Online Decentralized Tracking for Nonlinear Time-Varying Optimal Power Flow of Coupled Transmission–Distribution Grids

Wentian Lu, Member, IEEE, Kaijun Xie, Mingbo Liu, Member, IEEE, Xiaogang Wang, and Lefeng Cheng, Member, IEEE

Abstract—The coordinated transmission–distribution alternating current optimal power flow (TD-ACOPF) has become crucial to handle problems related to high penetration of renewable energy sources (RESs). However, obtaining all system details and solving TD-ACOPF centrally is sometimes less preferable in practice due to privacy concerns. Intermittent RESs and uncontrollable loads can swiftly change the operating condition of the power grid. Existing decentralized optimization methods can seldom track the optimal solutions of time-varying TD-ACOPFs. Here, we develop a continuous-time dynamic approach based on the primal-dual interior point method to online track the saddle points of the Lagrangian for time-varying ACOPFs from a control system perspective. To improve tracking accuracy, a prediction term denoted by the partial derivative with respect to time is added to the dynamic system. Next, a decentralized implementation for solving the dynamic system is designed based on only a few information exchanges with respect to boundary variables. Moreover, the proposed continuous-time dynamic approach is derived from the first-order optimality condition of the ACOPF model (without approximation) for coping with the nonlinear power flow equation model without the need for real-time measurements. Numerical test results reveal the effectiveness and fast-tracking performance of the proposed algorithm.

Index Terms—Alternating current optimal power flow (ACOPF), coupled transmission–distribution grids, dynamic system, fast tracking, online decentralized optimization, renewable energy sources (RESs).

NOMENCLATURE

Sets and Indices
\( N \) Set of buses.
\( i \) Index of bus.
\( N(i) \) Set of buses connected to bus \( i \).
\( t \) Index of time slot.
\( N_g \) Index of conventional generator.
\( N_{\text{res}} \) Index of RES.
\( \Omega \) Set of lines.
\( \Omega_{\text{TD}} \) Set of distribution systems.

Parameters
\( P_i^t(t), Q_i^t(t) \) Active and reactive power demand at bus \( i \) at time \( t \).
\( P^\text{max}_i(t) \) Maximum available active power output of the RES connected with bus \( i \).
\( \theta_i \) Maximum allowable power-factor angle of the RES at bus \( i \).
\( S_i^\text{av} \) Rated apparent power of the RES at bus \( i \).
\( V_i, \bar{V}_i \) Lower and upper bounds of voltage magnitude at bus \( i \).
\( P_i^g, \bar{P}_i^g \) Lower and upper active power output bounds of the generator at bus \( i \).
\( c_{g2,i}, c_{g1,i}, c_{g0,i} \) Quadratic, linear, and constant terms in the quadratic generation cost function of the generator at bus \( i \).
\( c_i^P, c_i^Q \) Quadratic and linear terms in the quadratic penalization function for the active power curtailment and reactive power generation of the RES at bus \( i \).
\( S_{ij} \) Maximum transmission capacity of the branch \( ij \).
\( g_{ij}, b_{ij} \) Conductance and susceptance of the branch \( ij \).

Variables
\( P_i^g(t), Q_i^g(t) \) Real and reactive power outputs of the generator at bus \( i \) at time \( t \).
\( P_{\text{res}}^g(t), Q_{\text{res}}^g(t) \) Real and reactive power outputs of the RES at bus \( i \) at time \( t \).
\( e_i(t), f_i(t) \) Real and imaginary parts of the voltage at bus \( i \) at time \( t \).

I. INTRODUCTION

IN CURRENT power grid operations, the transmission system (TS) and distribution systems (DSs) are physically connected but separately managed by the TS and DS operators, respectively. Each operator considers the other system as an equivalent; that is, the TS operator regards the DS as an equivalent load, and the DS operator regards the TS as an infinite equivalent generator [1]. With the increasing incorporation of renewable energy sources (RESs), such as photovoltaics (PVs) and wind turbines (WTs), into distribution systems, regarding DSs as equivalent loads in TS alternating current optimal power flow (ACOPF) computation causes numerous problems, such as power mismatch at boundary buses, voltage rise, and resource wastage [2]. Thus, increased coordination is required between the TS and DSs.

To solve this problem, researchers first thought of using the centralized method to solve the coordinated problem directly. This centralized approach requires full knowledge of the integrated TS and DSs; for example, in China, the energy management system (EMS) of a centralized control center collects considerable amounts of information from each DS, and then sends operation and control signals to the subordinated distribution management systems (DMS). In this way, TS and DS models are merged into a complete model [3]. However, solving such a large-scale ACOPF centrally is sometimes less preferred in practice due to privacy concerns; for example, in the US, the generators, loads, and network information of each autonomous system are usually considered commercially sensitive. Therefore, a decentralized coordinated optimization method is preferable for solving coordinated transmission and distribution ACOPF (TD-ACOPF). This solution can converge to a centralized solution with limited information exchange.

From the distributed coordination perspective, decomposition approaches for solving ACOPF problems can be categorized into the following three categories: i) distributed implementation; ii) convergence guarantees for distributed ACOPFs; and iii) online ACOPF applications. The proposed approach is an online ACOPF application. Next, we concisely review existing approaches for ACOPF applications from the three categories above.

The first set of methods is based on the augmented Lagrangian decomposition (ALD) [4], in which the consistency constraint is relaxed into the Lagrangian function, and a quadratic term about consistency constraint is added to enhance convergence. However, this addition increases decomposition difficulty. Extensive ALD application methods for ACOPF problems, including the auxiliary problem principle (APP) for multi-area optimal power flow (OPF) computation [5], [6] and the alternating direction method of multipliers (ADMM) for fully distributed ACOPF [7], have been conducted to decompose this ALD quadratic term. However, the aforementioned algorithms typically require tuning parameters through trial and error, which can affect convergence and result in slow convergence or convergence failure.

Another subbranch of the first set of optimization methods is based on decomposing the first-order Karush–Kuhn–Tucker (KKT) optimality condition decomposition (OCD) of ACOPF problems, such as the approximate Newton direction (AND) for multi-area OPF [8], [9], and heterogeneous decomposition (HGD) for coordinated transmission and distribution OPF [10], [11], [12]. Although excellent computation accuracy and speed have been achieved, the convergence stability could be considerably affected by system size and system partitioning. In [13], we have cast the primal-dual interior point methods (PDIPM) to solve multi-area OPF problems in a fully distributed fashion (denoted by F-DIPM), which inherits the second-order convergence rate of the original PDIPM. However, these methods require computation of the second-order Hessian inverse at each iteration, which could be prohibitively time-consuming for large-scale systems.

The second set of methods is designed to search ACOPF solutions with certain convergence guarantees. This solution includes approaches based on generating feasible cut constraints, such as generalized benders decomposition (GBD) [14], distribution-cost-correction (DCC) [15], and the two-stage decomposition approach [16], which are used for decentralized models with a leader-follower structure, such as in coordinated transmission and distribution ACOPFs. To guarantee convergence, some linearized or convexified approximation techniques, such as second-order conic relaxations [14], [15], semi-definite relaxations [17], or a linearized DisFlow model [18], are required for the nonlinear ACOPF model. However, the optimal solution of the approximate model is not necessarily a feasible solution to the original ACOPF model. In [19], the augmented Lagrangian alternating direction inexact Newton (ALADIN) algorithm was applied directly for solving nonconvex ACOPFs, and its locally quadratic convergence can be ensured under some technical assumptions. In [20], a two-level ADMM algorithm was proposed for solving ACOPF problems, which enjoys global convergence properties under certain mild assumptions.

The two sets of iterative-convergence methods are appropriate for conventional power system applications that operate on a slow timescale but are not suitable for online applications on future smart grids. This inspires researchers to develop the third set of methods, that is, online distributed ACOPF algorithms that can track the time-varying loads and RESs on a faster timescale. In [21], an online ACOPF algorithm was proposed for distribution networks based on a gradient projection algorithm; this method achieved speedup by using the approximate gradient; in [22], an online algorithm was developed for time-varying ACOPFs based on quasi-Newton methods for using estimated second-order information to accelerate computation; these two centralized methods considerably reduce the computational effort in each iteration but require high-complexity implementations. In [23], an online distributed algorithm was designed to pursue the time-varying ACOPF solutions based on linearized AC power flow equations as well as the primal-dual gradient method. In [24], a distributed reactive power feedback controller was developed based on the dual ascent method as well as the linear approximation technique. In both [23] and [24], voltage measurements were leveraged from the whole
distribution grid to cope with inaccuracies introduced by a linear approximation of AC power flows. In [25], an online distributed ACOPF algorithm was designed based on a time-varying bilevel problem formulation and a feedback-based online implementation of primal-dual projected-gradient methods to address model inaccuracies while avoiding pervasive metering.

Another sub-branch of the third set of methods utilized continuous-time dynamic systems [26] for time-varying optimization problems. Inspired by the fact that strong Lagrangian duality holds for many practical OPF examples [17], Ma et al. in [27], [28], [29] proposed a continuous-time primal-dual gradient dynamical approach to search the saddle points of OPF problems. The most crucial property of this gradient dynamics approach is that it has a naturally distributed computing structure; that is, each bus in the power network functions as the basic computing unit and communicates with its neighbors. However, simulations have revealed that too much oscillation of voltage variables occurs before their convergence. An Euler–Newton continuation method was proposed in [30] for tracking the solution trajectories of time-varying parametric programming by introducing predictors based on higher-order Taylor expansions. In [31], a second-order dynamic system with a prediction-correction structure was designed for time-varying convex optimization problems. In this model, the prediction term considered the time-varying nature of the objective and constraint functions. In our previous studies [32], we have discussed and analyzed discrete/continuous-time online algorithm applications for time-varying OPFs in active distribution networks. However, the continuous-time prediction-correction dynamic system can only be performed in a centralized manner, and the distributed implementation of the discrete-time online algorithm was achieved based on synchronized communication among nodes [33].

In this article, we propose an online distributed optimization algorithm for timely tracking of the TD-ACOPF solutions. This approach is inspired by [31], in which only centralized tracking was achieved for time-varying convex optimization problems. Here, we investigate the potential of the continuous-time prediction-correction dynamic system [31] for tracking distributed TD-ACOPF solutions. We develop a continuous-time dynamic system based on the polynomial-time PDIPM. A prediction term, the partial derivative with respect to time, is derived from the first-order optimality condition of the ACPF model, the proposed continuous-time dynamic approach is derived from the first-order optimality condition of the ACOPF model (without approximation). Thus, the model can cope with the nonlinear power flow equation model without real-time measurements.

3) A PDIPM-based continuous-time dynamic system is developed to fast track the saddle-point trajectory of TD-ACOPFs from the control perspective, instead of directly searching KKT points.

The remainder of this article is organized as follows. In Section II, a time-varying centralized TD-ACOPF problem is formulated. In Section III, first, a centralized tracking approach based on the PDIPM is proposed to solve the TD-ACOPF problem. Next, the online distributed tracking algorithm is described in detail, including basic formula derivation, computation diagrams, and convergence analysis. In Section IV, simulation results on illustrative systems are presented. The conclusion is presented in Section V.

II. TD-ACOPF PROBLEM FORMULATION

A. Primary TD-ACOPF Model

A typical topology of the integrated transmission and distribution system is displayed in Fig. 1, in which TS and DS are connected by the tie line. This connection is defined as a coupling boundary. The TS and DS are physically connected but separately managed by the TS operator and DS operator, respectively. Assuming that the TS operator can access DS information, the optimal results for the integrated transmission and distribution system can be obtained by the following centralized TD-ACOPF model:

$$\min \sum_{i \in N} C_i^p(P_i^e) + \sum_{i \in N_{res}} C_i^{res}(P_i^{res}, Q_i^{res}),$$

subject to $P_i^e + P_i^{res} - P_i(t) - \sum_{j \in N(i)} P_{ij} = 0, \forall i \in N,$

$$Q_i^e + Q_i^{res} - Q_i(t) - \sum_{j \in N(i)} Q_{ij} = 0, \forall i \in N,$$ (1c)

$$P_{ij} = g_{ij}(e_i^2 + f_j^2) - g_{ij}(e_i e_j + f_i f_j) + b_{ij}(e_i e_j - e_i f_j), \ ij \in \Gamma,$$ (1d)

$$Q_{ij} = -b_{ij}(e_i^2 + f_j^2) + b_{ij}(e_i e_j + f_i f_j) + g_{ij}(e_i f_j - e_j f_i), \ ij \in \Gamma,$$ (1e)
$P^g_i \leq P^s_i \leq \bar{P}^s_i, \forall i \in N_g,$ \hspace{1cm} (1f)

$0 \leq P^\text{res}_i \leq P^\text{res}_i(t), \forall i \in N_{\text{res}},$ \hspace{1cm} (1g)

$|Q^\text{res}_i| \leq P^\text{res}_i \tan \theta_i, \forall i \in N_{\text{res}},$ \hspace{1cm} (1h)

$(P^\text{res}_i)^2 + (Q^\text{res}_i)^2 \leq (S^\text{nom})^2, \forall i \in N_{\text{res}},$ \hspace{1cm} (1i)

$V^2_i \leq e^2_i + f^2_i \leq V^2_i, \forall i \in N,$ \hspace{1cm} (1j)

$P^2_{ij} + Q^2_{ij} \leq (\bar{S}_{ij})^2, \forall i,j \in \Gamma,$ \hspace{1cm} (1k)

where $t$ is a continuous index of time. In (1a), the first term represents the generation costs of all conventional generators (e.g., diesels or turbines) and is expressed as the quadratic function active power: $C^g_i(P^g_i) = c_{g2,i}(P^g_i)^2 + c_{g1,i}P^g_i + c_{g0,i}$. For the second term, a quadratic function $C^\text{res}_i(P^\text{res}_i, Q^\text{res}_i) = c^g_i(P^\text{res}_i - P^g_i)^2 + c^q_i(Q^\text{res}_i)^2$ is used to minimize both the amount of active power curtailment and reactive power injected or absorbed of RES. Constraints (1f)–(1e) denote nonlinear AC power flow equations. Constraint (1f) denotes the operational constraints of conventional generators. RESs are modeled as generators with coupled P-Q operational constraints (e.g., [12]); specifically, constraint (1g) limits the active power output by RESs to within the maximum available power and zero; constraint (1h) limits the reactive power output by RESs to within capacitive and inductive minimum power factors; and constraint (1i) is the power generation capacity for RESs. Constraints (1j) and (1k) describe the voltage magnitude limits of the buses and the transmission capacity limits of the lines, respectively.

Constraints (1b), (1c), and (1g) contain three time-varying parameters: the active and reactive power demand $P^d_i(t)$ and $Q^d_i(t)$, and the maximum available active power output of RESs $P^\text{res}_i(t)$. Thus, the aforementioned TD-ACOPF model (1a)–(1k) is essentially a time-varying optimization problem.

### B. Reformulation of the TD-ACOPF Model

For simplicity, the centralized TD-ACOPF (1a)–(1k) can be generalized in the following compact form:

$$\min \ C(x(t), t),$$  \hspace{1cm} (2a)

subject to $g(x(t), t) \leq 0,$  \hspace{1cm} (2b)

$h(x(t), t) \leq 0,$  \hspace{1cm} (2c)

where the variable vector $x(t)$ is specified as $x(t) = [P^g(t); P^\text{res}(t); Q^g(t); Q^\text{res}(t); e(t); f(t); f(t)],$ and the boldface terms represent the vector forms of the related variables; for example, $P^g(t)$ represents the vector consisting of the active power variables of conventional generators; $C(x(t), t)$ denotes the objective in (1a); and (2b) and (2c) denote the equality constraints (1b)–(1e) and inequality constraints (1f)–(1k), respectively.

To enable a decentralized solution, boundary variables $x^B(t) = \{e(t), f(t), e_j(t), f_j(t)/ij \in \Gamma^B\}$ are introduced, where $\Gamma^B$ denotes the set of tie lines connecting TS and DS, and the centralized TD-ACOPF model is reformulated as follows:

$$\min \ C_T(x_T(t), t),$$  \hspace{1cm} (3a)

subject to $g_T(x_T(t), x^B(t), t) = 0,$  \hspace{1cm} (3b)

$h_T \leq h_T(x_T(t), x^B(t), t) \leq h_T,$  \hspace{1cm} (3c)

where $x_T(t)$ represents the vector of the independent variables of the TS model, that is, variables that are only in the TS objective and constraints. Similarly, $x_{D,k}(t)$ is the vector of the independent variables of the $k$th DS model; the TS and DS $k$ models are linked through the vector of boundary variables $x_{B,k}(t)$, and have $x^B(t) = \{x_{B,k}(t), k \in \Omega_D\}$. Boundary variables only contain real and imaginary parts of the voltage variables at the two ends of tie lines, and all boundary variables $x^B(t)$ exist in TS model. The detailed variable partition for the TD-ACOPF model is displayed in Fig. 2.

For exposition, we decomposed the centralized model (3a)–(3e) into a decentralized form, that is, a TS-ACOPF and DS-ACOPFs.

For the TS, we have the following expression:

$$\min \ C_T(x_T(t), t),$$  \hspace{1cm} (4a)

subject to $g_T(x_T(t), x^B(t), t) = 0,$  \hspace{1cm} (4b)

$h_T \leq h_T(x_T(t), x^B(t), t) \leq h_T,$  \hspace{1cm} (4c)

and for the DS $k, k \in \Omega_D$, we have the following expression:

$$\min \ C_{D,k}(x_{D,k}(t), t),$$  \hspace{1cm} (5a)

subject to $g_{D,k}(x_{D,k}(t), x_{B,k}(t), t) = 0,$  \hspace{1cm} (5b)

$h_{D,k} \leq h_{D,k}(x_{D,k}(t), x_{B,k}(t), t) \leq h_{D,k}.$  \hspace{1cm} (5c)
From (4a)–(4c) and (5a)–(5c), because boundary variables exist between the TS and DSSs, models (4a)–(4c) and (5a)–(5c) cannot be solved independently even for a given time instant. Furthermore, the fast-varying parameters \(P^a(t), P^d(t)\) and \(Q^d(t)\) increase the difficulty of solvers for the decentralized TD-ACOPF model.

### III. PROPOSED METHOD

The polynomial-time PDIPM in [34] is widely used for nonlinear optimization problems, especially for dealing with the nonlinear AC power flow constraints in power grids. However, the method is a centralized and iterative-convergence solver and cannot be used in an online application (such as a very fast time scale: i.e., 20 ms). In this study, we recast the PDIPM to develop a primal-dual dynamic system to fast track the saddle points of the nonlinear time-varying TD-ACOPF models (4a)–(4c) and (5a)–(5c) in a decentralized manner.

#### A. Building Primal-Dual Dynamic Systems

To simplify the description, we first unify (4a)–(4c) and (5a)–(5c) as follows:

\[
\begin{align*}
\min & \quad f(x(t), t), \\
\text{subject to} & \quad g(x(t), t) = 0, \\
& \quad h(x(t), t) \leq h_t,
\end{align*}
\]

where \(f(x(t), t)\) denotes the time-varying objective for (4a)–(4c) or (5a)–(5c) without distinguishing upper and lower symbols. Similarly, \(g(x(t), t)\) and \(h(x(t), t)\) denote the equality and inequality in (4a)–(4c) or (5a)–(5c).

After introducing slack variables \(u(t)\) and \(l(t)\) (where \(u(t), l(t) > 0\)), and the barrier factor \(\mu(t)\), we can construct the following barrier function problem to approximate models (6a)–(6c):

\[
\begin{align*}
\min & \quad f(x(t), t) - \mu(t) \left( \sum_{i=1}^{n_h} \ln u_i(t) + \sum_{i=1}^{n_l} \ln l_i(t) \right), \\
\text{subject to} & \quad g(x(t), t) = 0, \\
& \quad h(x(t), t) + u(t) - h_t = 0, h(x(t), t) - l(t) - h_t = 0,
\end{align*}
\]

where \(\mu(t)\) is a barrier parameter, and \(n_h\) is the number of inequality constraints. The barrier parameter is required to be positive and monotonically decrease to zero such that the solutions of barrier function problem (7) approach the solutions of the original problem (6) as \(\mu(t) \to 0\). A convenient choice is \(\gamma e^{-t} (\gamma > 0)\) such that \(\lim_{t \to \infty} \mu(t) = 0\) [31].

Next, we can construct the following Lagrange function for the barrier function problem (7a)–(7c):

\[
\begin{align*}
L(x(t), y(t), w(t), z(t), u(t), l(t), t) &= f(x(t), t) - y(t)^T g(x(t), t) \\
&\quad - w(t)^T \left[ h(x(t), t) + u(t) - h_t \right] - z(t)^T [h(x(t), t) - l(t) - h_t], \\
&\quad - \mu(t) \left[ \sum_{i=1}^{n_h} \ln u_i(t) + \sum_{i=1}^{n_l} \ln l_i(t) \right]
\end{align*}
\]

where \(y(t), w(t)\) and \(z(t)\) are the Lagrange multipliers (dual variables) for equality and inequality constraints, respectively. Thus, the stationary point of (8) must be a KKT point for the barrier function problem (7a)–(7c), which satisfies the first-order optimal conditions of models (7a)–(7c)—that is, the KKT conditions listed as follows:

\[
\begin{align*}
L_x &= \nabla_x f(x(t), t) - \nabla_x^T g(x(t), t) y(t) \\
&\quad - \nabla_x^T h(x(t), t)(w(t) + z(t)) = 0, \\
L_y &= -g(x(t), t) = 0, \\
L_w &= -[h(x(t), t) + u(t) - h_t] = 0, \\
L_z &= -[h(x(t), t) - l(t) - h_t] = 0, \\
L_u &= I_W(t) I_U(t) e + \mu(t) e = 0, \\
L_l &= I_Z(t) I_L(t) e - \mu(t) e = 0,
\end{align*}
\]

where \(I_U(t), I_W(t), I_L(t), I_Z(t)\), and \(I_Z(t)\) are diagonal matrices of \(u(t), w(t), l(t)\) and \(z(t)\).

For simplicity, the aggregate variable vector is defined as \(\lambda(t) := [x(t), y(t), w(t), z(t), u(t), l(t)]\), and we can list the first-order derivative of the Lagrange function (8) at \(\lambda(t)\) as a condensed form by \(\nabla \lambda L(\lambda(t), t)\), which consists of (9a)–(9f). Note that the KKT conditions \(\nabla \lambda L(\lambda(t), t) = 0\) are equivalent in the mathematical sense to the true KKT conditions for the original problem (6) as \(\mu(t) \to 0\).

**Assumption 1:** The fast time-varying parameters—that is, the available active power output of RES and the active and reactive power demands—are continuous and differentiable with respect to \(t \in [0, T]\).

**Remark 1:** Assumption 1 can be satisfied by using Hermite interpolation for the real data.

**Assumption 2:** The ACOPF model has zero duality gaps with their dual semi-definite programming.

**Remark 2:** For most practical ACOPFs—for example, the standard IEEE benchmark systems with 14, 30, 57, 118, and 300 buses—Assumption 2 is satisfied, which implies the existence of saddle points (i.e., optimal solutions, or at least local optimal solutions) [17].

Under Assumption 2, we define the saddle points for the Lagrange (8) as \(\lambda^*(t)\). To obtain the saddle points \(\lambda^*(t)\)—that is, satisfying \(\nabla \lambda L(\lambda^*(t), t) = 0\) for \(t \geq t_0\), we design the following continuous-time dynamic system for the equation \(\nabla \lambda L(\lambda(t), t) = 0\) by using the Newton method:

\[
\dot{\lambda}(t) = -[\nabla \lambda L(\lambda(t), t)]^{-1} \nabla \lambda L(\lambda(t), t).
\]

Note that the trajectory \(\lambda(t)\) generated by (10) can approach a neighborhood around \(\lambda^*(t)\), however, this method cannot converge exactly to \(\lambda^*(t)\) because the problem is changing over time and so is the solution. Since a saddle point must be a
KKT point, the partial derivative of $\nabla_\lambda L(\lambda^*(t), t)$ with respect to $t \in [0, T]$ should also be equal to 0, that is,

$$0 = \nabla_{\lambda t} L(\lambda^*(t), t) \dot{\lambda}^*(t) + \nabla_{\lambda \lambda} L(\lambda^*(t), t), \forall t \geq 0,$$

and solving (11) for $\lambda^*(t)$ yields another dynamic system as follows:

$$\dot{\lambda}^*(t) = -[\nabla_{\lambda t} L(\lambda^*(t), t)]^{-1} \nabla_{\lambda \lambda} L(\lambda^*(t), t), \forall t \geq 0. \quad (12)$$

Note that, in addition to the system time-varying parameters $s(t) = [P^D(t); Q^D(t); P^{\infty}(t)]$ (i.e., the active and reactive power demand, and the maximal available active power output of RESs), the time-varying parameters of $\nabla_{\lambda \lambda} L(\lambda^*(t), t)$ include the barrier parameter $\mu(t)$. Therefore, we have $\nabla_{\lambda \lambda} L(\lambda^*(t), t) = \nabla_{\lambda \lambda} L(\lambda^*(t), t) \delta(t) + \nabla_{\lambda \mu} L(\lambda^*(t), t) \mu(t)$. If the system parameters are time-invariable, that is, $\delta(t) = 0$, the first term on right-hand side of the system (12) is equal to zero; otherwise, the system (12) can be considered a prediction term for correcting the tracking error of the dynamic system (10). Thus, a novel dynamic system can be constructed by combining (10) and (12) as follows:

$$\dot{\lambda}(t) = -[\nabla_{\lambda \lambda} L(\lambda(t), t)]^{-1} (\alpha \nabla_{\lambda \lambda} L(\lambda(t), t) + \beta \nabla_{\lambda \mu} L(\lambda(t), t)), \quad (13)$$

where $\alpha, \beta$ are positive weight coefficients for the Newton term and prediction term, respectively. This refers to a conclusion from [28] that states if the dynamic systems converge to an equilibrium point, such an equilibrium must be a saddle point for OPF problems. Thus, we searched the saddle points of TD-ACOPF problems by developing continuous-time dynamic systems. Moreover, we proved that if the saddle points $\lambda^*(t)$ were found for some $t_0$—that is, satisfying $\nabla_{\lambda \lambda} L(\lambda^*(t_0), t_0) = 0$—system (13) with $\beta = 1$ could be used to track the trajectories of $\lambda^*(t)$ for all $t \geq t_0$. Therefore, the default value of $\beta$ is set as 1. The detailed proof is presented in Part D of this section.

In the dynamic system (13), the first term from (10) can push $\lambda(t)$ toward the optimum (at least local optimal), and the second term from (11) can predict how the optimal solution changes over time by considering time variations of problem parameters. The general principle of building a dynamic system is displayed in Fig. 3, by which solving the nonlinear time-varying model (6a)–(6c) is converted into tracking the stable trajectory of the dynamic system.

We can obtain the dynamic systems of TS-ACOPF (4a)–(4c) and DS-ACOPFs (5a)–(5c), respectively:

$$\dot{\lambda}_T(t) = -[\nabla_{\lambda T} L_T(\lambda_T(t), t)]^{-1}$$

$$(\alpha \nabla_{\lambda T} L_T(\lambda_T(t), t) + \beta \nabla_{\lambda T} L_T(\lambda_T(t), t)), \quad (14a)$$

$$\dot{\lambda}_{D, k}(t) = -[\nabla_{\lambda k D, k} L_{D, k}(\lambda_{D, k}(t), t)]^{-1}$$

$$(\alpha \nabla_{\lambda k D, k} L_{D, k}(\lambda_{D, k}(t), t) + \beta \nabla_{\lambda k D, k} L_{D, k}(\lambda_{D, k}(t), t)), k \in \Omega_D, \quad (14b)$$

According to the superposition principle, a global variable vector can be defined as the union of the variable vector from TS and DSs, that is, $\lambda(t) = [\lambda_T(t) \cup \lambda_{D, k}(t), k \in \Omega_D]$ (the superscript $\sim$ represents the union of variables from TS and DSs), and then we can obtain the following centralized dynamic system:

$$\dot{\lambda}(t) = -[\nabla_{\lambda \lambda} L(\hat{\lambda}(t), t)]^{-1} (\alpha \nabla_{\lambda \lambda} L(\hat{\lambda}(t), t)$$

$$+ \beta \nabla_{\lambda \mu} L(\hat{\lambda}(t), t)), \quad (15)$$

where

$$\nabla_{\lambda \lambda} L(\hat{\lambda}(t), t) = \sum_{k \in \Omega_D} (\hat{E}_k)^T (\nabla_{\lambda k D, k} L_{D, k}(\lambda_{D, k}(t), t)) \hat{E}_k$$

$$+ (\hat{E}_T)^T (\nabla_{\lambda k T} L_T(\lambda_T(t), t)) \hat{E}_T,$$

$$\nabla_{\lambda \lambda} L(\hat{\lambda}(t), t) = \sum_{k \in \Omega_D} (\hat{E}_k)^T (\nabla_{\lambda k D, k} L_{D, k}(\lambda_{D, k}(t), t))$$

$$+ (\hat{E}_T)^T (\nabla_{\lambda k T} L_T(\lambda_T(t), t)),$$

$$\nabla_{\lambda \mu} L(\hat{\lambda}(t), t) = \sum_{k \in \Omega_D} (\hat{E}_k)^T (\nabla_{\lambda k D, k} L_{D, k}(\lambda_{D, k}(t), t))$$

$$+ (\hat{E}_T)^T (\nabla_{\lambda k T} L_T(\lambda_T(t), t)),$$

where $\hat{E}_T$ is a $R_T \times R$ incidence matrix obtained from an $R \times R$ identity matrix by removing rows with indices of $\lambda_T(t)$ in $\hat{\lambda}(t)$, that is, $\lambda_T(t) = \hat{E}_T \hat{\lambda}(t)$, and $R_T$ and $R$ denotes the number of elements in $\lambda_T(t)$ and $\hat{\lambda}(t)$, respectively. Similarly, $\hat{E}_k$ is a $R_k \times R$ incidence matrix obtained from an $R \times R$ identity matrix by removing rows with indices of $\lambda_{D, k}(t)$ in $\hat{\lambda}(t)$, and $R_k$ denotes the number of elements in $\lambda_{D, k}(t)$, that is, $\lambda_{D, k}(t) = \hat{E}_k \hat{\lambda}(k), k \in \Omega_D$.

However, (15) is a centralized solution because its execution requires computing the Hessian inverse $[\nabla_{\lambda \lambda} L(\hat{\lambda}(t), t)]^{-1}$, and implementing such a global computation is sometimes less preferable in practice due to privacy concerns. In the following subsection, we focus on solving the dynamic systems (15) in a decentralized fashion, that is, a protocol such that each DS only requires a few boundary information exchanges with the TS.
If \( \Delta x_k(t_m) \) and \( \Delta y_{D,k}(t_m) \) are obtained by (18), the remaining \( \Delta u_{D,k}(t_m), \Delta l_{D,k}(t_m), \Delta w_{D,k}(t_m), \) and \( \Delta z_{D,k}(t_m) \) can be calculated as follows:

\[
\begin{align*}
\Delta u_{D,k}(t_m) &= -(L_{k,w} + L_{k,wt}) - \nabla^2 x h_{D,k}(x_k(t_m)) \Delta x_k(t_m) \\
\Delta l_{D,k}(t_m) &= (L_{k,z} + L_{k,zt}) + \nabla^2 x h_{D,k}(x_k(t_m)) \Delta x_k(t_m) \\
\Delta w_{D,k}(t_m) &= -I_{k,U}^1(L_{k,u} + L_{k,ut}) - I_{k,U}^0 I_{k,W} \Delta u_{D,k}(t_m) \\
\Delta z_{D,k}(t_m) &= -I_{k,L}^1(L_{k,l} + L_{k,lt}) - I_{k,L}^0 I_{k,Z} \Delta l_{D,k}(t_m)
\end{align*}
\]

where

\[
H = -\left\{ \nabla_{xx} f_{D,k}(x_k(t_m)) - \nabla_{xx} g_{D,k}(x_k(t_m)) y_{D,k}(t) \right\} \\
+ \nabla^2 x h_{D,k}(x_k(t_m))(w_{D,k}(t_m) + z_{D,k}(t)) \\
- \nabla^2 x h_{D,k}(x_k(t_m)) (I_{k,L}^1 L_{k,Z} \\
- I_{k,U}^1 I_{k,W}) \nabla x h_{D,k}(x_k(t_m)) \\
R_{k,x} = (L_{k,x} + L_{k,xt}) \\
+ \nabla^2 x h_{D,k}(x_k(t_m)) \left\{ I_{k,L}^1 (L_{k,l} + L_{k,lt}) \\
+ I_{k,Z}(L_{k,z} + L_{k,zt}) \right\} \\
R_{k,y} = -(L_{k,y} + L_{k,yt})
\]

In (18)–(20), \( H \) is the Hessian matrix; \( I_{k,U}^1, I_{k,W}^1, I_{k,L}^1, I_{k,Z} \) are diagonal matrices of \( u, w, l, \) and \( z; \) \( L_{k,x}, L_{k,y}, L_{k,z}, L_{k,l}, \) and \( L_{k,xt} \) denote the residuals of the KKT condition; and \( L_{k,zt}, L_{k,yt}, L_{k,wt}, L_{k,zt}, L_{k,lt}, \) and \( L_{k,lt} \) denote the partial derivative of the residuals with respect to time \( t \).

From (18)–(19), the second partial derivative of the residuals of the KKT condition with respect to time \( t \) only exists on the left side of the (18). Thus, when the time-varying input parameters update, only the residual vectors \( R_{k,x} \) and \( R_{k,y} \) should be updated.

Similarly, on basis of the PDIPM, the TS’s correction (17a) can be reduced as a sparse linear system with respect to the primal-dual variable increments \( \Delta x_T(t_m) \) (where \( \Delta x_T(t_m) = [(\Delta x_{D,k}(t_m))^T, (\Delta x_{B,k}(t_m))^T]^T \)) and \( \Delta y_T(t_m) \) as follows:

\[
\begin{bmatrix}
\hat{H}_T & \nabla^2 x g_{D,k}(x, t_m) \\
\nabla^2 x g_{B,k}(x, t_m) & 0
\end{bmatrix}
\begin{bmatrix}
\Delta x_k(t_m) \\
\Delta y_{D,k}(t_m)
\end{bmatrix} = \begin{bmatrix}
\hat{R}_T \\
\hat{R}_{T,y}
\end{bmatrix}
\]

(20)

2) Solving Correction Equations in a Decentralized Manner: Here, we derive the distributed computation starting from the centralized sparse linear system. According to the variable partition displayed in Fig. 2, the centralized primal and dual variable vectors \( \hat{x} \) and \( \hat{y} \) (the superscript \( \hat{\cdot} \) represents the union of variables from the TS and DSs) can be decomposed as

\[
\hat{x} = [(x_T)^T, (x_{B,1})^T, (x_{B,1})^T, \ldots, (x_{B,N})^T, (x_{D,N})^T]^T
\]
(22)

\[ \Delta y = \left[ (\Delta y_T)^T, (\Delta y_{D,1})^T, \ldots, (\Delta y_{D,N})^T \right]^T. \]

Accordingly, the centralized primal and dual variable increments vectors \( \Delta \hat{x} \) and \( \Delta \hat{y} \) can be decomposed as

\[
\Delta \hat{x} = \left[ (\Delta x_T)^T, (\Delta x_{B,1})^T, (\Delta x_{D,1})^T, \ldots, (\Delta x_{B,N})^T, (\Delta x_{D,N})^T \right]^T
\]

\[
\Delta \hat{y} = \left[ (\Delta y_T)^T, (\Delta y_{D,1})^T, \ldots, (\Delta y_{D,N})^T \right]^T.
\]

On basis of the PDIPM, a centralized sparse linear system (23) is formed as

\[
\begin{bmatrix}
\hat{H} & \nabla^T \Delta g(x, t) \\
\nabla_x^T \Delta g(x, t) & 0
\end{bmatrix}
\begin{bmatrix}
\Delta \hat{x}(t_m) \\
\Delta \hat{y}(t_m)
\end{bmatrix} =
\begin{bmatrix}
\hat{R}_x \\
\hat{R}_y
\end{bmatrix},
\]

where

\[
\hat{H} =
\begin{bmatrix}
H_T & H_{TB,1} & \cdots & H_{TB,N} \\
H_{TB,1}^T & H_{B,1} & \cdots & 0 \\
& 0 & \ddots & \vdots \\
& \vdots & \ddots & \ddots \\
H_{TB,N}^T & 0 & \cdots & H_{B,N} \quad H_{BD,N} \\
0 & 0 & \cdots & H_{BD,N}
\end{bmatrix}
\]

\[
\begin{bmatrix}
\nabla^T \Delta g(x, t) \\
\nabla_x^T \Delta g(x, t)
\end{bmatrix} =
\begin{bmatrix}
\nabla^T g_T, \nabla^T g_{D,1} \ldots \nabla^T g_{D,N} \\
\nabla^T g_{D,1} \nabla^T g_{D,2} \ldots \nabla^T g_{D,N}
\end{bmatrix}.
\]

\[
\begin{bmatrix}
\hat{R}_x \\
\hat{R}_y
\end{bmatrix} =
\begin{bmatrix}
R_{T,x} & R_{B1,x} & \cdots & R_{BD,N,x} \quad R_{DN,x} \\
R_{T,y} & R_{B1,y} & \cdots & R_{BD,N,y}
\end{bmatrix}
\]

Therefore, we can obtain the following (24) from (23) by singling out rows with indices of \( \Delta x_{D,k}(t_m) \), \( \Delta y_{D,k}(t_m) \) and \( \Delta x_{B,k}(t_m) \).

\[
\begin{bmatrix}
H_{D,k} & \nabla^T_{x_{D,k}} g_{D,k} \\
\nabla_{x_{D,k}} g_{D,k}
\end{bmatrix}
\begin{bmatrix}
\Delta x_{D,k}(t_m) \\
\Delta y_{D,k}(t_m)
\end{bmatrix} =
\begin{bmatrix}
R_{D,k,x} \\
R_{D,k,y}
\end{bmatrix} -
\begin{bmatrix}
H_{BD,k}^T & \nabla^T_{x_{B,k}} g_{D,k} \\
\nabla_{x_{B,k}} g_{D,k}
\end{bmatrix}
\begin{bmatrix}
\Delta x_{B,k}(t_m)
\end{bmatrix},
\]

Solving (24), \( \Delta x_{D,k}(t_m) \) and \( \Delta y_{D,k}(t_m) \) can be represented parametrically as a linear function of \( \Delta x_{B,k}(t_m) \) as follows:

\[
\begin{bmatrix}
\Delta x_{D,k}(t_m) \\
\Delta y_{D,k}(t_m)
\end{bmatrix} =
\begin{bmatrix}
H_{D,k} & \nabla^T_{x_{D,k}} g_{D,k} \\
\nabla_{x_{D,k}} g_{D,k}
\end{bmatrix}^{-1}
\begin{bmatrix}
R_{D,k,x} \\
R_{D,k,y}
\end{bmatrix} -
\begin{bmatrix}
H_{BD,k} & \nabla^T_{x_{D,k}} g_{D,k} \\
\nabla_{x_{B,k}} g_{D,k}
\end{bmatrix}
\begin{bmatrix}
\Delta x_{B,k}(t_m)
\end{bmatrix}.
\]

The remaining equation in (23) with indices of \( \Delta x_T(t_m), \Delta x_{B,1}(t_m), \ldots, \Delta x_{B,N}(t_m) \) and \( \Delta y_T(t_m) \) can be written in the following form (26):

\[
\begin{bmatrix}
H_T & H_{TB,1} & \cdots & H_{TB,N} & \nabla^T_{x_T} g_T \\
H_{TB,1}^T & H_{B,1} & \cdots & 0 & \nabla^T_{x_{B,1}} g_T \\
& \vdots & \ddots & \ddots & \vdots \\
& \vdots & \ddots & \ddots & \vdots \\
H_{TB,N}^T & 0 & \cdots & H_{B,N} & \nabla^T_{x_{B,N}} g_T \\
\nabla_{x_T} g_T & \nabla_{x_{B,1}} g_T & \cdots & \nabla_{x_{B,N}} g_T & 0
\end{bmatrix}
\begin{bmatrix}
\Delta x_T(t_m) \\
\Delta x_{B,1}(t_m) \\
\vdots \\
\Delta x_{B,N}(t_m) \\
\Delta y_T(t_m)
\end{bmatrix} =
\begin{bmatrix}
R_{T,x} \\
R_{B1,x} \\
\vdots \\
R_{B1,N} \\
R_{T,y}
\end{bmatrix}
\]

\[
\begin{bmatrix}
\Delta x_{D,1} \\
\Delta y_{D,1}
\end{bmatrix} \times
\begin{bmatrix}
\Delta x_{D,N} \\
\Delta y_{D,N}
\end{bmatrix}.
\]

Substituting (25) into (26), (26) can be simplified as the following form:

\[
\begin{bmatrix}
H_T & H_{TB,1} & \cdots & H_{TB,N} & \nabla^T_{x_T} g_T \\
H_{TB,1}^T & H_{B,1} - J_{1,1} & \cdots & 0 & \nabla^T_{x_{B,1}} g_T \\
& \vdots & \ddots & \ddots & \vdots \\
& \vdots & \ddots & \ddots & \vdots \\
H_{TB,N}^T & 0 & \cdots & H_{B,N} - J_{N,2} & \nabla^T_{x_{B,N}} g_T \\
\nabla_{x_T} g_T & \nabla_{x_{B,1}} g_T & \cdots & \nabla_{x_{B,N}} g_T & 0
\end{bmatrix}
\begin{bmatrix}
\Delta x_T(t_m) \\
\Delta x_{B,1}(t_m) \\
\vdots \\
\Delta x_{B,N}(t_m) \\
\Delta y_T(t_m)
\end{bmatrix} =
\begin{bmatrix}
R_{T,x} \\
R_{B1,x} - J_{1,1} \\
\vdots \\
R_{B1,N} - J_{N,1} \\
R_{T,y}
\end{bmatrix}.
\]

where

\[
J_{k,1} = [H_{BD,k} \ \nabla^T_{x_{B,k}} g_{D,k}] \\
J_{k,2} = [H_{BD,k} \ \nabla^T_{x_{B,k}} g_{D,k}]
\]

According to the above derivation, the distributed computation for the centralized sparse linear system (23) contains three key steps:
Step 1: For DS $k, k \in \Omega_D$, compute the coefficient vector and matrix $J_{k,1}$ and $J_{k,2}$, respectively, which are taken as equivalent parameters and are sent to TS via the upward message passing channel.

Step 2: For TS, compute the variables’ increments $\Delta x_T(t_m), \Delta y_T(t_m)$, and $\Delta x_{B,k}(t_m), k \in \Omega_B$ by solving the integrated correction (27), and then sent $\Delta x_{B,k}(t_m)$ to DS $k$ via the down message passing channel.

Step 3: For DS $k, k \in \Omega_D$, compute the variables’ increments $\Delta x_{D,k}(t_m), \Delta y_{D,k}(t_m)$ by (24) given $\Delta x_{B,k}(t_m)$.

Since the above derivation is purely algebraic, the primal-dual variables’ increments obtained by the distributed computation steps (1)–(3) also satisfy the centralized sparse linear system (23).

3) Communication:

i) Communication topology: According to the variable partition structure in Fig. 2, all boundary variables exist in the TS model. Thus, a radial communication topology diagram can be designed to achieve a decentralized implementation for solving the sparse linear systems of the TS and DSSs. Fig. 5 displays the interaction between the TS and DSSs. When the TS receives the coefficient matrix and vector from DSSs, the boundary variables’ increments can be obtained by (27) and transmitted to the DSSs. Next, for each DS, after receiving the boundary variables’ increments from the TS, (24) can be used to compute its independent variables’ increments.

ii) Date exchange: For the upward message passing, considering DS $k$ for example, only the coefficient matrix and vector $J_{k,2}$ and $J_{k,1}$ should be sent to TS through the upward communication channels. According to the variable partition in Fig. 2, boundary variables only contain the real and imaginary parts of the voltage variables at the two ends of tie lines. Therefore, $J_{k,2}$ and $J_{k,1}$ are a $4 \times 4$ matrix and a $4 \times 1$ vector, respectively. For the down message passing, TS sends the boundary variables $\Delta x_{B,k}(t_m), \forall k \in \Omega_D$ to DS $k$ through the communication channels, which is a $4 \times 1$ vector. Practically, each DS is connected to the TS by just one tie line. Therefore, for each information channel, a $4 \times 4$ matrix and a $4 \times 1$ vector are the maximum amount of communication, which does not cause a heavy communication burden in real applications regardless of the large scale the couple transmission-distribution grid. In addition, for each sampling, only one exchange is conducted between the TS and DSSs. Thus, such an implementation can considerably decrease the communication burden.

4) Variable Updating: In the PDPM, the scalar primal-dual step sizes $\{\alpha_p^T, \alpha_d^T\}, \{\alpha_p^k, \alpha_d^k\}(k \in \Omega_D)$ are selected to preserve the feasible condition on slack variables $u$ and $I$, and Lagrange multipliers $w$ and $z$. Therefore, the variables of the dynamic system (13) are updated using the following variables:

\[
\begin{align*}
x_T(t_{m+1}) &= x_T(t_m) + \tau_p^T \Delta x_T(t_m), \\
x_B(t_{m+1}) &= x_B(t_m) + \tau_B^T \Delta x_B(t_m), \\
y_T(t_{m+1}) &= y_T(t_m) + \tau_B^T \Delta y_T(t_m), \\
l_T(t_{m+1}) &= l_T(t_m) + \tau_B^T \Delta l_T(t_m), \\
z_T(t_{m+1}) &= z_T(t_m) + \tau_B^T \Delta z_T(t_m), \\
u_T(t_{m+1}) &= u_T(t_m) + \tau_B^T \Delta u_T(t_m), \\
w_T(t_{m+1}) &= w_T(t_m) + \tau_B^T \Delta w_T(t_m), \\
\end{align*}
\]

and

\[
\begin{align*}
x_{D,k}(t_{m+1}) &= x_{D,k}(t_m) + \tau_p^k \Delta x_{D,k}(t_m), \quad \forall k \in \Omega_D, \\
y_{D,k}(t_{m+1}) &= y_{D,k}(t_m) + \tau_d^k \Delta y_{D,k}(t_m), \quad \forall k \in \Omega_D, \\
l_{D,k}(t_{m+1}) &= l_{D,k}(t_m) + \tau_d^k \Delta l_{D,k}(t_m), \quad \forall k \in \Omega_D, \\
z_{D,k}(t_{m+1}) &= z_{D,k}(t_m) + \tau_d^k \Delta z_{D,k}(t_m), \quad \forall k \in \Omega_D, \\
u_{D,k}(t_{m+1}) &= u_{D,k}(t_m) + \tau_p^k \Delta u_{D,k}(t_m), \quad \forall k \in \Omega_D, \\
w_{D,k}(t_{m+1}) &= w_{D,k}(t_m) + \tau_d^k \Delta w_{D,k}(t_m), \quad \forall k \in \Omega_D, \\
\end{align*}
\]

C. Online Tracking Framework for the Proposed Algorithm

To achieve online applications, the output power setting points of the RESs should be given quickly after each sampling. In this case, only one or a few iterations are affordable. Fig. 6 depicts the sequence diagram of the online implementation for the proposed algorithm. Suppose that the proposed algorithm starts at an initial time $t_0$ and achieves online tracking at time $t_1$; thus, the update results for all time $t \geq t_1$ can be applied directly into the command level. To speed up the tracking process, the proposed algorithm is allowed to iterate from initial time $t_0$
can be expressed satisfying (13) ≥ \(\beta\) with \(\lambda(40)\)

\[
\dot{\lambda}(t) = -\alpha \nabla \lambda L(\lambda(t), t) + \beta \nabla \lambda L(\lambda(t), t)
\]

After substituting \(\dot{\lambda}(t)\) from (13) into (30), (30) can further be converted as follows:

\[
\nabla \lambda L(\lambda(t), t) = -\alpha \nabla \lambda L(\lambda(t), t) + (1 - \beta) \nabla \lambda L(\lambda(t), t)
\]

When the default value of \(\beta\) is set as 1, (31) can be rewritten as follows:

\[
\nabla \lambda L(\lambda(t), t) = -\alpha \nabla \lambda L(\lambda(t), t)
\]

which yields the following solution:

\[
\nabla \lambda L(\lambda(t), t) = e^{-\alpha t} \nabla \lambda L(\lambda(0), 0)
\]

where \(\lambda(0)\) is the initial point. The following inequality holds:

\[
\|\nabla \lambda L(\lambda(t), t)\|_2 \leq e^{-\alpha t} \|\nabla \lambda L(\lambda(0), 0)\|_2
\]

Therefore, \(\lim_{t \to \infty} \|\nabla \lambda L(\lambda(t), t)\| = 0\).

Defining the tracking error \(e(t) = \lambda(t) - \lambda^*(t)\) and building the following continuously differentiable function [36], we have:

\[
V(e(t), t) = \|\nabla \lambda L(e(t) + \lambda^*(t), t)\|^2 / 2 = \|\nabla \lambda L(\lambda(t), t)\|^2 / 2
\]

and also have:

\[
\begin{cases}
V(0, t) = \|\nabla \lambda L(\lambda^*(t), t)\|^2 / 2 = 0 \\
V(e(t), t) = \|\nabla \lambda L(\lambda(t), t)\|^2 / 2 > 0, e(t) \neq 0
\end{cases}
\]

The derivative of \(V(e(t), t)\) with respect to \(t\) can be expressed as follows:

\[
\dot{V}(e(t), t) = \nabla^2 \lambda L(\lambda(t), t) \dot{\lambda}(t) + \nabla \lambda L(\lambda(t), t)
\]

After substituting \(\dot{\lambda}(t)\) from (13) with \(\beta = 1\) into (37), (37) can further be converted as follows:

\[
\dot{V}(e(t), t) = -\alpha \|\nabla \lambda L(\lambda(t), t)\|^2
\]

Therefore, we have:

\[
\begin{cases}
\dot{V}(0, t) = -\alpha \|\nabla \lambda L(\lambda^*(t), t)\|^2 = 0 \\
\dot{V}(e(t), t) = -\alpha \|\nabla \lambda L(\lambda(t), t)\|^2 < 0, e(t) \neq 0
\end{cases}
\]

According to the Lyapunov theorem (see theorem 4.1 in [37]), the continuously differentiable function \(V(e(t), t)\) satisfying (36) and (39) is called a Lyapunov function, and thus the dynamic system (13) with \(\beta = 1\) is asymptotically stable. Therefore, we can conclude

\[
\lim_{t \to \infty} \|e(t)\| \leq \lim_{t \to \infty} \|\lambda(t) - \lambda^*(t)\| = 0
\]

**Remark 3 (Role of \(\beta\)):** When \(\beta = 1\), if we can find the solution \(\lambda^*(0)\) satisfying \(\nabla \lambda L(\lambda^*(0), 0) = 0\), according to (34),

D. Convergence Analysis

An overview of the proposed online distributed tracking method is displayed in Fig. 8, which consists of the following two parts: building the continuous-time dynamic system and solving dynamic systems based on forward Euler discretization. The effectiveness of the distributed computation mode for solving dynamic systems has been proved in the appendix. Next, we discuss the convergence of continuous-time dynamic systems for tracking the saddle points of TD-ACOPF models.

The time derivative of the gradient \(\nabla \lambda L(\lambda(t), t)\) at any \((\lambda(t), t)\) can be expressed as follows:

\[
\dot{\nabla} \lambda L(\lambda(t), t) = \nabla \lambda L(\lambda(t), t) \dot{\lambda}(t) + \nabla \lambda L(\lambda(t), t)
\]

Fig. 6. Sequence diagram of the proposed online distributed algorithm for online applications.

Fig. 7. Sequence of the hierarchical-strategy-based slow and fast timescales alternately.
we can deduce $\nabla_\lambda L(\lambda(t), t) = 0$ for all $t \geq 0$; thus, the dynamic system (13) with $\beta = 1$ can be used to search the trajectories of $\lambda^*(t)$ for all $t \geq 0$.

**Remark 4 (Role of $\alpha$):** Inequality (34) reveals that increasing $\alpha$ can speed convergence for the continuous-time dynamics (13). However, when we use the forward Euler method with a constant sampling period $\tau$ to discretize the continuous-time dynamics (15), the coefficient $\alpha$ affects the stability and convergence of discretized dynamics; specifically, the effective step size $\alpha \tau$ has an upper bound for stability and convergence. More details can be found in [38]. Furthermore, when $\alpha = 0$, (32) can be rewritten as $\nabla_\lambda L(\lambda(t), t) = 0$, yielding the solution $\nabla_\lambda L(\lambda(t), t) = \nabla_\lambda L(\lambda(0), 0)$. If the initial point $\lambda(0)$ is not optimal—that is, $\nabla_\lambda L(\lambda(0), 0) \neq 0$—the dynamics (15) with $\alpha = 0$ will not track the optimal trajectories of $\lambda^*(t)$ at any time. Therefore, parameter $\alpha$ must be positive while has an upper bound, and its specific value analysis can be seen in the case study.

**IV. NUMERICAL RESULTS**

The proposed method is tested on an integrated system (Fig. 9). The IEEE 9-bus TS connects three DSs at buses 5, 7, and 9 of the TS. All the DSs are 33-bus systems, equipped with nine RESs, that is, four WT systems and five PV systems with the ratings of 200 kVA (Fig. 10).

To simulate a dynamic scenario, we used the real data of a particular day for PV/WT system generation and load profiles, available in [32]. The raw data are on a 5-min basis, and we interpolate the data to change the time granularity to 1 s. Furthermore, we used Hermite interpolation to obtain continuously time-varying curves. Fig. 11 depicts the available active power of a PV system at bus 17 and a WT system at bus 25 as well as the total loads of the integrated system. We also diversify the load and PV/WT generation profile for each bus by small random additive noises. This study focused on online tracking for ultra-short-term (e.g., 20 ms) optimization, and because such short-term RES and load forecasts are typically accurate, their prediction errors are assumed to be negligible [39]. Coefficients $c_\rho^p = 3$ and $c^q_i = 1$ were set for both the PV and WT owners’ objectives according to [32].

The simulation was run from 12:00:00 to 12:10:00 (30000 time slots in total; each time slot is 20 ms) on a DELL Precision
3640 workstation with 64 GB RAM and 3.7 GHz Intel (R) Core (TM) i9-10900K CPU. All the programs were developed in MATLAB 2018b. The traditional TD-ACOPF updates took place every 60 s starting from 12:00 am. Between each traditional ACOPF update, the online TD-ACOPF implemented every 20 ms. Every 60 s, the barrier parameter was set as \( \gamma_{e} = \frac{(l(t_{s})^T z(t_{s}) - u(t_{s})^T w(t_{s}))}{20 n_{h}} \), and \( t_{s} \) is the starting time for implementing online ACOPF.

To analyze the performance of the proposed approach, three TD-ACOPF methods were conducted on the IEEE 9-bus TS connected with three DSs:

i) **Centralized method**: To verify the accuracy of the proposed decentralized optimization, the IPOPT solver, provided by the MATLAB-based toolbox “MATPOWER” [40], was used to solve the centralized TD-ACOPF model (1) with a sampling period of 20 ms. Here, IPOPT is an iterative-convergence centralized algorithm, and it is sometimes less preferable in practice due to privacy concerns or the limitation of computational capabilities for TD-ACOPF problems. The centralized results were only used to verify the accuracy of the proposed decentralized method.

ii) **Independent method**: To verify the validity of the proposed coordination optimization, an independent optimization method was used for the separate transmission and distribution ACOPF based on a predefined boundary condition. Specifically, TS performed its ACOPF by assuming the distribution systems as load models with constant power, and DS performed its ACOPF by assuming the boundary bus (substation bus) as the given value, which is typically regulated by the TS. A satisfactory predefined value of the boundary bus voltage can be obtained according to the base case flow [12]. For the IEEE 9-bus TS connected with three DSs, the voltage amplitudes of three boundary buses were set to 1.0813, 1.0881, 1.0687 p.u., respectively. Both TS and DS optimize their own ACOPF also by IPOPT solver.

iii) **Decentralized method**: The proposed online decentralized approach based on the continuous-time dynamic system was used to solve decentralized TD-ACOPF models (4) and (5) to track the centralized optimal trajectories. The default values of \( \beta \) and \( \alpha \) were set as 1 and 0.1, respectively.

### A. Coordination Effectiveness Analysis

The trajectories of total objective values and voltage profiles of boundary buses obtained by methods (i)–(iii) are displayed in Figs. 12 and 13. The trajectories of the decentralized optimization are very close to those of centralized optimization after approximately 1.5 s. This result implies that the proposed decentralized approach can achieve exact tracking for the centralized solution within a short time. Compared with the sampling period
As displayed in Fig. 12, the objective values of the decentralized optimization are always smaller than that of independent optimization, and the mean objective reduction ($195.61$) accounts for $4.0\%$ of the mean objective ($4858.41$). Thus, the proposed decentralized optimization method can provide superior economic performance for the coupled transmission and distribution systems.

RES penetration can be increased three times. Fig. 14 presents a comparison of the voltage profiles of the distribution networks at $t = 12:02:00$. Most of the bus voltages obtained by the independent optimization method reach the upper limit (1.1 p.u.), which likely leads to overvoltage in the presence of unexpected system disturbance. However, this risk can be mitigated by the proposed decentralized optimization because the narrow margin of voltage improved considerably, as displayed in Fig. 14.

Thus, the proposed decentralized optimization method is highly suitable for handling the voltage rise issue of the distribution system.

**B. Prediction Effectiveness Analysis**

To analyze the effect of the prediction term, we used the proposed decentralized approach with and without the prediction term to solve the decentralized TD-ACOPF model, denoted by “decentralized with $\beta = 1$” and “decentralized with $\beta = 0$”, and then compared them with centralized results. As shown in Fig. 15, the proposed decentralized approach with $\beta = 1$ can achieve performance comparable to that of the centralized solver and sustain stable tracking performance. In contrast, without the prediction term, the objective trajectories of the decentralized optimization deviate considerably from the centralized trajectories.

Fig. 16 displays the relative deviations of total objective values between the results obtained using the centralized method and those obtained using the decentralized methods with and without the prediction term. For decentralized optimization with $\beta = 1$, the relative deviation was less than $0.014\%$ for the whole time horizon, and the mean relative deviation was $0.002\%$. This result implied that using the prediction term in the proposed decentralized approach can considerably decrease the tracking error.

Fig. 15 reveals that the objective values obtained by the decentralized method with $\beta = 0$ are sometimes lower than that with $\beta = 1$. However, this result does not indicate that the results without using the prediction term are more economical. Because the problem is changing over time and so is the solution, without such a prediction term, the obtained results would deviate from the centralized results, and could even be infeasible. Fig. 17 depicts the max mismatch in the calculation of power flow equations, denoted by “Max(dP, dQ)”, which can reveal how unfeasible they are for power flow equations:

$$\text{Max}(dP, dQ) = \|g(x)\|_\infty,$$

(41)

Meanwhile, Fig. 18 depicts the max violations of inequality constraints, represented by the maximum residuals of the KKT
LU et al.: ONLINE DECENTRALIZED TRACKING FOR NONLINEAR TIME-VARYING OPTIMAL POWER FLOW OF COUPLED TD GRIDS

Fig. 17. Max(dP,dQ) obtained by the proposed decentralized approach with and without the prediction term.

Fig. 18. Max(L_w, L_z) obtained by the proposed decentralized approach with and without the prediction term.

condition associated with inequality constraints as follows:

\[
\text{Max}(L_w, L_z) = \max\left(\|h(x, t) + u(t) - h\|_{\infty}, \|h(x, t) - l(t) - h\|_{\infty}\right),
\]

(42)

Figs. 17 and 18 reveal that both the Max(dP,dQ) and Max(L_w,L_z) obtained by the decentralized method with \( \beta = 1 \) are always considerably lower than that with \( \beta = 0 \). As displayed in Fig. 17, most of the Max(dP,dQ) values obtained by the decentralized method with \( \beta = 0 \) exceeded \( 10^{-3} \) (p.u.); that is, 0.1 MW. Thus, most solutions are infeasible. In contrast, most of the Max(dP,dQ) values with \( \beta = 1 \) were less than \( 10^{-4} \) (p.u.), whereas the mean Max(dP,dQ) value was \( 4.4502 \times 10^{-5} \), satisfying the feasibility.

C. Tracking Performance Analysis

To analyze the effect of parameter \( \alpha \) on the stability of the proposed decentralized method, the prediction term coefficient was fixed to \( \beta = 1 \), and the simulation was conducted on six cases for \( \alpha \in \{0.01, 0.05, 0.1, 0.5, 1, 5\} \). To reveal the tracking performance, we only depict the trajectories for 60 s from 12:00:00 to 12:01:00, including objective values, their relative deviations from the centralized results, and Max(dP,dQ) in Figs. 19–21. As displayed in Fig. 19, increasing \( \alpha \) causes the objective trajectories of the decentralized optimization to deviate earlier from the centralized trajectories. Furthermore,
Fig. 22. Relative errors of objective values obtained by the proposed decentralized approach with various sampling periods ($\beta = 1, \alpha = 0.1$).

Fig. 23. Max(dP,dQ) values obtained by the proposed decentralized approach with various sampling periods ($\beta = 1, \alpha = 0.1$).

Figs. 20 and 21 reveal that the higher values of $\alpha$ could speed up the tracking process. However, too large values could lead to poor performance and even infeasibility (e.g., $\alpha \geq 1$). Thus, to sustain stable tracking performance, we selected $\alpha = 0.1$ as a compromise.

To analyze the performance of the proposed decentralized approach for various sampling periods, four cases were simulated for $\tau \in \{0.01, 0.02, 0.05, 0.1, 0.2, 0.5\}$ (s) with fixing $\beta = 1$ and $\alpha = 0.1$. As displayed in Figs. 22 and 23, the smaller the sampling period is, the smaller the tracking error is. Smaller periods denote smaller changes in adjacent periods, and the prediction is accurate. However, a too small sampling period (e.g., $\tau = 0.01$ s) requires high communication and computation requirements. By contrast, a too large sampling period (e.g., $\tau = 0.5$ s) results in poor tracking performance and even infeasibility, as shown in Fig. 23. Thus, we selected $\tau = 0.02$ s as a compromise.

D. Large-Scale System Tests

To demonstrate the tracking performance of the proposed decentralized method on large-scale couple transmission and distribution systems, we tested the proposed approach on an IEEE 118-bus TS connected with thirteen 69-bus DSs at buses 2, 7, 20, 28, 43, 44, 48, 50, 51, 52, 83, 97, and 20 of the TS [15], resulting in 1015 buses and fourteen control centers in total. We added five RESs—specifically, three WT systems and two PV systems with ratings of 500 kVA—at buses 10, 20, 30, 40, and 50 of each 69-bus DS. We still used the load and PV/WT generation profiles in Fig. 11 and normalized each profile to have a maximum value of 1.

Figs. 24 and 25 depict the objective values obtained by the proposed method and their relative deviations from the centralized results. The default values of $\alpha$ and $\tau$ were set as 0.1 and 0.02 s, respectively. The results indicated the proposed decentralized method with prediction term can achieve almost the same objective values as the centralized solver, and the maximum and mean relative errors were 0.038% and 0.002%, respectively. As displayed in the subgraph of Fig. 24, without the prediction term, the proposed decentralized method resulted in deviated trajectories from the centralized results. This result implied that using the prediction term in the proposed decentralized approach can improve the tracking accuracy considerably.

We analyzed the feasibility of the decentralized results by depicting Max(dP,dQ) and Max($L_w$, $L_z$) in Fig. 26. Most of the Max(dP,dQ) and Max($L_w$, $L_z$) values were less than $10^{-4}$ (p.u.) and $10^{-6}$ (p.u.), respectively; thus, almost all the equality and inequality constraints were satisfied, which reveals the feasibility of the decentralized results.
Fig. 26. Max(dP,dQ) and Max(L_p,L_z) obtained by the proposed decentralized approach with the prediction term ($\alpha = 0.1$ and $\tau = 0.02$ s).

| IEEE 9-bus TS with three 33-bus DSs | IEEE 118-bus TS with thirteen 69-bus DSs |
|-------------------------------------|------------------------------------------|
| Iteration number | Time (s) | Errors (%) | Iteration number | Time (s) | Errors (%) |
|---------|---------|------------|---------|---------|------------|
| Proposed | 1 | 1 | 0.006 | 0.009 | 0.002 | 0.004 | 1 | 1 | 0.014 | 0.016 | 0.002 | 0.005 |
| F-DIPM | 28 | 30 | 0.381 | 0.453 | 0.001 | 0.001 | 38 | 40 | 3.026 | 3.425 | 0.001 | 0.002 |
| DCC | 5 | 6 | 1.028 | 1.139 | 0.001 | 0.001 | 16 | 17 | 10.319 | 11.902 | 0.001 | 0.001 |
| HGD | 35 | 48 | 1.98 | 2.35 | 0.002 | 0.003 | HGD/OC/DADM | APP did not converge within 500 iterations |
| OCD | 59 | 74 | 3.01 | 5.80 | 0.009 | 0.014 | OCD/OC/DADM | APP did not converge within 500 iterations |
| ADMM | 102 | 134 | 68.00 | 90.10 | 0.010 | 0.023 | ADMM | APP did not converge within 500 iterations |
| APP | 97 | 128 | 61.20 | 88.00 | 0.014 | 0.019 | APP did not converge within 500 iterations |

Table I: Comparison of Various Algorithms for Decentralized Coordinated Optimization

E. Online Application Analysis

To analyze the online application capacity of the proposed decentralized method, the proposed decentralized approach with six conventional iterative-convergence algorithms, namely, F-DIPM [13], DCC [15], HGD [12], OCD [9], ADMM [7], and APP [6], can be compared. The comparison terms include the mean and maximum values of the computation time, iteration number, and the relative deviations (denoted by “errors”) of the objective value to the centralized value, in the range of 12:01:00 to 12:02:00 (total 60 s, 3000 time-slots) in Table I.

As presented in Table I, the proposed algorithm only conducts one iteration and exhibits fast-tracking performance. The maximum CPU times of the proposed method were 0.009 s and 0.016 s for the IEEE 9-bus TS with three 33-bus DSs, and the IEEE 118-bus TS with thirteen 69-bus DSs, respectively. The relative deviation of objective values of the proposed method was acceptable although higher than that of F-DIPM and DCC.

By contrast, the conventional iterative-convergence algorithm requires an excessive number of iterations, which could require additional information communication with a higher probability of failure and is highly vulnerable to attack.

For the IEEE 118-bus TS with thirteen 69-bus DSs, the HGD and OCD based on the first-order optimality condition decomposition, iterate diverged because the convergence criterion $\rho^* < 1$ ($\rho^*$ denotes the spectral radius, see theorem 1 [41]) does not hold at the KKT point. In the augmented Lagrangian decomposition (ALD) methods, APP and ADMM, the multiplier updating process requires tuning parameters by trial and error, potentially affecting convergence. As presented in Table I, the best parameter values of APP and ADMM algorithms were selected to converge for IEEE 9-bus TS with three 33-bus DSs. We could not obtain the converged result for the IEEE 118-bus TS with thirteen 69-bus DSs within 500 iterations.

V. CONCLUSION AND FURTHER DIRECTIONS

An online decentralized tracking algorithm was proposed for the coordinated TD-ACOPF in coupled transmission and distribution grids. The proposed method is designed from the perspective of control to construct a dynamic system to capture the information of the objectives and constraints that evolve over time. In this method, continuous-time dynamic systems are used to search the saddle points of TD-ACOPF problems. A decentralized implementation was developed based on only a few information exchanges with respect to boundary variables. The results of the case studies verified the coordination effectiveness of the proposed algorithm compared with the independent optimization method. The results also revealed that the performance of the proposed method is accurate, capable of fast-tracking, and exhibits online application capacity that is greater than conventional iterative-convergence–based decentralized algorithms.

Despite the proposed model, designing the online TD-ACOPF algorithm remains challenging. In the simulation, the sampling period is set as 20 ms. However, in practice, 20 ms could be too short for handling the dynamics of power devices with large inertia. A possible method is to add a new tracking objective that can minimize the deviation from the slower time-scale control signals, which will be investigated in the future.

Additionally, for a controllable device, there might be some discrepancy between the control command it receives and its actual implementation. For example, the settling time of the device is larger than the interval between real-time updates. Such a discrepancy might lead to constraint violation. A possible method is to partition the variables into controllable variables and dependent variables, update the controllable variables by the actual power injections, and use the power network to automatically solve the power flow equations for the dependent variables (voltages), like [22]. If there exist some voltage or current magnitude constraint violations, we can add a penalty function to the objective to prevent the constraint violations. Inevitably, real-time measurements are required, and how to minimize the measurement requirements will be investigated in the future.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
