Transient Stability Enhancement via a Scalable RL Method with VSG Parameter Tuning

Xiaoge Huang, Ziang Zhang
Department of Electrical and Computer Engineering
Binghamton University - SUNY
Binghamton, U.S.A.
xhuang98@binghamton.edu, ziang.zhang@binghamton.edu

Shufan Wang, Jian Li
AMS department, CS department
Stony Brook University
Stony Brook, USA
Shufan.wang@stonybrook.edu, Jian.li.3@stonybrook.edu

***Abstract*—This paper presents a reinforcement learning (RL)-driven strategy to improve the transient stability of power systems via tuning parameters of multiple virtual synchronous generators (VSGs). We proposed a scalable method to support RL training convergence probability and speed, even when a large number of contingencies are considered. The proposed scalable RL framework first decomposes the large number of contingencies into multiple groups and then conducts parallel training for each group, decreasing the state space and complexity of each training. Additionally, we propose a contingency grouping algorithm to streamline the RL action space and facilitate the training. The proposed method is validated across various standard test systems.**

Keywords—virtual synchronous generator, transient stability, reinforcement learning, contingency-grouping, optimization

I. INTRODUCTION

The supply resources in the modern power grid are transitioning from centralized thermal synchronous generators to distributed renewable inverter-based resources (IBRs). Against this backdrop, integrating IBRs into the electrical grid has garnered significant interest. The latest studies highlight that incorporating inverters can influence the transient response of the power systems [1]. However, enhancing or maintaining systems' synchronism with mixed IBRs and synchronous generators remains an unresolved issue. This work focuses on using the VSGs, a representative IBR, to offer supplementary aid for the transient stability of power systems with mixed IBRs and generators.

In general, control methods to enhance transient stability can be applied at either the power plant level or the system level. The primary challenge with plant-level control lies in developing an algorithm capable of detecting faults, determining control actions with restricted observations, and implementing actions in real-time (within a second). Conversely, control at the system operator level can perform offline simulations with estimated contingency scenarios. Yet, vast controlled parameters and significant uncertainties on the potential contingencies and pre-fault power flow need to be considered. This work focuses on system-level control.

Many studies have focused on formulating the transient stability enhancement as a preventive control [2] or a model predictive control problem [3]. While these methods yield near-optimal solutions for control decisions, they typically require linearization or simplification of the complex system model to ensure solvability in optimization. Moreover, applying these methods to large-scale power systems introduces great computational complexities in the solutions.

Transient stability enhancement can also be achieved by calculating the transient stability margin based on Lyapunov theory [4], [5]. Then, control strategies or stability constraints can be constructed for generators/IBRs based on the computed margin. Although these methods have a solid theoretical foundation and superior computational efficiency, they normally depend on simplified power systems, thereby restricting their utility in complex applications.

Some studies also involve formulating a transient stability assessment model and determining the control action response to assessment outputs [6], [7]. However, these techniques do not ensure complete assessment reliability, particularly with large-scale systems.

Many studies have recently applied RL to enhance power system transient stability. RL is a technique for solving Markov decision process (MDP) problems. The RL-based methods first formulate the transient stability enhancement as an MDP by building a training environment, then let the RL agent solve the MDP by repeatedly interacting with the environment [8]-[11]. RL does not require the linearity and differentiability of the MDP, offering a viable approach to enhancing transient stability without simplifying the system.

Although RL can effectively enhance the power system's transient stability, it still suffers from poor training convergence and hard to scale. This is particularly pronounced when training encompasses large-number potential contingencies, where the vastness and sparsity of the state space present substantial hurdles to the training convergence. Moreover, when RL is employed to control multiple components, the expansion of the action space further complicates the convergence of training. To address the above issues, we propose a scalable reinforcement learning-based strategy for multiple VSG control considering large-number potential contingencies. The advantages and contributions of the proposed strategy are summarized as follows:

1) A mathematical formulation of the transient stability enhancement problem via RL-based VSG parameter tuning was presented, considering a large number of potential contingency scenarios. A scalable RL framework is proposed to solve this problem, significantly enlarging the number of contingencies that could be handled by RL training.

2) A contingency grouping (CG) algorithm is proposed to identify relevant machines for potential contingencies and exclude irrelevant machines, reducing the complexity of RL's action space and improving the convergence of RL training.

The rest of the paper is organized as follows: Section II provides a new problem formulation of transient stability enhancement via tuning VSGs' parameters. The CG-assisted RL is proposed in Section III as the solution to the formulated problem. Section IV presents case studies of the proposed strategy. Finally, the conclusion is provided in Section V.

II. PROBLEM STATEMENT

A. Transient Stability Enhancement Problem Formulation

The *system-level* stability enhancement can be formulated as an optimization problem to maximize the transient stability

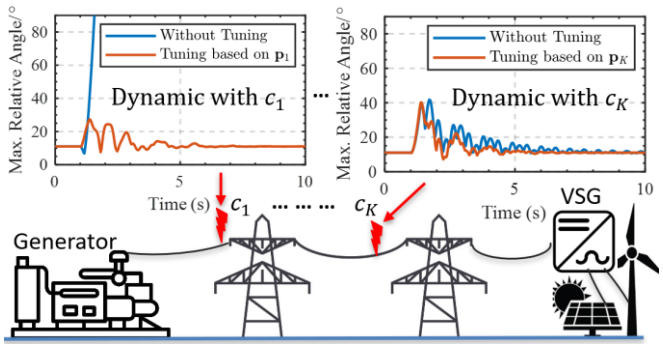

Fig. 1. The stability enhancement problem considers K contingencies.

metric at each potential contingency scenario, as depicted in Fig. 1. Given a K-scenario contingency set for a power grid, the stability enhancement problem is defined as follows:

$$\max_{\mathbf{p}_k} \sum_{k=1}^{K} F_{c_k}\left(\mathbf{x}_{k,1:T}, \mathbf{y}_{k,1:T}\right) \tag{1a}$$

$$\text{s.t. } \dot{\mathbf{x}}_{k,t} = f_{c_k}\left(\mathbf{x}_{k,t}, \mathbf{y}_{k,t}, \mathbf{p}_k\right) \tag{1b}$$

$$0 = g_{c_k}\left(\mathbf{x}_{k,t}, \mathbf{y}_{k,t}, \mathbf{p}_k\right) \tag{1c}$$

where t denotes the time index; T is the length of a time domain simulation; c_k represents the k-th scenario in the contingency set; $\mathbf{x}_{k,t}$ and $\mathbf{y}_{k,t}$ are the transient state variable and algebraic variable with contingency c_k; $\mathbf{p}_k$ are K dimensional parameters corresponding to various contingencies; $F_{c_k}(\,)$ denotes the metric of the system's transient stability with contingency c_k. $f_{c_k}(\cdot)$ and $g_{c_k}(\cdot)$ are differential and algebraic equations which describe the power system model with contingency c_k. Equations (1a)-(1c) constitute the problem of enhancing the system stability by controlling the parameters $\mathbf{p}_k$ against various contingency c_k. The objective (1a) is to maximize the transient metric based on the complete system's trajectories of various contingencies.

B. Transient Stability Metric

The transient stability metric in (1a) can be selected from widely used indicators, e.g., critical clearing time and transient stability index (TSI). In this study, we use TSI as the metric:

$$F_{c_k}\left(\mathbf{x}_{k,1:T}, \mathbf{y}_{k,1:T}\right) = \frac{360^\circ - \left|\delta_{c_k}^{max}\right|}{360^\circ + \left|\delta_{c_k}^{max}\right|} \times 100, \tag{2}$$

where $\delta_{c_k}^{max}$ is the maximum angle difference between any two generators/VSGs at any point of the system's trajectory with contingency c_k. The system is stable when TSI >0.

C. Parameter Tuning of VSG to Enhance Stability

We consider tuning the control parameters of VSGs as the control action to improve the transient stability of the power system. Essentially, VSG is a digital controller for the inverter, can be turned more frequently, and has fewer tuning limitations than conventional generators.

The classical swing equation of rotor dynamics, serving as a foundational component within the VSG model [12], is given as follows

$$M\frac{d\omega}{dt} = T_{m,t} - T_{e,t} - T_{D,t}. \tag{3}$$

where T_m, T_e, and T_D, are the mechanical, electromagnetic, and damping torque of the VSG. M denotes the virtual inertia of the VSG. ω is the virtual rotor speed of the VSG. Equation (3) states that the virtual rotor speed ω can be affected by tuning the virtual inertia M. Considering that the virtual rotor angle and rotor speed are linearly related through the relationship $\delta = 2\pi f(\omega - 1)$ and δ is employed to form TSI in (2), the tuning of M can effectively influence the transient stability metric of the system.

Recent study [10] have shown that it is promising to provide additional support for system stability with VSG by tuning the parameter M. In light of [10], we combine the tuning of parameter M with reinforcement learning via the learning framework proposed in Section III. Therefore, $\mathbf{p}_k$ in this paper is the parameter M setting in multiple controlled VSGs in the system. However, note that the method can be general and extended to other VSG parameters.

III. Contingency Grouping-assisted Reinforcement Learning for VSG Parameter Tuning

This section presents the proposed VSG parameter tuning method for transient stability enhancement. Firstly, we present the scalable reinforcement learning framework for VSG tuning. The modeling and implementation details of the CG model are proposed in Section III. B-C.

A. Scalable Reinforcement Learning Framework

Given the nonlinear dynamic of the system, the uncertainty on power flow, and the large number of contingency cases of problem (1a)-(1c), finding an optimal solution is challenging. Reinforcement learning, without demanding linearity, presents a promising and flexible solution for power system model optimization and parameter tuning. Fig. 2 (a) presents a regular application framework of RL to solve the problem (1a)-(1c), which consists of RL training and control decision-making stages. The RL training lets an agent interact with the power system environment, and the control performance of the agent will be evaluated with rewards. The training objective is to maximize the reward of control in iterations. Since the total contingency number K in (1a)-(1c) is normally large, only $\tilde{K}$ representative contingencies will be considered in the training, encapsulated as random samples in each iteration. The trained agent is applied in the decision-making stage to determine the optimal control action under all K contingencies. The optimal decision calculation is normally time costly due to problem (1a)-(1c) nonlinearity. RL decision-making takes only less than a second, highlighting its efficiency advantage.

Note that the extensive setting of $\tilde{K}$ can lead to challenges such as poor convergence and inefficiency during RL training.

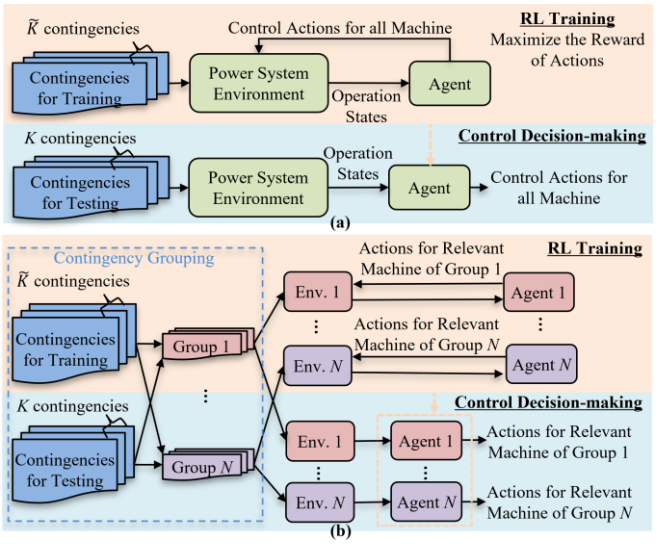

Fig. 2. Compare regular RL and the proposed scalable RL in transient stability improvement. (a) The regular framework to apply RL. (b) The proposed scalable framework to apply RL.

To avoid this problem, many studies maintain $\widetilde{K}$ as a small value and only consider limited contingency in training. For example, the fault location is fixed, and only the fault duration can change in training [8]. In [9], only three types of fault setting are considered. While reducing the number of potential contingencies can improve the training efficiency and stability of RL, too few potential contingencies fail to adequately reflect the transient risks in systems, sacrificing the capability of the RL agent to handle diverse contingencies.

To encompass a large number of potential contingencies in training, we proposed a scalable training framework, as shown in Fig. 2 (b). The proposed framework adds a grouping process ahead of the training. The CG is to identify the relevant machines and group the K contingencies based on their relevant machines. The CG enables contingencies with similar relevant machines to be trained with one independent RL agent, reducing the randomness of states in an environment. Besides, the RL agent will only apply control action on machines relevant to its contingency group, reducing the training difficulty from the action complexity. This training framework is termed CG-RL. The design of CG-RL reduces the number of potential contingencies in each environment, allowing more contingencies to be considered in offline learning. During the online stage, the control action against the grouped contingencies will be determined by utilizing the agent specific to each group.

B. Contingency Grouping Algorithm

The CG method is designed to identify relevant machines for potential contingencies and exclude irrelevant machines in the RL agent's action space, thereby reducing the complexity of RL training. The primary goal of CG is to assist and improve the RL training by increasing both the probability of convergence and the value achieved upon convergence.

The CG algorithm consists of two steps to achieve this goal: i) identify relevant machines for each contingency, and ii) group the contingencies based on their relevant machines.

For step i), we identify the relevant machines by measuring the electrical distance from the machine to the fault. Recent results have reflected that machines closer in electrical distance to the fault contribute more effectively to stabilizing the post-fault system [10]. We used a threshold of electrical distance to separate machines into relevant and irrelevant machines, which measures machines' potential contribution to the fault. However, the measurement of electrical distance poses challenges because transmission grids typically feature meshed connections, and the combined characteristics of line impedance vary between series and parallel configurations.

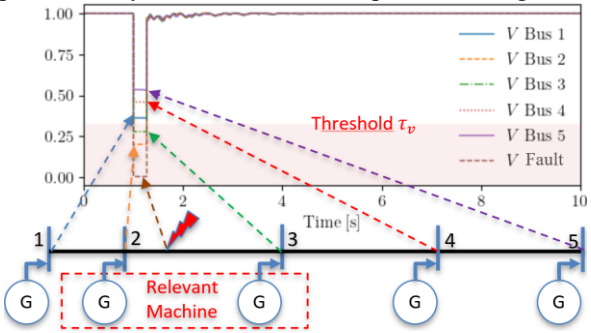

Fig. 3. Illustration of a three-phase fault on a simple 5-bus system: The voltage drops at each VSG point of interconnection during the fault-on period reflect the electrical distance between the VSG and the fault location. If the voltage drops are in a pre-defined threshold τ_v, the corresponding machines 2 and 3 will be identified as **relevant machines** to this fault.

For simplicity, we use the fault-on voltage drops to measure electrical distance. An illustration of relevant machine identification using voltage drops is given in Fig. 3.

For step ii), we formulate a contingency grouping model:

$$\min_{\mu,\, CGI_{k,n},\, GRM_{n,v}} \mu \tag{4a}$$

$$\text{s.t. } \mathrm{RM}_{k,v} \cdot CGI_{k,n} \le GRM_{n,v} \tag{4b}$$

$$\sum_{n=1}^{N} CGI_{k,n} = 1 \tag{4c}$$

$$\sum_{v=1}^{VSGs} GRM_{n,v} \le \mu \tag{4d}$$

where n denotes the n-th group in N contingency groups; v represents the v-th VSG in V VSGs; $\mu \in \mathbb{R}$ is an auxiliary variable that denotes the max number of group-relevant machines in CG; $\boldsymbol{CGI} \in \{0,1\}^{K\times N}$ is a binary matrix that records the group index of contingencies, $CGI_{k,n}$=1 denotes the k-th contingency is allocated to the n-th group; $\boldsymbol{GRM} \in \{0,1\}^{N\times VSGs}$ is a binary matrix that represents the group-relevant machine, $GRM_{n,v}$ =1 denotes the v-th VSG is identified as a group-relevant machine of the n-th group; $\mathbf{RM} \in \{0,1\}^{K\times VSGs}$ is a parameter matrix that records the relevant machines of all contingencies, obtained from step i).

Constraint (4b) specifies that if the k-th contingency is allocated to the n-th group, its relevant machines will be identified as belonging to the n-th group-relevant machines. Constraint (4c) limits each contingency to only one group. Constraint (4d) specifies that μ denotes the maximum number of group-relevant machines. Constraints (4b)-(4d) allocate potential contingencies to their groups and identify group-relevant machines for each group. Objective (4a) minimizes the controlled VSGs in groups, thereby reducing the action space of RL to simplify the training.

The model (4a)-(4d) is formulated as a strictly mix-integer linear programming problem, which can be solved using commercial solvers such as Gurobi or Moseck.

Note that the CG outputs are sensitive to the selection of parameters such as threshold τ_v and number of contingency groups N. The procedure of CG is outlined in Algorithm 1 to select optimal parameters. Rows 4-9 in Algorithm 1 constitute the selection procedure of τ_v. The main idea of this selection

Algorithm 1. The contingency grouping algorithm

Given: contingencies c_k; initial voltage settings of VSG
1 $V_{\text{set},v}$; search step of threshold d; upper limit of group number $\bar{N}$; upper limit of relevant machine number $\bar{\eta}$.
2 Simulate voltage drops at contingency-on period $VD_{k,v}$.
3 Normalize voltage drop: $VD_{k,v} \leftarrow VD_{k,v}/V_{\text{set},v}$
4 Initialize: $\tau_v = d$.
5 **While** $\tau_v \le 1$:
6 Calculate the relevant machine number of each contingency η_k.
7 **If** the $max_{k\in\{1,\dots,K\}}(\eta_k) \le \bar{\eta}$: **break**
8 **Else:** τ_v += d
9 **End While**
10 Calculate the relevant machine matrix **RM** using τ_v.
11 **For** $N = 2,\dots, \bar{N}$
12 Solve (4a)-(4d) and record the N-th minimum $\mu^*_{(N)}$.
13 **End for**
14 Optimal $N \leftarrow \text{argmin}_{N\in\{2,\dots\bar{N}\}}(\mu^*_{(N)})$.
15 **Output:** The group number N; the CG result from solving (4a)-(4d) with the selected N.

is to gradually increase the threshold until the relevant machine number of all contingencies can be bounded in a pre-defined limit. Rows 11-14 in the algorithm select the optimal N within a pre-defined range. The core objective of the parameter selection is to further minimize the maximum number of group-relevant machines.

C. Reinforcement Learning-based VSG Parameter Tuning

The scalable RL decomposes the model (1a)-(1c) into N subproblems. Each subproblem is formulated as an MDP and handled by an independent RL agent.

In terms of model representation, the n-th subproblem $\mathrm{SP}^{(n)}$ is defined as follows:

$$\mathrm{SP}^{(n)} = \left\{ \begin{array}{l} \max \sum_{k\in\mathcal{K}_n} F_{c_k}\left(\mathbf{x}_{k,t}, \mathbf{y}_{k,t}\right) \\ \text{s.t. (1b)-(1c), and (2)} \end{array} \right\} \tag{5}$$

where $\mathcal{K}_n$ denotes the index set of training contingencies allocated to the n-th contingency group.

$\mathrm{SP}^{(n)}$ can be represented by a MDP, which is specified by a tuple $\left(\mathcal{S}^{(n)}, \mathcal{A}^{(n)}, \mathcal{P}^{(n)}\left(s_{\varsigma+1}^{(n)} | s_{\varsigma}^{(n)}, a_{\varsigma}^{(n)}\right), R^{(n)}\left(s_{\varsigma}^{(n)}, a_{\varsigma}^{(n)}\right)\right)$. Where $\mathcal{S}^{(n)}$ is the state space, $\mathcal{A}^{(n)}$ is the action space, $\mathcal{P}^{(n)}$ is the state transition probability, and $R^{(n)}$ is the reward. The index of steps in training episodes is denoted as ς.

1) State Space: States are the contingency, and the TSI corresponds to the current VSGs parameter settings. The state is a partial observation of the operating condition of the system at step ς. The vector of state is defined as:

$$s_{\varsigma}^{(n)} = \left(\mathrm{c}_k, F_{\mathrm{c}_k}\right) \tag{6}$$

where k is sampled from $\mathcal{K}_n$ prior to the initial step and remains constant throughout an episode.

2) Action Space: The RL agent decides the parameters of relevant VSGs of $\mathrm{SP}^{(n)}$. The vector of action is defined as:

$$a_{\varsigma}^{(n)} = \left(M_{v\in\{v|GRM_{n,v}\neq 0\}}\right) \tag{7}$$

where M_v denotes the setting of the v-th controlled VSG.

3) State Transition Probability: The state transition probability encompasses all transitions of states in the RL environment, specifically addressing the problem (5). For each training episode, only one contingency c_k will be sampled and employed. Given that different contingencies are independent of each other, the objective for one episode is to improve the transient metric only against the sampled contingency. The RL agent can improve the transient metric under various contingencies through iterations over multiple episodes, thus fulfilling the objective in (5).

A training episode consists of multiple steps allowing the RL agent to try different VSG parameter settings. The objective of one episode is to search for the optimal VSG parameter setting for the c_k sampled in this episode. As such, each step transition probability is defined as

$$\mathcal{P}^{(n)}\left(s_{\varsigma+1}^{(n)} \text{ s.t. } \left\{\mathbf{p}_k = a_{\varsigma}^{(n)}, \text{(1b)-(1c), (2)}\right\} | s_{\varsigma}^{(n)}, a_{\varsigma}^{(n)}\right). \tag{8}$$

For each step, the RL agent will pick a set of VSG parameters based on the transient stability metric of the current step. Then, a complete time domain simulation will be simulated to calculate the transient stability metric based on the VSG setting picked by the RL agent, subject to (1b)-(1c) and (2). The calculated transient stability metric will be used as the next step state. It should be noted that the state transition is probabilistic, considering the randomness of pre-contingency power flow in different episodes.

4) Reward Function and Stopping Criteria: The reward evaluates the RL agent's capability to improve the system transient metric. Thus, the reward function is defined as:

$$R_{\varsigma}^{(n)} = \begin{cases} 1 - \dfrac{\varsigma - 1}{\Sigma}, & if\ F_{\mathrm{c}_k}^{(\varsigma+1)} > F_{\mathrm{c}_k}^{(\varsigma)} \\ 0, & if\ F_{\mathrm{c}_k}^{(\varsigma+1)} = F_{\mathrm{c}_k}^{(\varsigma)} \\ -1 + \dfrac{\varsigma - 1}{\Sigma}, & if\ F_{\mathrm{c}_k}^{(\varsigma+1)} < F_{\mathrm{c}_k}^{(\varsigma)} \end{cases} \tag{9}$$

where Σ denotes the maximum step number of an episode, and reward (9) indicates that the RL agent will receive a positive reward if the action improves the transient metric. Otherwise, it will receive a zero or negative reward. The term $(\varsigma - 1)/\Sigma$ motivates the agent to initiate improvement actions promptly.

The stopping criterion for one episode is twofold: i) the episode reaches step Σ; ii) the agent receives a negative reward.

5) Implementation of RL Algorithms: RL is a technique for solving MDP problems by learning to make decisions that maximize cumulative rewards. Given (5) is formulated as an MDP, any state-of-the-art RL algorithms can be applied as the solution of (5). The transient stability enhancement problem (1) consists of N of subproblem (5). Therefore, parallel training using RL algorithms is required across N environments. The selection of RL algorithms is flexible. Different environments can use distinct RL algorithms.

IV. Case Studies

This section is dedicated to the case studies on a Modified Kundur system and a Modified NPCC system to demonstrate the effectiveness of the proposed scalable RL method. Test configuration, comparative experiments, and scheme analysis are presented. The transient simulations are conducted using ANDES [13]. The model and parameters of used transient models can also be found in ANDES. The Proximal Policy Optimization (PPO)-based RL agents are applied via the open-source tool Tianshou [14]. The max training episodes is 1000, and the max steps of an episode Σ is 10. All tests are conducted with an Intel Core i7 CPU.

A. Case I: The Modified Kundur System

To demonstrate the execution process of the proposed method, we start with a simple Modified Kundur system, as shown in Fig. 4. The system consists of 10 buses and 4 VSGs with identical settings. The system is divided into two areas: Area 1 comprises buses 1, 2, and 5 to 7, while Area 2 comprises the remaining buses. The generation and load on the two areas are symmetrical. We assume the three-phase-to-ground fault contingency can occur at any location on the lines in Modified Kundur, and the fault duration is a random value of 0.05-0.2 seconds. Loads on bus 7 and bus 8 were randomly

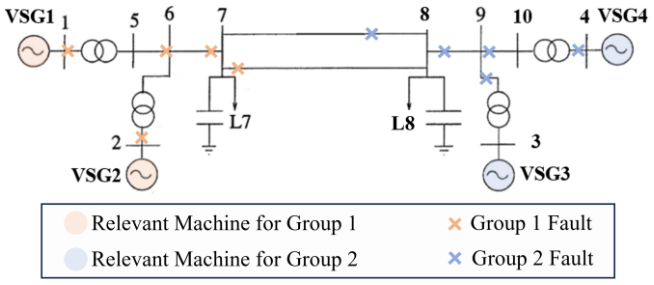

Fig. 4. An example of contingency grouping output on Modified Kundur.

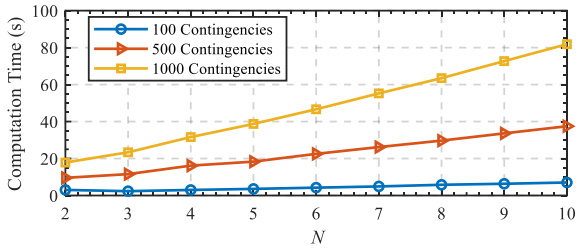

Fig. 5. Efficiency of the contingency grouping model on Modified Kundur with different numbers of potential contingencies $\widetilde{K}$ and group numbers N.

initialized within 95%-105% of the default setting. The RL agent is applied to control the transient parameters of each VSG in the system. In this study, we use a discrete action space: $M_{vsg=1,..,4} \in \{M_{vsg}^{dflt}, M_{vsg}^{\uparrow}\}$, which means the RL agent will pick the optimal parameter for each VSG. Where M_{vsg}^{dflt} and $M_{vsg}^{\uparrow}$ denote the default and increased settings of the VSG's virtual inertia. Since we have 4 VSGs, the space scale for the regular PPO agent is 2^4.

Fig. 4 also depicts a grouping output example of 10-contingency on Modified Kundur. As observed, contingencies are grouped into 2 groups based on areas, and the 2-VSG in each area are identified as the group-relevant machines. In this case, we need two independent RL agents to handle the two-group contingencies. The group 1 agent will control the parameter setting of VSG1 and VSG2, and the group 2 agent will control the setting of VSG3 and VSG4. Therefore, the space scale for each agent reduces to 2^2. Given the contingencies in each group are basically limited to one region, the diversity of the state that each RL agent may observe is also limited, reducing the complexity of the training task.

Fig. 5 shows the computation times required to solve the contingency grouping model (4) on Modified Kundur. The solution's efficiency is sensitive to the pre-setting of parameters. Increasing $\widetilde{K}$ and N both lead to longer computation times. The contingency grouping model can be solved within 2 minutes, deemed acceptable for an offline computation task. On average, the 1000-episode RL training on Modified Kundur takes 43.3 minutes. Compared to RL training, the computation time of (4) is negligible.

For comparison, we benchmark the proposed CG-assisted PPO (CG-PPO) against the regular application of PPO. Table I compares the convergence probability using PPO and CG-PPO on Modified Kundur. Given that CG-PPO undertakes two parallel training sessions to encompass all contingencies, a CG-PPO training session is deemed converged only if both sessions achieve convergence. Considering the relatively simple constitution of Modified Kundur, the application of PPO and CG-PPO yields comparable results in convergence probabilities. The convergence probability based on PPO drops to 95% with 500 considered contingencies and further to 85% as contingencies increase to 1,000. This demonstrates a clear trend where the convergence probability of PPO training diminishes as more contingencies are considered in the training process. Fig. 6 compares moving average rewards using PPO and CG-PPO, with 100 considered contingencies in training. Moving average rewards monitor the performance of RL agents in the iteration of episodes, which reflects the training progress of RL algorithms. Rewards based on CG-PPO exhibit smaller fluctuations and converge to higher values than those based on PPO. This indicates that CG-PPO training is more stable, and trained agents via CG-PPO can perform better in decision-making. Table I and Fig. 6 verify that CG-PPO can improve the convergence of regular PPO.

TABLE I. CONVERGENCE PROBABILITY ON MODIFIED KUNDUR

Considered Contingencies	Convergence Probability	
	PPO	CG-PPO
100	20/20 (100%)	**20/20 (100%)**
500	19/20 (95%)	**20/20 (100%)**
1,000	17/20 (85%)	**20/20 (100%)**

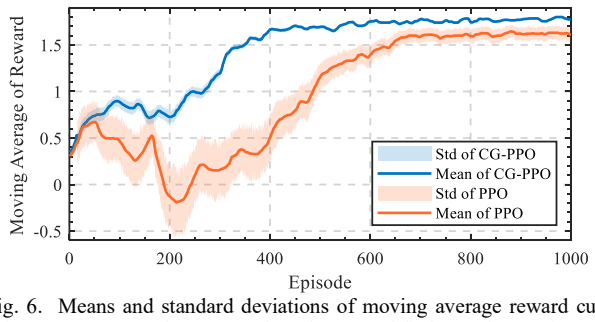

Fig. 6. Means and standard deviations of moving average reward curves using pure PPO and CG-PPO based on Modified Kundur environment. Higher reward denotes the RL agent have better performance in training.

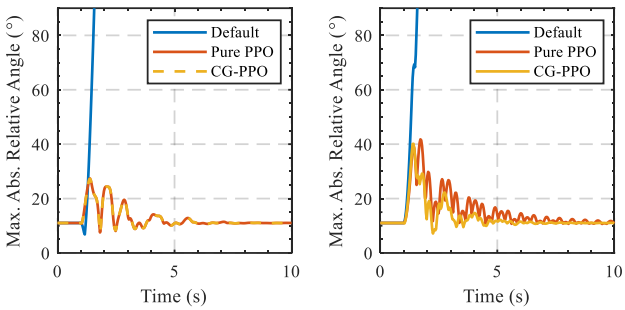

Fig. 7. Examples of transient dynamics on Modified Kundur. Ex. 1: fault on line 8-9, 50% along its electrical distance, lasting 0.35s (left). Ex. 2: fault on line 5-6, 10% along its electrical distance, lasting 0.35s (right).

We conduct simulations for 1,000 contingencies to verify the trained agents on Modified Kundur. Three types of parameter settings are compared: default parameters, pure PPO-based parameters, and CG-PPO-based parameters. Fig. 7 shows examples of transient dynamics in Modified Kundur. The maximum absolute relative angle of all machines in the system is used to demonstrate the synchronization level. The system loses synchronization when the maximum absolute relative angle exceeds 90 degrees. In example 1, machines with default parameter settings lose synchronization after the fault, while PPO and CG-PPO agents have the same control action. Example 2 shows that CG-PPO outperforms PPO in

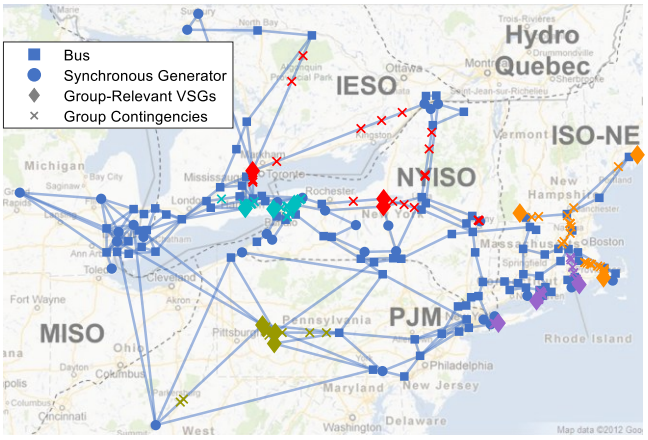

Fig. 8. An example of contingency grouping output on modified NPCC. Distinct colors represent contingencies and relevant VSGs across 5 groups.

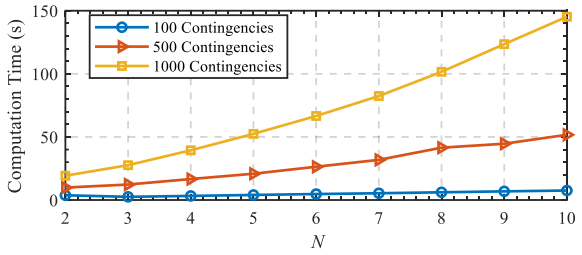

Fig. 9. Efficiency of the contingency grouping model on Modified NPCC with different numbers of potential contingencies $\widetilde{K}$ and group numbers N.

performance. The average TSIs of 1,000 contingencies based on three parameter settings are 23.07 (default), 32.44 (PPO), and **37.12** (CG-PPO), respectively, which also proves the effectiveness of the proposed CG-RL.

B. Case II: The Modified NPCC 140-bus System

The proposed CG-RL is also verified on the Modified NPCC system, as shown in Fig. 8. The system consists of 140 buses and 48 machines with identical settings. Generators on buses 21, 22, 24, 26, 27, 50, 51, 55, 57, 79, 91, 101, 133, 134, and 135 are replaced as VSGs, respectively. We assume the three-phase-to-ground fault contingency can occur at any location on lines 121, 95, 207, 9, 213, 15, 20, 22, 18, 13, 1, 55, 54, 45, 41, 48, 136, 130, 231, 66, 223, 65, 119,120, 198, 202, 201, and 204. The fault duration is a random value of 0.03-0.13 seconds. Loads on buses connected to the fault were randomly initialized within 95%-105% of the default setting. The RL agent is applied to control 15 VSGs. Thus, the space scale for the regular PPO agent is 2^{15}.

Fig. 9 shows the computation times required to solve the contingency grouping model (4) on Modified NPCC. The solution's efficiency shows a similar trend as in Fig. 5. Since Modified NPCC is much more complex, solving (4) on Modified NPCC takes longer than on Modified Kundur. However, the solution can generally be completed within 2.5 minutes, which is acceptable for an offline computation.

Table II compares the convergence probability using PPO and CG-PPO on Modified NPCC. The difference between PPO and CG-PPO in convergence probabilities can be seen as Modified NPCC is much more complex than Modified Kundur. The convergence probability based on PPO drops to less than 20% with complex contingencies in training.

TABLE II. CONVERGENCE PROBABILITY ON MODIFIED NPCC

Considered Contingencies	Convergence Probability	
	PPO	CG-PPO
100	3/20 (15%)	**20/20 (100%)**
500	1/20 (5%)	**20/20 (100%)**
1,000	1/20 (5%)	**20/20 (100%)**

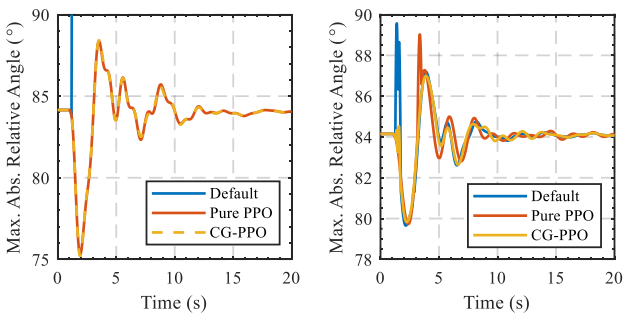

Fig. 10. Examples of transient dynamics on Modified NPCC. Example 1: fault on line 213, 80% along its electrical distance, lasting 0.07s (left). Example 2: fault on line 95, 80% along its electrical distance, lasting 0.13s (right).

We conduct transient simulations for 1,000 contingencies to verify trained agents on Modified NPCC. Fig. 10 shows two examples of transient dynamics. The average TSIs of 1,000 contingencies based on various parameter settings are 44.13 (default), 45.88 (PPO), and **46.75** (CG-PPO), respectively. These results all show the effectiveness of CG-PPO.

V. CONCLUSIONS

This paper proposed a scalable RL framework to enhance power system transient stability by tuning multiple VSGs' parameters, enabling a large number of contingencies to be considered in RL training. Specifically, a CG algorithm is added to the regular RL to reduce the complexity and improve the convergence of RL training. The proposed framework was applied and verified in two standard test systems. The RL training assisted by the CG algorithm maintains a 100% convergence probability compared to merely 5% without the CG algorithm. Additionally, the proposed framework yields the TSI improvement on both test configurations.

ACKNOWLEDGMENT

This work was supported by the U.S. DOE's Office of Energy Efficiency and Renewable Energy under the Solar Energy Technologies Office Award Number DE-EE0009341.

REFERENCES

[1] B. Tan, and J. Zhao, "Data-Driven Time-Varying Inertia Estimation of Inverter-Based Resources," *IEEE Trans. Power Systems*, vol. 38, no. 2, pp. 1795-1798, Mar. 2023.

[2] T. Su et al., "Deep Sigma Point Processes-Assisted Chance-Constrained Power System Transient Stability Preventive Control," *IEEE Trans. Power Systems*, vol. 39, no. 1, pp. 1965-1978, Jan. 2024.

[3] A. Bonfiglio et al., "Improving power grids transient stability via model predictive control," in *Proc. 18th Power Syst. Comput. Conf.*, 2014, pp. 18–22.

[4] P. Bhui and N. Senroy, "Real-time prediction and control of transient stability using transient energy function," *IEEE Trans. Power Syst.*, vol. 32, no. 2, pp. 923–934, Mar. 2017.

[5] Z. Shuai et al., "Transient Angle Stability of Virtual Synchronous Generators Using Lyapunov's Direct Method," *IEEE Trans. Smart Grids*, vol. 10, no. 4, pp. 4648-4661, Jul. 2019.

[6] S. Yang, Z. Hao, B. Zhang, and M. Hojo, "An Accurate and Fast Start-Up Scheme for Power System Real-Time Emergency Control," *IEEE Trans. Power Syst.*, vol. 34, no. 5, pp. 3562-3572, Sept. 2019.

[7] L. Zhao et al., "An online power system transient stability assessment method based on graph neural network and central moment discrepancy," *Front. Energy Res.*, vol. 11, p. 1082534, Feb. 2023.

[8] Q. Huang et al., "Adaptive Power System Emergency Control Using Deep Reinforcement Learning," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1171-1182, Mar. 2020.

[9] Y. Chen et al., "Distributed Hierarchical Deep Reinforcement Learning for Large-Scale Grid Emergency Control," *IEEE Trans. Power Systems*, vol. 39, no. 2, pp. 4446-4458, Mar. 2024

[10] X. Huang et al., "Transient Stability Preventive Control via Tuning the Parameters of Virtual Synchronous Generators," in *Proc. 2023 IEEE Power & Energy Society General Meeting*, 2023, pp. 1-5.

[11] Z. Liu, Y. Liu, J. He, X. Huang, and Z. Ding, "Double DQN-based Power System Transient Stability Emergency Control with Protection Coordinations," in *Proc. 2023 IEEE 6th International Electrical and Energy Conference*, 2023, pp. 4182-4187.

[12] D. Li, Q. Zhu, S. Lin, and X. Y. Bian, "A Self-Adaptive Inertia and Damping Combination Control of VSG to Support Frequency Stability," *IEEE Trans. Energy Conver.*, vol. 32, no. 1, pp. 397-398, March 2017.

[13] H. Cui, F. Li, and K. Tomsovic, "Hybrid Symbolic-Numeric Framework for Power System Modeling and Analysis," *IEEE Trans. Power Syst.*, vol. 36, no. 2, Mar. 2021.

[14] J. Weng, H. Chen, D. Yan, K. You, A. Duburcq, M. Zhang, Y. Su, H. Su, and J. Zhu, "Tianshou: A highly modularized deep reinforcement learning library," *J. Mach. Learn. Res.*, vol. 23, no. 261, pp. 1-6, 2022.