7 shows the analogous results for the rollout schedule obtained by solving the linearized DistFlow rollout scheduling problem in (43). The linearized version requires one fewer time slot. However, voltage lower limits are again violated.

E. REDS 10,476-Bus/83-Feeder Distribution Grid

This benchmark is the largest example from the repository in [22], containing 10,476 nonreference buses and 83 feeders. We assume that inverters are installed in all the nonreference buses with a uniform rating. The total rating is 68.4 p.u., while the total active power load is 38.2 p.u. The safety limits are 0.9 p.u., 1.1 p.u., and 600 A for voltage and current (uniform over all the relevant elements). Solving the proposed rollout scheduling problem in (41) with this benchmark yields a schedule with the Gantt diagram and voltage and current margins shown in Fig. 8. The schedule requires four time slots to arrange all 10,476 updates, and the total time to set up and solve the rollout problem is less than 3 s (with Algorithm 2 requiring about 50% of the time). It can be verified that it is impossible to assign all the updates to one single time slot, excluding the trivial solution. To

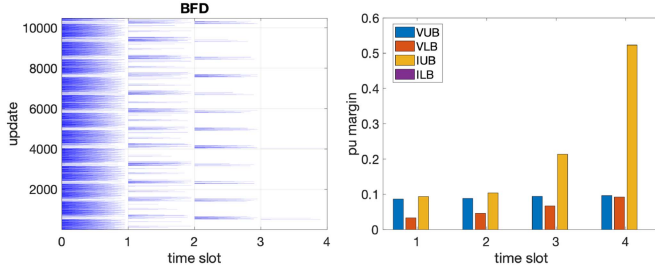


Fig. 8. Results of 10,476-bus benchmark case study concerning (41). Left: Gantt diagram. Right: voltage and current margins.

TABLE II
COMPUTATION TIME AND NUMBER OF TIME SLOTS OF ROLLOUT
SCHEDULES OBTAINED BY THE PROPOSED METHOD

case name	time	slots	case name	time	slots
REDS 13+3	4.98 ms	1	33bw	4.35 ms	2
REDS 29+1	4.77 ms	2	51he	6.08 ms	1
REDS 32+1	4.84 ms	2	69	8.85 ms	4
REDS 83+11	6.35 ms	4	85	11.6 ms	6
REDS 135+8	10.9 ms	3	94pi	20.5 ms	11
REDS 201+3	30.7 ms	4	118zh	11.3 ms	8
REDS 873+7	151 ms	2	136ma	10.2 ms	3
REDS 10476+84	1.97 s	2	141	20.1 ms	5

obtain a rollout schedule with four time slots using enumeration would require checking the second constraints in (41) more than 5×10^{14} times. Suppose that each such examination requires 2×10^{-5} s (based on the time recording in our experiment). The total enumeration would require more than 300 years, which is not suitable for real-time applications.

F. Summary for all the Cases

Instances of the rollout problem in (41) derived from benchmarks from the REDS repository [22] and Matpower 7.1 [24] are solved using Algorithm 2. In all these cases, the voltage limits are 0.9 and 1.1 p.u. for all the buses, and the current upper limits are 600 A or above (depending on the nominal current due to default load setting). Typical load and inverter rating scenarios are assumed, which could be different from the scenario in Section V-E. Table II summarizes the results of the rollout schedules.

We note that the computation time for all the cases, including the 10,476-bus example, remains modest. In addition, the number of time slots used by the schedules is not excessive in comparison to the “sequential schedules,” where each time slot holds exactly one update. This suggests, despite all the approximations, that our proposed method is able to obtain safe and effective rollout schedules in real time, even for distribution networks of substantial size.

VI. CONCLUSION

Rapid system-level deployment and reconfiguration of control software is gaining prominence as power systems evolve to integrate large amount of inverter-based distributed resources with little or zero inertia, thus requiring more dynamic control. As the power system is part of the critical infrastructure, its control software update should be carefully executed to prevent major reliability incidents. This requires proper accounting of the cyber-physical relationship between software behavior (e.g.,

update failure) and power system operational state (e.g., voltage and current). For radial distribution systems, the DistFlow equation accurately describes the underlying physics. However, the intrinsic nonlinearity renders the equation unfit for any real-time decision making. The linearized DistFlow equation is generally a good balance between model complexity and fidelity. Nevertheless, as demonstrated in this article, it may not be accurate enough to maintain system safety standard when conflicting operational goals are present (e.g., makespan minimization for software update rollout). This necessitates innovations in distribution system modeling exemplified by the proposed linearized relationships between software update failure and the worst case voltage/current, which are theoretically justifiable with our detailed analysis and practically viable as demonstrated by benchmarks with complexity far exceeding the norm in the literature (e.g., 10,476-bus versus 33-bus). Besides the software update rollout problem studied here, these linearized relationships can potentially enable online distribution system contingency monitoring and redispatch. In addition, to enable realistic large-scale applications, computational innovations are needed. The reformulation of the rollout problem into a bin-packing problem amenable to efficient best fit decreasing algorithm is one such example. In addition, the fixed-point iteration procedure to quickly estimate the worst case voltages and currents crucially enables the real-time large-scale application of the proposed rollout scheduling procedure. The investigation to apply the fixed-point procedure to other situations (e.g., quick power flow analysis, contingency evaluation, etc.) appears to be worthwhile.

APPENDIX A AUXILIARY STATEMENTS

Lemma 1: Let $a + jb \in \mathbb{C}$ and $r > 0$ be given. Also, let

$$\theta := \begin{cases} \tan^{-1}(b/a), & \text{if } a \neq 0 \text{ or } b \neq 0 \\ 0, & \text{if } a = b = 0. \end{cases}$$

Then, the maximum objective value of the following problem:

$$\begin{aligned} & \underset{p,q}{\text{maximize}} \quad |(a + jb) - (p + jq)| \\ & \text{s.t.} \quad p \geq 0, \quad |p + jq| \leq r \end{aligned} \quad (44)$$

is

$$\max \{ |a + j(b+r)|, |a + j(b-r)|, |a + jb - re^{j\theta}| \} \quad (45)$$

Proof: First, we consider the special case where $a = b = 0$. In this case, the maximum of (44) is r , and all three terms in (45) are equal to r . Thus, the statement is trivially true when $a = b = 0$. Next, we consider the case where at least one of a or b is not zero. Let (p^*, q^*) be the argument of maximum of (44). We claim that $|p^* + jq^*| = r$. Suppose not, then there exists $\Delta q \neq 0$ such that $|p^* + j(q^* + \Delta q)| = r$ and $|b - q^* - \Delta q| > |b - q^*|$. This implies that $(p^*, q^* + \Delta q)$ is a better feasible solution than (p^*, q^*) , leading to a contradiction. Hence, (44) can be reparameterized as

$$\underset{\alpha \in [-\pi/2, \pi/2]}{\text{maximize}} \quad |r \cos(\alpha) - a + j(r \sin(\alpha) - b)|. \quad (46)$$

It can be verified that the only stationary point of the objective for $-\pi/2 \leq \alpha \leq \pi/2$ is at $\tan^{-1}(b/a)$ (well defined because

at least one of a or b is nonzero). Therefore, the maximum of (46) can be attained only at one of the three possibilities: $-\pi/2$, $\pi/2$, and $\tan^{-1}(b/a)$, which is θ according to the statement definition. Thus, the maximum of (46) must be one of the following: $|a + j(b+r)|$, $|a + j(b-r)|$, or $|a + jb - r(\cos(\theta) + j\sin(\theta))|$, same as (45). ■

Lemma 2: Let \mathcal{D} be the descendant matrix defined in Section II-B. Let $\hat{p}^L, \hat{q}^L, \hat{p}^G, \hat{q}^G \in \mathbb{R}^N$ be given. Then, for any $n \in \mathcal{N}$ and $\mathcal{I} \subseteq \mathcal{N}$, the optimal objective value of

$$\begin{aligned} & \underset{p^G, q^G}{\text{maximize}} \quad e_n^\top |\mathcal{D}(p + jq)| \\ & \text{s.t.} \quad (p, q) \in \mathcal{S}(\mathcal{I}) \quad \text{in (6)} \end{aligned} \quad (47)$$

is

$$\max \{ |a + j(b+r)|, |a + j(b-r)|, |a + jb - re^{j\theta}| \} \quad (48)$$

where

$$\begin{aligned} a &:= e_n^\top \mathcal{D}(\hat{p}^L - (\mathbf{1} - \mathbf{1}_{\mathcal{I}}) \odot \hat{p}^G) \\ b &:= e_n^\top \mathcal{D}(\hat{q}^L - (\mathbf{1} - \mathbf{1}_{\mathcal{I}}) \odot \hat{q}^G) \\ r &:= e_n^\top \mathcal{D}(\mathbf{1}_{\mathcal{I}} \odot C) \\ \theta &:= \begin{cases} \tan^{-1}(b/a), & \text{if } a \neq 0 \text{ or } b \neq 0 \\ 0, & \text{if } a = b = 0. \end{cases} \end{aligned} \quad (49)$$

Proof: Note that for any $x \in \mathbb{R}^N$, $e_n^\top \mathcal{D}x = \sum_{m \in d(n)} x_m$. Hence, with $(p, q) \in \mathcal{S}(\mathcal{I})$, the objective function can be written as

$$\begin{aligned} & \left| \sum_{m \in d(n)} (p_m^G - \hat{p}_m^L) + j \sum_{m \in d(n)} (q_m^G - \hat{q}_m^L) \right| \\ &= \left| \left(\sum_{m \in d(n)} \hat{p}_m^L - \sum_{m \in d(n) \setminus \mathcal{I}} \hat{p}_m^G \right) + j \left(\sum_{m \in d(n)} \hat{q}_m^L - \sum_{m \in d(n) \setminus \mathcal{I}} \hat{q}_m^G \right) \right. \\ & \quad \left. - \sum_{m \in d(n) \cap \mathcal{I}} (p_m^G + jq_m^G) \right|. \end{aligned}$$

According to the definition of $\mathcal{S}(\mathcal{I})$, for each $m \in \mathcal{I}$, the set of all possible $p_m^G + jq_m^G$ is the half-disk $\Omega(C_m) := \{(x, y) \mid x \geq 0, |x + jy| \leq C_m\}$. Thus, the set of all possible sum $\sum_{m \in d(n) \cap \mathcal{I}} (p_m^G + jq_m^G)$ is the Minkowski sum $\sum_{m \in d(n) \cap \mathcal{I}} \Omega(C_m)$, which can be verified to be the half-disk $\Omega(\sum_{m \in d(n) \cap \mathcal{I}} C_m)$. By noting that

$$\begin{aligned} \sum_{m \in d(n)} \hat{p}_m^L - \sum_{m \in d(n) \setminus \mathcal{I}} \hat{p}_m^G &= e_n^\top \mathcal{D}(\hat{p}^L - (\mathbf{1} - \mathbf{1}_{\mathcal{I}}) \odot \hat{p}^G) = a \\ \sum_{m \in d(n)} \hat{q}_m^L - \sum_{m \in d(n) \setminus \mathcal{I}} \hat{q}_m^G &= e_n^\top \mathcal{D}(\hat{q}^L - (\mathbf{1} - \mathbf{1}_{\mathcal{I}}) \odot \hat{q}^G) = b \\ \sum_{m \in d(n) \cap \mathcal{I}} C_m &= e_n^\top \mathcal{D}(\mathbf{1}_{\mathcal{I}} \odot C) = r \end{aligned}$$

problem (47) can be rewritten as maximizing $|a + jb - (x + jy)|$ with respect to x and y , such that $x \geq 0$ and $|x + jy| \leq r$.

According to Lemma 1, the maximum objective value is given in (48). ■

APPENDIX B PROOF OF PROPOSITION 1

It suffices to show the following induction: suppose that for some k , it holds that $v^k \leq (\nu^L)^{1/2}$ and $i^k \geq (\ell^U)^{1/2}$; then, $v^{k+1} \leq (\nu^L)^{1/2}$ and $i^{k+1} \geq (\ell^U)^{1/2}$.

For any $n \in \mathcal{N}$, according to (2) and (7), there exists some optimizing pair $(p^{G(n)}, q^{G(n)}) \in \mathbb{R}^N \times \mathbb{R}^N$ satisfying (4) with (p^G, q^G) interpreted as $(p^{G(n)}, q^{G(n)})$ such that $\nu_n^L = v_n^2$ with v (together with i) satisfying

$$\begin{aligned} v &= \left(\nu_0 \mathbf{1} + 2R(p^{G(n)} - \hat{p}^L) + 2X(q^{G(n)} - \hat{q}^L) + Mi^2 \right)^{1/2} \\ i &= v^{-1} \odot |\mathcal{D}((p^{G(n)} - \hat{p}^L) + j(q^{G(n)} - \hat{q}^L)) - (\mathcal{D} - I)D_z i^2|. \end{aligned}$$

We note that all the entries of M are nonpositive. This, together with $i^k \geq (\ell^U)^{1/2}$ by the induction assumption and $(\ell^U)^{1/2} \geq i$ by (7), implies that $Mi^2 \geq M(i^k)^2$. Thus, for any $n \in \mathcal{N}$

$$\begin{aligned} (\nu_n^L)^{1/2} &= \left(\nu_0 + e_n^\top \left(2R(p^{G(n)} - \hat{p}^L) \right. \right. \\ & \quad \left. \left. + 2X(q^{G(n)} - \hat{q}^L) + Mi^2 \right) \right)^{1/2} \\ &\geq \left(\nu_0 + e_n^\top \left(2R(p^{G(n)} - \hat{p}^L) \right. \right. \\ & \quad \left. \left. + 2X(q^{G(n)} - \hat{q}^L) + M(i^k)^2 \right) \right)^{1/2} \\ &\geq \min_{(p^G, q^G) \text{ satisfying (4)}} \left\{ \left(\nu_0 + e_n^\top \left(2R(p^G - \hat{p}^L) \right. \right. \right. \\ & \quad \left. \left. + 2X(q^G - \hat{q}^L) + M(i^k)^2 \right) \right)^{1/2} \Big\} \\ &= e_n^\top (\nu_0 \mathbf{1} - 2R\hat{p}^L - 2X(\hat{q}^L + C) + M(i^k)^2)^{1/2} \\ &= v_n^{k+1}. \end{aligned}$$

Similarly, there exists some $(p^{g(n)}, q^{g(n)}) \in \mathbb{R}^N \times \mathbb{R}^N$ satisfying (4) with (p^G, q^G) interpreted as $(p^{g(n)}, q^{g(n)})$ such that $\ell_n^U = i_n^2$ with i (together with v) satisfying

$$\begin{aligned} v &= \left(\nu_0 \mathbf{1} + 2R(p^{g(n)} - \hat{p}^L) + 2X(q^{g(n)} - \hat{q}^L) + Mi^2 \right)^{1/2} \\ i &= v^{-1} \odot |\mathcal{D}((p^{g(n)} - \hat{p}^L) + j(q^{g(n)} - \hat{q}^L)) - (\mathcal{D} - I)D_z i^2|. \end{aligned}$$

Since all the entries of $(\mathcal{D} - I)D_r$ and $(\mathcal{D} - I)D_x$ are non-negative, $i^k \geq (\ell^U)^{1/2}$ and $v^k \leq (\nu^L)^{1/2}$ by the induction assumption, and $(\ell^U)^{1/2} \geq i$ and $(\nu^L)^{1/2} \leq v$ by (7), it holds that: 1) $v \geq v^k$; 2) $(\mathcal{D} - I)D_r i^2 \leq (\mathcal{D} - I)D_r (i^k)^2$; and 3) $(\mathcal{D} - I)D_x i^2 \leq (\mathcal{D} - I)D_x (i^k)^2$. Hence, for any $n \in \mathcal{N}$

$$\begin{aligned} (\ell_n^U)^{1/2} &= v_n^{-1} e_n^\top |\mathcal{D}((p^{g(n)} - \hat{p}^L) + j(q^{g(n)} - \hat{q}^L)) \\ & \quad - (\mathcal{D} - I)D_z i^2| \\ &\leq (v_n^k)^{-1} e_n^\top (|\mathcal{D}((p^{g(n)} - \hat{p}^L) + j(q^{g(n)} - \hat{q}^L))| \\ & \quad + |(\mathcal{D} - I)D_z (i^k)^2|) \end{aligned}$$

$$\leq \max_{(p^g, q^g) \text{ satisfying (4)}} ((v_n^k)^{-1} |e_n^\top \mathcal{D}((p^g - \hat{p}^L) + j(q^g - \hat{q}^L))|) + (v_n^k)^{-1} e_n^\top |(\mathcal{D} - I)D_z(i^k)^2|.$$

According to Lemma 2 with $\mathcal{I} = \mathcal{N}$, the first term above can be rewritten as $\bar{S}_n := \max\{|a_n + j(b_n + r_n)|, |a_n + j(b_n - r_n)|, |a_n + j b_n - r_n(\cos(\theta_n) + j \sin(\theta_n))|\}$, where $a_n = e_n^\top \mathcal{D} \hat{p}^L$, $b_n = e_n^\top \mathcal{D} \hat{q}^L$, $r_n = e_n^\top \mathcal{D} C$, and $\theta_n = \tan^{-1}(b_n/a_n)$ if at least one of a_n and b_n is nonzero (otherwise $\theta_n = 0$). Therefore

$$(\ell_n^U)^{1/2} \leq (v_n^k)^{-1} (\bar{S}_n + e_n^\top |(\mathcal{D} - I)D_z(i^k)^2|) = i_n^{k+1}.$$

ACKNOWLEDGMENT

The authors would like to thank Dr. Sei Zhen Khong for helpful discussions.

REFERENCES

- [1] K. E. Hemsley and R. E. Fisher, "History of industrial control system cyber incidents," Idaho Nat. Lab., Idaho Falls, ID, USA, Tech. Rep. INL/CON-18-44411-Rev002, Dec. 2018.
- [2] A. A. Cárdenas, "Cyber-physical systems security," *CyBOK, Knowl. Area Rep.*, 2019.
- [3] A. A. Cardenas, S. Amin, and S. Sastry, "Research challenges for the security of control systems," *HotSec*, vol. 5, no. 15, p. 1158, 2008.
- [4] P. Griffioen, S. Weerakkody, and B. Sinopoli, "A moving target defense for securing cyber-physical systems," *IEEE Trans. Autom. Control*, vol. 66, no. 5, pp. 2016–2031, May 2021.
- [5] A. Kanellopoulos and K. G. Vamvoudakis, "A moving target defense control framework for cyber-physical systems," *IEEE Trans. Autom. Control*, vol. 65, no. 3, pp. 1029–1043, Mar. 2020.
- [6] R. Romagnoli, B. H. Krogh, D. de Niz, A. D. Hristozov, and B. Sinopoli, "Software rejuvenation for safe operation of cyber-physical systems in the presence of run-time cyberattacks," *IEEE Trans. Control Syst. Technol.*, vol. 31, no. 4, pp. 1565–1580, Jul. 2023.
- [7] M. G. de Medeiros, K. C. Sou, and H. Sandberg, "Minimum-time secure rollout of software updates for controllable power loads," *Electr. Power Syst. Res.*, vol. 189, 2020, Art. no. 106797.
- [8] K. C. Sou and H. Sandberg, "Resilient scheduling of control software updates in power distribution systems," in *Proc. IEEE 61st Conf. Decis. Control*, 2022, pp. 6146–6153.
- [9] A. Froger, M. Gendreau, J. E. Mendoza, É. Pinson, and L.-M. Rousseau, "Maintenance scheduling in the electricity industry: A literature review," *Eur. J. Oper. Res.*, vol. 251, no. 3, pp. 695–706, 2016.
- [10] M. E. Baran and F. F. Wu, "Optimal capacitor placement on radial distribution systems," *IEEE Trans. Power Del.*, vol. 4, no. 1, pp. 725–734, Jan. 1989.
- [11] M. E. Baran and F. F. Wu, "Network reconfiguration in distribution systems for loss reduction and load balancing," *IEEE Trans. Power Del.*, vol. 4, no. 2, pp. 1401–1407, Apr. 1989.
- [12] C. Coffrin, H. L. Hijazi, and P. Van Hentenryck, "Strengthening the SDP relaxation of ac power flows with convex envelopes, bound tightening, and valid inequalities," *IEEE Trans. Power Syst.*, vol. 32, no. 5, pp. 3549–3558, Sep. 2017.
- [13] D. K. Molzahn and L. A. Roald, "Towards an ac optimal power flow algorithm with robust feasibility guarantees," in *Proc. Power Syst. Comput. Conf.*, 2018, pp. 1–7.
- [14] R. Louca and E. Bitar, "Robust ac optimal power flow," *IEEE Trans. Power Syst.*, vol. 34, no. 3, pp. 1669–1681, May 2019.
- [15] X. Wu, A. J. Conejo, and N. Amjadi, "Robust security constrained ACOPF via conic programming: Identifying the worst contingencies," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 5884–5891, Nov. 2018.
- [16] C. Coffrin, R. Bent, B. Tasseff, B. Sundar, and S. Backhaus, "Relaxations of ac maximal load delivery for severe contingency analysis," *IEEE Trans. Power Syst.*, vol. 34, no. 2, pp. 1450–1458, Mar. 2019.
- [17] J. Liu, B. Cui, D. K. Molzahn, C. Chen, X. Lu, and F. Qiu, "Optimal power flow in dc networks with robust feasibility and stability guarantees," *IEEE Trans. Control Netw. Syst.*, vol. 9, no. 2, pp. 904–916, Jun. 2022.
- [18] "The grid edge revolution: Innovative drivers towards net-zero energy," Siemens white paper, 2019. [Online]. Available: <https://new.siemens.com/global/en/company/topic-areas/smart-infrastructure/grid-edge/white-paper-grid-edge-net-zero-energy-drivers.html>
- [19] F. Kintzler et al., "Large scale rollout of smart grid services," in *Proc. Glob. Internet Things Summit*, 2018, pp. 1–7.
- [20] D. Shelar and S. Amin, "Security assessment of electricity distribution networks under DER node compromises," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 1, pp. 23–36, Mar. 2017.
- [21] D. Shelar, S. Amin, and I. A. Hiskens, "Evaluating resilience of electricity distribution networks via a modification of generalized benders decomposition method," *IEEE Trans. Control Netw. Syst.*, vol. 8, no. 3, pp. 1225–1238, Sep. 2021.
- [22] R. Kavasseri and C. Ababei, "Efficient network reconfiguration using minimum cost maximum flow-based branch exchanges and random walks-based loss estimations," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 30–37, 2010. [Online]. Available: <http://www.dejazzer.com/reds.html>
- [23] L. Thurner et al., "Pandapower—An open-source Python tool for convenient modeling, analysis, and optimization of electric power systems," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6510–6521, Nov. 2018.
- [24] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2011.



Kin Cheong Sou (Member, IEEE) received the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2008.

From 2008 to 2010, he was a Postdoctoral Researcher with Lund University, Lund, Sweden. From 2010 to 2012, he was a Postdoctoral Researcher with the KTH Royal Institute of Technology, Stockholm, Sweden. From 2013 to 2016, he was an Assistant Professor with the Department of Mathematical Sciences, Chalmers University of Technology, Gothenburg, Sweden, and the University of Gothenburg, Gothenburg. He is currently an Associate Professor with the Department of Electrical Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan. His research interests include decision-making and computation techniques for power systems applications.



Henrik Sandberg (Fellow, IEEE) received the M.Sc. degree in engineering physics and the Ph.D. degree in automatic control from Lund University, Lund, Sweden, in 1999 and 2004, respectively.

From 2005 to 2007, he was a Postdoctoral Scholar with the California Institute of Technology, Pasadena, CA, USA. In 2013, he was a Visiting Scholar with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, USA. He has also held visiting appointments with the Australian National University, Canberra, ACT, Australia, and the University of Melbourne, Melbourne, VIC, Australia. He is currently a Professor with the Division of Decision and Control Systems, KTH Royal Institute of Technology, Stockholm, Sweden. His current research interests include security of cyber-physical systems, power systems, model reduction, and fundamental limitations in control.

Dr. Sandberg is the recipient of the Best Student Paper Award from IEEE Conference on Decision and Control in 2004, an Ingvar Carlsson Award from the Swedish Foundation for Strategic Research in 2007, and a Consolidator Grant from the Swedish Research Council in 2016. He was on the Editorial Board of IEEE TRANSACTIONS ON AUTOMATIC CONTROL and *Automatica*.