Predicting Power System Dynamics and Transients: A Frequency Domain Approach

Wenqi Cui, Weiwei Yang and Baosen Zhang

Abstract—The dynamics of power grids are governed by a large number of nonlinear ordinary differential equations (ODEs). To safely operate the system, operators need to check that the states described by this set of ODEs stay within prescribed limits after various faults. Limited by the size and stiffness of the ODEs, current numerical integration techniques are often too slow to be useful in real-time or large-scale resource allocation problems. In addition, detailed system parameters are often not exactly known. Machine learning approaches have been proposed to reduce the computational efforts, but existing methods generally suffer from overfitting and failures to predict unstable behaviors.

This paper proposes a novel framework for power system dynamic predictions by learning in the frequency domain. The intuition is that although the system behavior is complex in the time domain, there are relatively few dominate modes in the frequency domain. Therefore, we learn to predict by constructing neural networks with Fourier transform and filtering layers. System topology and fault information are encoded by taking a multi-dimensional Fourier transform, allowing us to leverage the fact that the trajectories are sparse both in time and spatial (across different buses) frequencies. We show that the proposed approach does not need detailed system parameters, speeds up prediction computations by orders of magnitude and is highly accurate for different fault types.

I. INTRODUCTION

Increasing the amount of renewable resources integrated in the electric grid is fundamental to reducing carbon emissions and mitigating climate change. Many governments and companies have set ambitious goals to generate their electricity with close to 100% renewables by 2050 [1]. So far, much of the attention has been paid to increasing the aggregate generation capacities [2], [3]. However, increased capacities do not necessarily imply more load are actually served by renewable resources [4].

A key factor that limits the actual power generated by renewables is dynamic stability of the grid. A grid can be thought as a large interconnected system of generators, loads and power electronic components, governed by nonlinear ordinary differential equations (ODEs) [5]. This system also undergo constant disturbances, from load changes to line outages [6]. The predominant goal of power system operations is to make sure that the system stays within acceptable limits under these disturbances.

The system states are governed by the shifted ODEs during the transient across pre-fault, fault-on and post-fault stages [7]. Fig. 1 shows a tripped line in Florida that led to rolling blackouts that impacted lower two-thirds of state [8]. Ideally, a system should withstand these types of single events without performance degradation \((N-1)\) security) [6], but this was not the case in Fig. 1. The fundamental reason is that solving the governing ODEs is extremely computationally challenging and not all contingencies can be checked. These ODEs are highly nonlinear and numerical methods (e.g., implicit and explicit integration) are used to solve them [9]. To study transient stability, numerical algorithms need to use fairly fine discretizations. For a moderately sized system, existing numerical solvers take minutes to simulate only seconds of system trajectories [10].

Consequently, only a limited number of scenarios are studied offline and operators tend to restrict the system to operate close to these scenarios. As shown in Fig. 2 under certain conditions of power generation, demand and topology, system operators simulate the system trajectories after disturbances through solving the shifted ODEs. By studying scenarios of system conditions offline, system operators will have a checklist about what action needs to be taken in real time to ensure that the systems’ states are within the permissible range after critical contingencies. Since renewables have much larger uncertainties than conventional resources, operators often curtail them to artificially limit their generation to avoid operating at “unknown” regions [5]. For example, some European grids are not allowed to operate at above 40% wind, no matter how much wind is actually blowing [11]. Therefore, fast and accurate dynamic simulations would greatly increase the actual utilization of renewables in the system.

Recently, machine learning (ML) approaches have been proposed for dynamical simulation instead of ODE solvers and can reduce the computation time by orders of magnitude. Given the fault information and present measurement of states, ML methods learn feed-forward neural networks to predict the future trajectories. Most works focus on classification to iden-
Fig. 2. Power system operators conduct dynamic power system transient prediction by solving ODEs \[12\]. Under certain conditions of power generation, demand and topology, system operators simulate the system trajectory after disturbances through solving the shifted ODEs. By studying scenarios of system conditions offline, system operators will have a checklist about what action needs to be taken in real time to ensure that the systems’ states are within the permissible range after critical disturbances.

To alleviate the limitation of binary prediction, some recent works provide finer trajectory prediction by learning the time domain solutions to the governing ODEs. Polynomial basis are used in [15] to approximate the solutions, but the number of basis functions grows exponentially with the system size. Deep neural networks are used in [16], [17] to directly learn to predict the future trajectory from past and current measurements. However, since power system are large and sampled sparsely in time, direct regression on time-domain data does not perform very well. A rolling prediction is used in [16] to improve accuracy, but with a high cost in computational efficiency.

More importantly, since the majority of trajectory used in training is stable, the learned networks fail predict unstable behaviors. Physics informed neural networks that directly attempt to solve the ODEs have been proposed as an alternative [18], but it does not currently scale beyond small networks.

In this paper, we propose a novel framework for power system dynamic prediction by learning in the frequency domain. From the observation that after a disturbance, the states will undergo sinusoidal oscillations, we propose to make predictions in the frequency domain. We adopt and extend the structure of Fourier Neural Operator in [19] to learn in the frequency domain and recover the time domain trajectories through the inverse Fourier transform. Specifically, we design the dataframe to encode the power system topology and fault information, which lead to a 3D Fourier transform. This method is able to make smooth and accurate predictions, capturing both stable and unstable behaviors without the need to manually tune the training data. It improves the MSE prediction error by more than 50% compared with state-of-the-art AI methods, and vastly improves the detection of unstable behavior. Furthermore, it provides a computation speed up of more than 400 times compared to existing power system tools. Code and data are available at https://github.com/Wenqi-Cui/Predict-Power-System-Dynamics-Frequency-Domain.

In summary, the main contributions of the paper are:
1) We propose a novel dynamic transient prediction approach for learning in the frequency domain, which can accurately predict state trajectories based on a few measurements.
2) We develop a dataframe that encodes spatial-temporal information about the system topology, which greatly reduces the computational complexity in multi-dimensional Fourier transforms.
3) The time-varying active/reactive power injection and fault-on/clear actions are incorporated in the proposed framework, enabling the prediction of the step changes from fault-on to post-fault dynamics.

II. MODEL AND PROBLEM FORMULATION

A. Power System Swing Equations

For a system with \( N \) buses, the main variables of interest are the angle \( \delta_i \), frequency deviation \( \omega_i \) and voltage \( V_i \) at each bus \( i \). The states of power systems are described by a three-tuple, denoted by \( s = (\delta, \omega, V) \). The dynamics of the states depend on a myriad of components including governors, exciters, stabilizers, etc., as illustrated in Fig. 3. Each component is described by a set of ODEs. Compactly, the evolution of states \( s \) is \( \dot{s} = f(s) \) with \( f(\cdot) \) being the system of ODEs. If the initial condition is \( s_0 \), we write \( x = f(s; s_0) \).

Here we give an example of what \( f \) looks like. The susceptance and conductance of the line between bus \( i \) and \( j \) are \( B_{ij} \) and \( G_{ij} \), respectively. Let \( P_i \) be the net power
injection at bus $i$. The inertia and damping at bus $i$ are $M_i$ and $D_i$, respectively [20]. After simplifying the response from governors, exciters and stabilizers, the system is described by

$$\delta^i = \omega^i$$

\[ M_i \dot{\omega}^i = P_i - D_i \omega^i - \sum_{j=1, j \neq i}^N V_j V_j B_{ij} \sin(\delta_i - \delta_j) - \sum_{j=1, j \neq i}^N V_j V_j G_{ij} \cos(\delta_i - \delta_j) \]

\[ T_{d_{di}} \dot{V}_i = \frac{E_{f_{di}}}{x_{di} - x'_{di}} \left( \frac{V_i}{x_{di} - x'_{di}} - \frac{1}{x_{di} - x'_{di}} \right) - \sum_{j=1, j \neq i}^n \sum_{j=1, j \neq i}^n V_j \left( B_{ij} \cos \delta_{ij} + G_{ij} \sin \delta_{ij} \right) \]

where $T_{d_{di}}$, $x_{di}$, $x'_{di}$ and $E_{f_{di}}$ are constants for the synchronous generator at bus $i$ [7], [9].

Under changing generator and load conditions, power systems are operated to withstand the occurrence of certain contingencies. To ensure that cascading outages will not occur for the set of critical disturbances, the state variables need to stay within permissible ranges during the transient process of power system after disturbances [7], [21]. Note that in reality, equations with higher order than $\dot{s}=s(t)$ are used to accurately simulate the dynamics of power system resulted from the interactions of components such as governors and exciters. Therefore, each bus is often described by more than 10 equations. For even a moderate power system with tens or hundreds of buses, it may be governed by hundreds or thousands of ODEs.

### B. Transient Dynamics

Disturbances lead to deviations in the states $s$ through a step change of parameters in $s_{t_0}$. For example, a short circuit on a transmission line between the bus $i$ and $j$ will result in a sudden change of the terms $B_{ij}$ and $G_{ij}$, depending on the specific fault type (e.g., single-phase-to-ground, two-phase-to-ground, line-to-line, three-phase-to-ground, etc.). Because of uncertainties in power generation and demand, the active and reactive power are treated as variables, denoted by $a = (P, Q)$.

Suppose a fault happens at the time $t_f$ and is cleared at the time $t_{cl}$. The pre-fault stage is defined as the period before the fault happens at $t_f$. The system evolves from the initial state $s(t_0)$ subject to parameter variations as:

$$\dot{s} = f(s, a; s(t_0)), \quad t_0 < t < t_f$$

The sudden parameter changes after disturbances will lead to a shift of the swing equations. The fault-on system evolves with $k$ subsequent actions from system relays and circuit breaks. Suppose the $j$-th action is taken at $t_{F,j}$, the fault-on system is described by several set of equations [7]

$$\dot{s} = f_{F,1}(s, a; s(t_f)), \quad t_f < t < t_{F,1}$$

$$\dot{s} = f_{F,2}(s, a; s(t_{F,1})), \quad t_{F,1} < t < t_{F,2}$$

$$\vdots$$

$$\dot{s} = f_{F,k}(s, a; s(t_{F,k})), \quad t_{F,k} < t < t_{cl}$$

The post-fault stage refers to the system after the fault is cleared. The states of the post-fault system evolve with the same differential equation of the pre-fault system starting from the post-fault initial state $s(t_{cl})$

$$\dot{s} = f(s, a; s(t_{cl})), \quad t > t_{cl}$$

Let $u = 1, \ldots, m$ denote different stages of the ODEs corresponding to the type of the fault as well as fault-clear actions. The dynamics \[(2) - (4)\] is written in a compact form as

$$\dot{s} = f_u(s, a; s_{t_0})$$

where $f_u$ stands for the state transition function with respect to the fault, encoded by $u$.

The key to power system dynamics is to predict the future of the system trajectories, given the fault information, some measured observations $s_{t_0}$ and the expected clearing actions in $u$. Based on these trajectories, interim actions like load shedding or emergency generation can be taken to reduce the impact of the faults [6], [9].

### C. Current Approaches and Challenges

Current approaches in power system dynamic prediction is based on solving $\dot{s}=s(t)$. These equations are nonlinear and lack closed-form solutions. System operators typically rely on numerical integration, such as Runge-Kutta (RK) methods and trapezoidal rule, to iteratively approximate the solution of $\dot{s}=s(t)$ in small time intervals [9]. However, because of the highly nonlinear nature of the ODEs, very fine discretization steps are required for these numerical methods. As a result, these approaches may be too slow for real-time decision-making. For a moderately sized system, existing numerical solvers take thousands of ODEs.

As an alternative, system operators also use manual heuristics to take actions, but this strategy is becoming less tenable as renewables introduce distinctly different operating scenarios.

1The disturbance such as a short circuit on a transmission line is automatically cleared by protective relay operation after a certain amount of time.
III. LEARNING FOR DYNAMIC TRANSIENT PREDICTIONS

A. Problem Setup

Learning-based approaches try to find a mapping from present measurements to future trajectories. The predictions are then obtained through function evaluations, which significantly reduces the computational time compared to the conventional numerical approaches.

The problem we are interested in is to predict the trajectory of the states starting at the time step \( t_{on} \) for \( t_{out} \) number of time steps. The input are \( \tau_{in} \) observations of the states from \( s(t_{on} - \tau_{in}) \) to \( s(t_{on}) \), the net injections from \( \alpha(t_{on} - \tau_{in}) \) to \( \alpha(t_{on}) \) and the fault and control actions from \( u(t_{on} - \tau_{in}) \) to \( u(t_{on}) \). We write \( s_{in} = (s(t_{on} - \tau_{in}), \ldots, s(t_{on})), \alpha = (\alpha(t_{on} - \tau_{in}), \ldots, \alpha(t_{on})), u = (u(t_{on} - \tau_{in}), \ldots, u(t_{on})) \) and \( s_{out} = (s(t_{on}), \ldots, \theta(t_{on} + t_{out})) \) with the goal of predicting \( s_{out} \) from \( (s_{in}, \alpha, u) \).

A learning-based approach finds a mapping \( G \) from the space of input trajectories \( s_{in} \) to output trajectories \( s_{out} \). In this paper we consider parameterized models, which are typically (deep) neural networks with parameters \( \theta \). The prediction is then given by

\[
\hat{s}_{out} = G_{\theta}(s_{in}, \alpha, u). \tag{6}
\]

The exact form of \( \theta \) depends on the structure of neural network used.

Let \( H \) be the batch size and \( s_{out}^i \) be the prediction of the \( i \)-th sample for \( i = 1, \ldots, H \). The weights of neural network \( \theta \) are updated by back-propagation to minimize loss function \( L(\theta) \) defined by the mean absolute percentage error (MAPE) between predicted trajectory and the actual trajectory \[19\]

\[
L(\theta) = \frac{1}{H} \sum_{i=1}^{H} \frac{||\hat{s}_{out}^i - s_{out}^i||_1}{||s_{out}^i||_1} \tag{7}
\]

B. Current ML Approaches and Limitations

Learning the power system transient dynamics is not trivial because states undergo nonlinear oscillations. Predictions with three existing approaches are illustrated in Fig. 4. The blue line is the trajectory of the frequency deviation on a bus before and after a fault. The blue squares are the accurate dynamics and the grey block is the prediction horizon.

A standard approach is to use a neural network to learn the time-domain mapping from the input to output. Given the input trajectory \( x \) with \( \tau_{in} \) measurement points, past trajectories are used to learn a neural network (NN) that maps \( x \) to the space of output trajectory with \( \tau_{out} \) measurement points. As illustrated in Fig. 4(a) and Fig. 4(b), purely learning in the time domain will easily overfit and cannot learn a smooth curve like the true trajectories. More importantly, generic machine learning approaches prone to false negative errors. Since the vast majority of historical trajectories are stable, a ML method tends to not predict unstable trajectories. This would lead to catastrophic consequences if the system operator does not take actions to mitigate instabilities.

Similarly, fitting the nonlinear dynamics with polynomial basis will also easily lead to over-fitting, as illustrated in Fig. 4(c). Recently, physics-Informed Neural Networks have been proposed in \[22\] to learn solutions that satisfy equations from implicit Runge-Kutta (RK) integration. This approach has been applied to power system swing dynamics in \[18\]. Since RK method is the weighted sum of ODE solutions in discretized intervals, its accuracy decreases sharply when predicting trajectories with large oscillations for a longer horizon (e.g., larger than 1 second), as illustrated in Fig. 4(d) (the prediction errors are larger than then limit of the y-axis).

IV. LEARNING IN THE FREQUENCY DOMAIN

A. System Dynamics in the Frequency Domain

Because of the above challenges when learning in the time domain, we propose a new approach for learning power system transient dynamics in the frequency domain. Here we use a simplified model for transient dynamics to illustrate the intuition for this approach. Under the DC power flow model that assumes a lossless power network and constant voltage at each bus, the dynamics of the system is reduced to \[9\]

\[
\dot{\delta}_i = \omega_i \tag{8a}
\]

\[
\dot{\omega}_i = P_i - D_i \omega_i - \sum_{j=1, j \neq i}^{N} B_{ij}(\delta_i - \delta_j) \tag{8b}
\]

The equations \(8\) are with form of first-order ODEs \( \dot{s} = As \) whose solution consists of the terms \( e^{(\lambda_k) t} \) for \( k = 1, \ldots, K \), with \( K \) be the number of distinct eigenvalues of the state transition matrix \( A \) and \( \lambda_k \) be the \( k \)-th eigenvalue. Note that \( \lambda_k \) are complex numbers, the terms \( e^{(\lambda_k) t} \) for \( k = 1, \ldots, K \) bring nonlinear oscillations. Because of the finite set of eigenvalue, there is a finite number of modes in the frequency domain. As a result, the trajectories are sparse in the frequency domain, making it easier to learn after Fourier transform.

Moreover, the extra nonlinear terms in \(1\) compared with \(8\) come from \( \sin(\delta_{ij}) \) and \( \cos(\delta_{ij}) \). Therefore, the

\[A\text{ transition matrix} K \text{ with} \]
\[\text{be the number of distinct eigenvalues of the state} \]
\[\text{and } \lambda_k \text{ be the } k \text{-th eigenvalue. Note that } \lambda_k \text{ are complex numbers, the terms } e^{(\lambda_k) t} \text{ for } k = 1, \ldots, K \text{ bring nonlinear oscillations. Because of the finite set of eigenvalue, there is a finite number of modes in the frequency domain. As a result, the trajectories are sparse in the frequency domain, making it easier to learn after Fourier transform.} \]

\[\text{Moreover, the extra nonlinear terms in } 1 \text{ compared with } 8 \text{ come from } \sin(\delta_{ij}) \text{ and } \cos(\delta_{ij}). \text{ Therefore, the} \]
sinusoidal basis in Fourier transform (and inverse Fourier transform) is a natural fit for the nonlinearity of power system transient dynamics.

### B. Fourier Neural Operator

For a system with $N$ buses, there are $3N$ state variables: the voltage, angle and frequency at each bus. The input trajectory is the function of state $s$, written as $s_m = g_0(s)$. Let $l$ be the number of layers in a neural network. The input of the $j$-th layer is $g_{j-1}(s)$ and the output is $g_j(s)$ for $j = 1, \cdots, l$. For the input of each layer, we conduct discrete Fourier transform $F$ to convert the input trajectory into the frequency domain $\mathbb{C}^{k_{\text{max}} \times 3N \times 3N}$. Inspired by the work in [19], we use neural networks parameterized by $\phi$ to learn in the frequency domain, and then recovered the time-domain sequences by inverse Fourier transform $F^{-1}$. This process is defined as Fourier neural operator $K(\phi)$ represented by

$$K(\phi)g_{j-1}(s) = F^{-1}(R(\phi \cdot \psi(0Fg_{j-1}))(s),$$

where $\psi$ is a low-pass filter that truncating the Fourier series at a maximum number of modes $k_{\text{max}} = | \{ k \in \mathbb{Z}^{3N} : |k| \leq k_{\text{max}}, i \text{ for } i = 1, \ldots, 3N \} |$ for efficient computation [19]. Then $R(\phi \cdot \psi(0Fg_{j-1})) \in \mathbb{C}^{k_{\text{max}} \times 3N \times 3N}$ is the weight tensor conduct linear combination of the modes in the frequency domain defined by

$$R(\phi \cdot \psi(0Fg_{j-1}))_{k,i} = \sum_{v=1}^{3N} R(\phi \cdot \psi(0Fg_{j-1}))_{k,i,v}$$

for $k = 1, \ldots, k_{\text{max}}, i = 1, \ldots, 3N$ and $j = 1, \ldots, l$.

The output of the $j$-th layer adds up Fourier neural operator with the initial time-domain sequence weighted by $W_{j-1} : \mathbb{R}^{3N} \rightarrow \mathbb{R}^{3N}$ to recover non-periodic modes

$$g_j(s) = \sigma(W_{j-1}g_{j-1}(s) + K(\phi)g_{j-1}(s))$$

where $\sigma$ is a nonlinear activation function whose action is defined component-wise. The prediction is obtained from a dense combination of the last layer with weight $W_0$ given by

$$\hat{s}_{\text{out}} = W_0g_l(s).$$

### C. Multi-dimensional Fourier transforms

The above approach of learning the weights in the frequency domain and recover the trajectory with inverse Fourier transform provides the advantage in fitting oscillatory functions, by learning smooth curvatures and avoiding over-fitting. However, the states are in $\mathbb{R}^{3N}$ and conducting Fourier transform with $3N$ dimensions is time consuming even for moderately sized power systems.

Another choice is to neglect the dependence and purely conduct 1D Fourier transform on the time dimension. However, this will reduce the prediction performance since the networked structure is an important cause of the oscillations. Moreover, the time varying parameters and the fault-clear actions should be considered as well.

To overcome these challenges, we design a novel dataframe that encodes time-varying parameters, fault information and spatial-temporal relationships in transient dynamics in the next section.

### D. Spatial-Temporal Relationship in Transient Dynamics

We construct 3D tensors to encode the input trajectories such that the spatial-temporal relationships in the power system can be included. In addition, computation complexities of Fourier transforms are also reduced. The proposed framework is shown in Fig. 5(a).

Importantly, we aim to predict the trajectories under changing net injections $a$ and faults $u$. This is different from most previous works that learn a static mapping from input to output time sequences for fixed parameters. Therefore, we encode parameters and fault-clear actions explicitly in the input tensor.

### E. Encoding Time-Varying Parameters and Actions

Time-varying parameters include the net active power injection $p(t) = (P_1(t), \cdots, P_N(t))$ and reactive power $q(t) = (Q_1(t), \cdots, Q_N(t))$. We stack them on the $y$ axis on $y = 4$ and $y = 5$ as shown in blue part of Fig. 5(a). The benefit of this design is that the variance of $p(t)$ and $q(t)$ through time is naturally incorporated in the $z_{\text{in}}$ axis. Let $u_1(t)$ and $u_2(t)$ encodes the location of fault and the type of fault, respectively. The fault information $u_1(t)$ and $u_2(t)$ are stacked to the $y$-axis as $y = 6$ and $y = 7$, shown in red part of Fig. 5(a).

Fault-clear actions may happen in the predicted time horizon $[t_{\text{on}}, t_{\text{on}} + t_{\text{out}}]$. To incorporate future actions and temporal dependence in prediction time steps, we expand the 3D input tensor in Fig. 5(b) along the output time sequence, with the new axis $z_{\text{out}} = 1, \cdots, t_{\text{out}}$ correspond to the time stamp from $t_{\text{on}}$ to $t_{\text{on}} + t_{\text{out}}$. As shown in Fig. 5(b), the axis of $z_{\text{in}}$ is attached with $z_{\text{out}} + t_{\text{on}}, u_1(z_{\text{out}} + t_{\text{on}})$ and $u_2(z_{\text{out}} + t_{\text{on}})$, respectively. This way, fault-clearing actions and the output time stamps are encoded in the dataframe without adding extra complexity for batch operation. To allow better representation capability, the dimension of $z_{\text{in}}$ corresponds to each $z_{\text{out}}$ conduct a linear combination with weights of the shape $\mathbb{R}^{(\tau_{\text{in}}, +3) \times (\tau_{\text{in}}, +3)}$.

After such an encoder, the data frame is converted from a 3D tensor with dimension $\mathbb{R}^{N \times Y \times \tau_{\text{in}}}$ to a 4D tensor with dimension $\mathbb{R}^{3N \times N \times Y \times \tau_{\text{in}}}$ for the predicted dynamics of $\delta, \omega$ and $V$ along $t_{\text{out}}$ time steps for $N$ buses.

### F. 3D Fourier Transform

After the encoding, the Fourier transform in [9] is reduced to 3D Fourier transform along the axis of $x, y$ and $z_{\text{out}}$ as

$$\left(\mathcal{F}g_{j-1}\right)(\xi_1, \xi_2, \xi_3; z_{\text{in}}) = \sum_{x=0}^{N-1} \sum_{y=0}^{Y-1} \sum_{z_{\text{out}}=0}^{t_{\text{out}}-1} e^{-2\pi i \left(\frac{x\xi_1}{N} + \frac{y\xi_2}{Y} + \frac{z_{\text{out}}\xi_3}{t_{\text{out}}}\right)}$$

for $g_{j-1}(x, y, z_{\text{out}}; z_{\text{in}})$, where $\xi_1, \xi_2$ and $\xi_3$ are modes in the frequency domain in the three dimensions after the discrete Fourier transform.

The structure of Fourier layer with the 3D Fourier Transform is visualized in Fig. 5(c). After truncating the
Fourier series at a maximum number of modes $k_{\text{max},i}$ for the $i$-th dimension, an equivalent convolution in the frequency domain is conducted with weights $R_\phi \in \mathbb{R}^{2k_{\text{max},1} \times 2k_{\text{max},2} \times 2k_{\text{max},3} \times (\tau_{\text{in}} + 3) \times (\tau_{\text{in}} + 3)}$. The time domain signal is recovered by inverse Fourier transform as follows:

$$\mathcal{F}^{-1}(R_\phi \cdot \psi(\mathcal{F}g_{j-1})(\xi_1, \xi_2, \xi_3; z_{\text{in}})) = F^{-1}(R_\phi \cdot \psi(\mathcal{F}g_{j-1})(\xi_1, \xi_2, \xi_3; z_{\text{in}}))$$

(14)

The output tensor after each Fourier layer has the dimension $\mathbb{R}^{\tau_{\text{out}} \times N \times Y \times (\tau_{\text{in}} + 3)}$. The predicted state trajectory are obtained from the output of the last layer after a dense combination weight $W_\psi \in \mathbb{R}^{(\tau_{\text{in}} + 3) \times 1}$.

$$\hat{s}_{\text{out}}(x, y, z_{\text{out}}) = (W_\psi \cdot (\mathcal{K}(\phi)g_{l})(x, y, z_{\text{out}}; z_{\text{in}}))$$

(15)

where increased layer $l$ enable the structure to learn more complex dynamic patterns. In simulation, we found that four layers are sufficient.

V. CASE STUDY

In this section, we conduct several case studies to illustrate the effectiveness of the proposed method. First, a single-machine infinite bus system is used to show the benefit of learning in the frequency domain compared to the time domain. Then we validate the performance of the proposed approach on a realistic power grid by the case studies with the Northeastern Power Coordinating Council (NPCC) 48-machine, 140-bus test system as shown in Fig. 6 [24], [25].

A. Simulation Setting

We construct the Fourier Neural Operator with four layers and batch normalization. The maximum number of modes in the frequency domain is set to be $k_{\text{max},1} = 3$, $k_{\text{max},2} = 3$ and $k_{\text{max},3} = 6$. The episode number and the batch size are 4000 and 800, respectively. Weights of FNO are updated using Adam with learning rate initializes at 0.02 and decays...
every 100 steps with a base of 0.85. We use Pytorch and a single Nvidia Tesla P100 GPU with 16GB memory. The power system toolbox in MATLAB is used to generate dataset of power system transient dynamics with the full 6-order generator model, turbine-governing system and exciters [24].

B. A Single-Machine Infinite Bus System Example

To visualize and compare the performance of different prediction approaches, we show an illustrative example on a generator connected to an infinite bus, modeling a connection to a large grid that appears as a voltage source [26]. The proposed method using Fourier Neural Operator (FNO) is compared with Physics-Informed Neural Networks (PINN) and deep neural network (DNN). The parameters for PINN is the same as [18] where the case study is also a single-machine infinite bus system. The DNN have a dense structure and seven layers. For the prediction with 4.5 seconds, the average relative mse on the test set for FNO, DNN, PINN are 0.0098, 0.1811, and 15.53, respectively. The extreme large mse for PINN reflects that it fails when a long prediction horizon is needed.

Fig. 7 shows the dynamics of frequency deviation $\omega$ and angle $\delta$ in a prediction of 4.5 seconds for FNO and DNN. The ground truth is the trajectory found by numerical simulation. FNO fits the ground truth almost perfectly. By contrast, the learned dynamics from DNN are not smooth and show much larger deviations from the ground truth compared with FNO. This verifies the analysis illustrated in Fig. 4 that purely time-domain tend to overfit easily and have difficulties learning smooth oscillations.

![Fig. 7. Example trajectories on a single-machine infinite bus system. FNO almost exactly matches the true one, while DNN shows much larger errors.](image)

C. Performance on Larger Power Systems

The performance of the proposed method on a practical power system is verified by simulations on Northeastern Power Coordinating Council (NPCC) test system, which represents the power grid of the northeastern United States and Canada and was involved in the 2003 blackout event [25].

The input trajectory evolves $t_{in} = 20$ time steps, with time interval $\Delta t$ between neighbouring time step to be 0.03 (i.e., approximate two cycles that can be attained by most phasor measurement unit (PMU)). We predict the subsequent trajectory of the length $t_{out} = 150$ with total duration of 450s. Fig. 8 shows accurate and predicted trajectory of the system after a three-phase line fault between bus 54 and bus 103 cleared at the time of 0.42s. Let the time of fault happens as the time $t = 0$. The prediction starts at $t_{on} = 0.03s$, which means one time step in the input trajectory corresponds to the fault-on system. The grey area is the envelope of trajectories in all generator buses and the lines are the trajectories in ten generator buses. For all the three states variables (i.e., $\delta$, $\omega$ and $V$), the predicted trajectory in Fig. 8(b) has similar envelope as the accurate trajectory in Fig. 8(a). The convergence of the envelope in frequency deviation $\omega$ to zeros indicates that the system is stable after the fault and its clear action. Moreover, both the magnitude and the periodic oscillations in the ten generator buses are all captured by the prediction with FNO for the on-fault and post-fault period. As illustrated in Fig. 5 the type of fault and the fault-clear action at $t = 0.42s$ is encoded in the input tensor of FNO. Correspondingly, Fig. 8(b) predicts a step increase of voltage at $t = 0.42s$, which is the same as Fig. 8(a). Notably, the magnitude of voltage at the bus 54 and bus 101 is below 0.8 p.u. before $t = 1s$, making it exceeds the permissible ranges of 5% from nominal. This may cause low-voltage curtailment of the generators and thus require the attention from system operators. Therefore, the proposed prediction can provide sufficient information for identifying how danger the system is.

To illustrate the performance of the proposed method in predicting an unstable system, Fig. 9 shows accurate and predicted trajectory of the system after a line-to-line fault between bus 48 and bus 100 and recovered at the time of 0.42s. The accurate trajectory of $\omega$ in Fig. 9(a) shows that the fault leads to loss of synchronicity of generators at bus 47 and bus 48. This also result in large voltage drop at bus 47 and bus 48. The states of all the other buses are still stable and appear to be a colored block because of small magnitude compared with the two unstable cases. Correspondingly, Fig. 9(b) shows that the proposed method successfully predict the unstable trajectory of $\omega$ over 4Hz at bus 47 and bus 48. Therefore, both the location of unstable buses and the level of instability can be provided by the proposed method. In the next subsection, we verify in the test set that the proposed method can predict all the unstable system accurately shortly after a fault happens.

D. Quantifying the Performance on NPCC

As shown in Fig. 8 and Fig. 9, the dynamics of the power system transient states differ greatly with different fault types and system parameters. To quantify the performance of the proposed method in stochastic scenarios, we calculate the mean prediction error in the test set with 100 cases where initial states, location of fault, type of fault and fault-clearing time are randomly generated. Three metrics are included:

1. Relative mean squared error (mse): The ratio of sum of square of error to the sum of square of state variables in the test set, i.e., $\frac{||s_{out} - \hat{s}_{out}||^2}{||s_{out}||^2}$.

2. Type1-error: Percentage of unstable cases predicted to be stable. This is the most severe type of error and may cause blackouts of power systems. If the average value of absolute $\omega$ from $t = 4s$ to $t = 4.5s$ is larger than 0.5Hz, then the case is identified to be unstable.

3. Type2-error: Percentage of stable cases predicted to be unstable. This would lead to conservative action from system operators.
(a) Accurate trajectory (lines) and envelope (grey area) after a three-phase line fault between bus 54 and bus 103 and recovered at the time of 0.42s

(b) Predicted trajectory (lines) and envelope (grey area) after a three-phase line fault between bus 54 and bus 103 and recovered at the time of 0.42s

Fig. 8. Stable dynamics of angle \( \delta \) (left), frequency deviation \( \omega \) (middle) and voltage \( V \) (right) in NPCC corresponding to (a) accurate trajectory. (b) prediction of FNO. The grey area shows the envelope of the trajectory for all generator buses. Lines with different colors shows the trajectory in selected generator buses. The proposed method predict both the magnitude and osculations accurately.

(a) Accurate trajectory (lines) and envelope (grey area) after a line-to-line fault between bus 48 and bus 100 and recovered at the time of 0.42s

(b) Predicted trajectory (lines) and envelope (grey area) after a line-to-line fault between bus 48 and bus 100 and recovered at the time of 0.42s

Fig. 9. Unstable dynamics of angle \( \delta \) (left), frequency deviation \( \omega \) (middle) and voltage \( V \) (right) in NPCC corresponding to (a) accurate trajectory. (b) prediction of FNO. The grey area shows the envelope of the trajectory for all generator buses. Lines with different colors shows the trajectory in selected generator buses. The proposed method predict both the magnitude and oscillations accurately.

Notably, we fix the length of the input trajectory \( s_{in} \) to be \( \tau_{in} = 20 \). With the fault evolves through time, more data point after fault will be involved in \( s_{in} \) and this decreases the prediction error. Table I and Table II summarize the metrics for the prediction error corresponding to different number of cycles (one cycle is \( 1/60=0.017s \)) involved in the input trajectory \( s_{in} \). Importantly, fault-clear actions are set to be uniformly distributed in one to thirty cycles. Therefore, Table I generally cover the prediction of the dynamics of both the on-fault and post-fault process, while Table II covers more post-fault prediction with the increase of cycles after fault.

From Table II the relative mse of FNO corresponds to the three on-fault cycles is 0.0546, 0.0084, 0.0056, which is 23.3%, 87.9%, 91.6% lower than the case in DNN, respectively. Interestingly, DNN has the Type2-error to be approximate zero while extremely high Type1-error. The reason is that DNN will easily overfit to the majority stable cases that account for 93% in the training set. This results in the bad performance in identifying unstable cases that accounts for 7% in the training set. By contrast, FNO brings zero Type1-error once there is an on-fault data point entered in the input trajectory.
TABLE I
PERFORMANCE - ON FAULT

| Metric | Relative mse | Error-Type1 | Error-Type2 |
|--------|--------------|-------------|-------------|
|        | 0            | 2           | 4           |
|        | 0            | 2           | 4           |
| Cycle after fault | 0.0546 | 0.0084 | 0.0056 | 0.22 | 0 | 0 | 0.022 | 0.011 | 0.011 |
| FNO    | 0.0712       | 0.0696      | 0.0663      | 1    | 0.714 | 0.714 | 0 | 0 | 0.011 |
| DNN    |              |             |             |      |       |       |       |       |       |

TABLE II
PERFORMANCE - POST FAULT

| Metric | Relative mse | Error-Type1 | Error-Type2 |
|--------|--------------|-------------|-------------|
|        | 10           | 20          | 30          |
|        | 10           | 20          | 30          |
| Cycle after fault | 0.0035 | 0.0026 | 0.0016 | 0 | 0 | 0 | 0.011 | 0.011 | 0 |
| FNO    | 0.0710       | 0.0324      | 0.0193      | 0.429 | 0 | 0.143 | 0.022 | 0.022 | 0.011 |
| DNN    |              |             |             |       |       |       |       |       |       |

In Table II, the relative mse of FNO will decrease to lower than 0.0035 if more than 10 cycles after fault are involved in the input sequence $s_{in}$. For all the three cycles after fault, relative mse and Type1(2)-error of DNN is at least 90% higher than that of FNO. Notably, FNO bring zero Type1-error and 0.01 Type2-error. Therefore, the proposed prediction with FNO will capture all the dangerous unstable case and also mislabel stable cases with low probability. The low relative mse indicates that the proposed method can also simulate the dynamics of trajectory accurately.

TABLE III
AVERAGE COMPUTATIONAL TIME

| Methods                      | Prediction horizon = 3s | Prediction horizon = 4.5s | Prediction horizon = 6s |
|------------------------------|-------------------------|---------------------------|-------------------------|
|                              | FNO         | Matlab toolbox | Speed up | FNO         | Matlab toolbox | Speed up | FNO         | Matlab toolbox | Speed up |
| Prediction horizon = 3s      | 0.0036      | 1.69          | 469x      | 0.0037      | 2.24          | 605x      |
| Prediction horizon = 4.5s    | 0.0037      | 2.24          | 605x      | 0.0039      | 3.38          | 867x      |

Lastly, we compare the average computational time in the test set for FNO and Power System Toolbox in MATLAB as shown in Table III. For the prediction time horizon ranges from 3s, 4.5s, and 6s, the computational time of FNO is 0.0036s, 0.0037s, and 0.0039s, respectively. By contrast, the computational time of MATLAB toolbox is 1.69s, 2.24s, and 3.38s, which is 469, 605, and 867 times slower than FNO. Therefore, the proposed approach will significantly speed up the simulation for power system transient dynamics.

VI. CONCLUSION

This paper proposes a frequency domain approach for predicting power system dynamics and transients. Inspired by the intuition that there are relatively few dominate modes in the frequency domain, we construct neural networks with Fourier transform and filtering layers. We design the dataframe to encode the power system topology and fault-on-clear information in transient dynamics, allowing the extraction of spatial-temporal relationship through 3D Fourier transform. Simulation results show that the proposed approach speeds up prediction computations by orders of magnitude and is highly accurate for different fault types without the requirement on the detailed system parameters. Compared with state-of-the-art AI methods including Physics-Informed neural networks and generic deep neural network, the proposed method reduce MSE prediction error by more than 50% and vastly improves the detection of unstable dynamics.

REFERENCES

[1] T. T. Mai, P. Jadun, J. S. Logan, C. A. McMillan, M. Muratori, D. C. Steinberg, L. J. Vinmerstedt, B. Haley, R. Jones, and B. Nelson, “Electrification futures study: Scenarios of electric technology adoption and power consumption for the united states,” National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2018.

[2] M. Z. Jacobson and M. A. Delucchi, “Providing all global energy with wind, water, and solar power, part i: Technologies, energy resources, quantities and areas of infrastructure, and materials,” Energy policy, vol. 39, no. 3, pp. 1154–1169, 2011.

[3] M. A. Delucchi and M. Z. Jacobson, “Providing all global energy with wind, water, and solar power, part ii: Reliability, system and transmission costs, and policies,” Energy policy, vol. 39, no. 3, pp. 1170–1190, 2011.

[4] R. Golden and B. Paulos, “Curtailment of renewable energy in california and beyond,” The Electricity Journal, vol. 28, no. 6, pp. 36–50, 2015.

[5] P. S. Kundur, N. J. Balu, and M. G. Lauby, “Power system dynamics and stability,” Power System Stability and Control, vol. 3, 2017.

[6] P. W. Sauer, M. A. Pai, and J. H. Chow, Power system dynamics and stability: with synchrophasor measurement and power system toolbox. John Wiley & Sons, 2017.

[7] H.-D. Chang, C.-C. Chu, and G. Cauley, “Direct stability analysis of electric power systems using energy functions: theory, applications, and perspective,” Proceedings of the IEEE, vol. 83, no. 11, pp. 1497–1529, 1995.

[8] F. E. A. Team, “Free system disturbance and underfrequency load shedding event report,” FRCC Operating Committee, Tech. Rep., 2008.

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

[10] R. Huang, S. Jin, Y. Chen, R. Diao, B. Palmer, Q. Huang, and Z. Huang, “Faster than real-time dynamic simulation for large-size power system with detailed dynamic models using high-performance computing platform,” in 2017 IEEE Power & Energy Society General Meeting. IEEE, 2017, pp. 1–5.

[11] Y.-K. Wu, S.-M. Chang, and P. Mandal, “Grid-connected wind power plants: A survey on the integration requirements in modern grid codes,” IEEE Transactions on Industry Applications, vol. 55, no. 6, pp. 5584–5593, 2019.

[12] V. H. Chalishazar, S. Poudel, S. Hanif, and P. Thekumparambath Mana, “Power system resilience metrics augmentation for critical load prioritization,” Pacific Northwest National Lab.(PNNL), Richland, WA (United States), Tech. Rep., 2021.

[13] R. Yan, G. Geng, Q. Jiang, and Y. Li, “Fast transient stability batch assessment using cascaded convolutional neural networks,” IEEE Transactions on Power Systems, vol. 34, no. 4, pp. 2802–2813, 2019.

[14] L. Zhu and D. Hill, “Networked time series shapelet learning for power system transient stability assessment,” IEEE Transactions on Power Systems, 2021.
[15] B. Xia, H. Wu, Y. Qiu, B. Lou, and Y. Song, “A galerkin method-based polynomial approximation for parametric problems in power system transient analysis,” IEEE Transactions on Power Systems, vol. 34, no. 2, pp. 1620–1629, 2018.

[16] H. Yang, W. Zhang, F. Shi, J. Xie, and W. Ju, “Pmu-based model-free method for transient instability prediction and emergency generator-shedingd control,” International Journal of Electrical Power & Energy Systems, vol. 105, pp. 381–393, 2019.

[17] J. C. Cepeda, J. L. Rueda, D. G. Colomé, and D. E. Echeverría, “Real-time transient stability assessment based on centre-of-inertia estimation from phasor measurement unit records,” IET Generation, Transmission & Distribution, vol. 8, no. 8, pp. 1363–1376, 2014.

[18] J. Stiasny, S. Chevalier, and S. Chatzivasileiadis, “Learning without data: Physics-informed neural networks for fast time-domain simulation,” arXiv preprint arXiv:2106.15987, 2021.

[19] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, “Fourier neural operator for parametric partial differential equations,” arXiv preprint arXiv:2010.08895, 2020.

[20] C. Zhao, E. Mallada, S. Low, and J. Bialek, “A unified framework for frequency control and congestion management,” in 2016 Power Systems Computation Conference (PSCC). IEEE, 2016, pp. 1–7.

[21] P. W. Sauer, M. A. Pai, and J. H. Chow, Power system dynamics and stability: with synchrophasor measurement and power system toolbox. John Wiley & Sons, 2021.

[22] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” Journal of Computational Physics, vol. 378, pp. 686–707, 2019.

[23] S. L. Brunton and J. N. Kutz, Data-driven science and engineering: Machine learning, dynamical systems, and control. Cambridge University Press, 2019.

[24] J. H. Chow and K. W. Cheung, “A toolbox for power system dynamics and control engineering education and research,” IEEE transactions on Power Systems, vol. 7, no. 4, pp. 1559–1564, 1992.

[25] W. Ju, J. Qi, and K. Sun, “Simulation and analysis of cascading failures on an npcc power system test bed,” in 2015 IEEE Power & Energy Society General Meeting. IEEE, 2015, pp. 1–5.

[26] D. Ernst, M. Glavic, F. Capitanescu, and L. Wehenkel, “Reinforcement learning versus model predictive control: a comparison on a power system problem,” IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 39, no. 2, pp. 517–529, 2008.