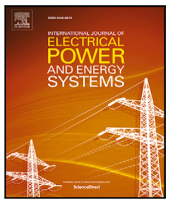
Electrical Power and Energy Systems 162 (2024) 110307

Contents lists available at ScienceDirect

International Journal of Electrical Power and Energy Systems

journal homepage: www.elsevier.com/locate/ijepes



Data-driven transient stability analysis using the Koopman operator

Amar Ramapuram Matavalam [a], Boya Hou [b,*], Hyungjin Choi [c], Subhonmesh Bose [b], Umesh Vaidya [d]

[a] School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ, 85287, USA
[b] Department of Electrical and Computer Engineering, Coordinated Sciences Laboratory, University of Illinois Urbana-Champaign, Urbana, IL 61801, USA
[c] Sandia National Laboratories, USA
[d] Department of Mechanical Engineering, Clemson University, Clemson, SC, 29634, USA

ARTICLE INFO

Keywords:
Power systems
Transient stability
Koopman operator
Reproducing kernel Hilbert space

ABSTRACT

We present data-driven methods for power system transient stability analysis using a unit eigenfunction of the Koopman operator. We show that the Koopman eigenfunction with unit eigenvalue can identify the region of attraction of the post-fault stable equilibrium. We then leverage this property to estimate the critical clearing time of a fault. We provide two data-driven methods to estimate said eigenfunction; the first method utilizes time averages over long trajectories, and the second method leverages nonparametric learning of system dynamics over reproducing kernel Hilbert spaces with short bursts of state propagation. Our methods do not require explicit knowledge of the power system model, but require a simulator that can propagate states through the power system dynamics. Numerical experiments on three power system examples demonstrate the efficacy of our method.

1. Introduction

Transient stability analysis of a power system seeks to answer whether grid dynamics will converge to a stable equilibrium point, following the clearing of a line fault or a generator failure. It is a mature field of research with extensive prior literature, e.g., see [1–3] among others. Time-domain simulation via numerical integration of power system models is perhaps the most widely used technique, as stated in [3]. Essentially, one simulates the post-fault dynamics starting from various points on the fault-on trajectory to certify whether the system will quickly converge to a stable equilibrium point, exhibit large oscillations, or even diverge. However, such a method suffers from several drawbacks when applied to large and complex modern power systems. To begin with, it requires an accurate model of the power system, which becomes more challenging with the integration of renewable energy and distributed power grid components. Consider for instance wind farms with interconnected wind turbines. The interaction of fluid dynamical flows itself presents a challenge in deriving an analytical model [4], let alone uncertainty in wind speed, direction, etc. Moreover, numerical integration of differential equations describing fault-on and post-fault systems can be computationally intensive, thus rendering it unsuitable for the stability assessment close to the time of power delivery which is required for power systems with a wide range of operating points.

The direct method is an alternative approach to assessing transient stability without explicitly integrating differential equations of post-fault systems. It utilizes Lyapunov-type functions to assess the stability of a post-fault system, based on an approximate stability boundary. Transient energy functions (TEF) expounded in [5–8] are examples of such Lyapunov-style functions that decrease along the system trajectory once the fault is cleared [9,10], a critical value of TEF (called the critical energy) is then computed, such that the sublevel sets can be certified to lie within the region of attraction (ROA) of the stable equilibrium point of the post-fault dynamics. In order to compute such critical energy, the controlling unstable equilibrium point (UEP), closest UEP [11–14] and the boundary of stability region based controlling unstable equilibrium point (BCU) method in [6,15] are widely established. Nevertheless, computing critical energy is not easy, as it requires finding several unstable equilibrium points surrounding the stable equilibrium point of post-fault dynamics and determining which UEP is the most relevant to the faulted system. The Potential Energy Boundary Surface (PEBS) method [16] avoids computing the relevant UEP by finding a local approximation of the stability boundary of the original system model using an analytically derived approximate potential energy function of the power system. However, beyond the single-machine infinite bus system, sufficient conditions

* Corresponding author.
E-mail addresses: amar.sagar@asu.edu (A.R. Matavalam), boyahou2@illinois.edu (B. Hou), hchoi@sandia.gov (H. Choi), boses@illinois.edu (S. Bose), uvaidya@clemson.edu (U. Vaidya).

https://doi.org/10.1016/j.ijepes.2024.110307
Received 24 March 2024; Received in revised form 29 August 2024; Accepted 9 October 2024
Available online 23 October 2024

for PEBS method to provides accurate stability assessments are not well-understood.

Another major limitation of the direct method is the choice of the Lyapunov function itself, which depends on the mathematical description of the power system model. These functions are only known for a limited class of simplified power system models. Authors in [17–20] have proposed to construct such functions for general power systems using polynomial optimization, following the seminal work of Parrilo in [10]. While promising in theory, state-of-the-art software for these methods does not scale favorably with the size of the model. In particular, they require explicit analytical models of various power system components. This poses a challenge since deriving accurate system equations that capture nuances of modern power system components ex-ante is not only tedious but may be impossible in some situations. For example, consider the emerging inverter-dominated power systems where the inverter/plant dynamics are not expressed as analytical equations, but are instead represented by compiled dynamically linked libraries in order to protect the intellectual property of inverter manufacturers. In other words, the underlying dynamics of an inverter are deliberately hidden from system operators in order to protect intellectual property, thus limiting the applicability of the direct method. Even if the mathematical models of these components were easily accessible, the exact representation of the model, whether in the phasor domain or a more detailed counterpart, can heavily influence stability assessments, e.g., see [21]. Furthermore, the aforementioned methods are all fault-dependent and focus on assessing a particular fault. Since the critical energy is different for each fault, the computational challenge compounds with multiple possible counterfactual fault scenarios.

To cope with some of the above challenges, we propose a data-driven transient stability assessment tool based on the Koopman operator. First developed in [22], the Koopman operator lifts the finite-dimensional nonlinear evolution of the state to a linear but infinite-dimensional evolution in the function space of observables (scalar mappings of the state). Approximation of the Koopman operator from data is well studied in the literature, see [23–28] for details. As a linear operator, its spectra reveal information relevant to stability monitoring [29–33]. In this paper, we show that eigenfunctions of the Koopman operators of the post-fault dynamics reveals its ROA. One advantage of using the Koopman operator for transient stability analysis is that these eigenfunctions can be learned from data of states propagated through the system dynamics, but do not require explicit analytical models. Prior literature such as [34–36] have investigated data-driven transient stability assessment by leveraging the learned Koopman operator. Nevertheless, accurate estimates of the Koopman operator and its leading eigenfunctions require a pre-selected set of observables to be rich enough. Given the challenges of selecting such a basis, we leverage the results in [37] to approximate the Koopman eigenfunction nonparametrically in a reproducing kernel Hilbert space (RKHS). As shown in [38], kernel methods demonstrate excellent empirical performance in data-limited regimes. Still, the scalability often suffers with larger datasets. To counter this difficulty, we allow selective loss in the representation of that operator to control model complexity.

The paper is organized as follows. In Section 2, we recast the problem of transient stability analysis into learning the ROA-indicator function, which in turn relies on approximate eigenfunctions of the Koopman operator associated with eigenvalue one. We then relate numerical integration used by commercial software to the computation of such an eigenfunction that is similar in spirit to a discretized version of a Fourier averaging introduced in [39]. Such a connection will likely make our exposition and algorithms more amenable to adoption as an add-on to existing software. Building on those insights, we then leverage this connection to approximate the ROA using a so-called "seeding function" in Section 3. Given the difficulty of selecting a proper seeding function, we next propose a non-parametric *kernel* method in Section 4. This exposition builds on well-studied properties

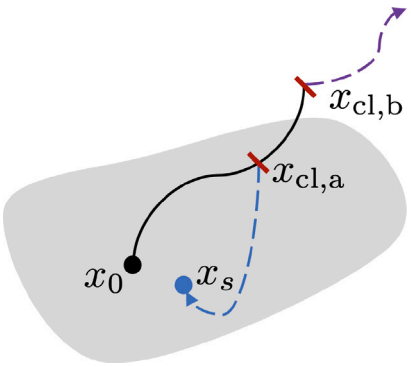


Fig. 1. ROA of post-fault stable equilibrium (gray). Starting from x_0, when the fault is cleared before the fault-on trajectory (black) exits ROA (e.g., at $x_{\mathrm{cl},a}$), the post-fault dynamics converges to x_s. In contrast, the purple trajectory plots post-fault behavior when the fault is cleared at $x_{\mathrm{cl},b}$ after the fault-on dynamics leaves the ROA.

of the interaction of the Koopman operator with RKHSs in [40], for which sample complexity has been characterized in [37]. In addition, to control the growth of model complexity, we propose to learn a *sparse* variant of the Koopman operator from data using the framework in [37]. We emphasize that thanks to the data-oriented nature of our proposed methods, our algorithm can be used in tandem with black-box simulators such as PSSE, PSLF, PowerWorld, DigSilent, etc., and do not rely on the mathematical structure of the power system model. With a learned post-fault dynamical system, our method allows one to analyze multiple fault-on trajectories efficiently. We demonstrate the efficacy of our data-driven methods using a single-machine infinite bus system, a 3-bus power system example, and the 39-bus New England example in Section 5. The experiments in this paper are meant to provide a proof of concept; our ultimate goal is to scale our methods to large power systems in future work. Concluding remarks and interesting research directions are outlined in Section 6.

Our key contributions are as follows: (a) we establish the connection between the unit Koopman eigenfunction and time-domain simulation, where the latter is widely adopted in industry. (b) In order to approximate the unit eigenfunction of the Koopman operator, we present a nonparametric algorithm to construct the Koopman operator associated with the post-fault dynamics. Since the learned Koopman operator serves as a model of the system dynamics, we further control model complexity via sparsification. (c) Leveraging the learned unit eigenfunction of the Koopman operator, we propose a data-driven algorithm to assess power system transient stability.

2. Transient stability analysis with Koopman operator

Transient stability analysis requires one to compute the critical clearing time (CCT) of a fault, that is, the time required for the fault-on trajectory to leave the region of attraction (ROA) of the stable equilibrium point of the post-fault power system dynamics. In this section, we relate the eigenfunction of the Koopman operator to the identification of said ROA.

To make the connection between CCT estimation and Koopman eigenfunctions, consider a power system for which the shaded region in Fig. 1 depicts the ROA of the post-fault system whose equilibrium point is x_s. Henceforth, we indicate this region by ROA(x_s). The trajectory (marked in black) starting from x_0 is the fault-on trajectory. If the fault is cleared at $x_{\mathrm{cl},a}$ before that trajectory leaves the ROA of the post-fault system, the system dynamics will naturally drive the state to x_s. If the fault is cleared at $x_{\mathrm{cl},b}$ outside of the shaded region, the system will diverge from x_s.[1]

[1] Such dynamics will typically converge to other equilibria but can oscillate wildly before it does so, which runs the risk of tripping relays or switches.

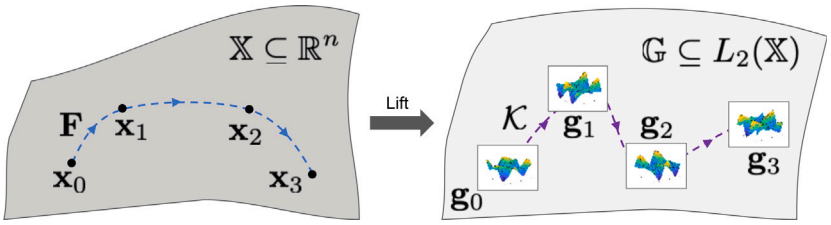


Fig. 2. Nonlinear evolution of state in the finite-dimensional state space is lifted to a linear evolution of observables in the infinite-dimensional space of all L_2-measurable functions of the state.

Consider the indicator function of ROA(x_s), defined as

$$\bar{\varphi}(x) = \begin{cases} 1, & \text{if } x \in \text{ROA}(x_s), \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

To calculate the CCT for a fault originating at x_0, one then needs to numerically integrate the on-fault trajectory (denoted $x_{\text{on-fault}}(t)$) and evaluate

$$\text{CCT} := \max\{t \geq 0 \mid \bar{\varphi}(x_{\text{on-fault}}(t)) = 1\}, \tag{2}$$

i.e., the time until which the evaluation of $\bar{\varphi}$ along the fault-on trajectory remains unity before falling off to zero.

We next relate $\bar{\varphi}$ to an eigenfunction of the Koopman operator. Consider a discrete-time deterministic non-linear dynamical system with states in $\mathbb{X}$, described by

$$x_{t+1} = F(x_t), \tag{3}$$

where $x_t \in \mathbb{X}$ and $F : \mathbb{X} \to \mathbb{X}$ is locally Lipschitz continuous and invertible. For power systems represented by differential–algebraic equations (DAE), F represents the numerical integration of the DAE with a fixed time-step.

Let $h : \mathbb{X} \to \mathbb{C}$ be an observable (i.e., a scalar-valued function) over $\mathbb{X}$. The Koopman operator $\mathcal{K}$ acts on h as

$$[\mathcal{K}h](x) = h(F(x)). \tag{4}$$

This operator thus maps the observable g to $\mathcal{K}g$, which when evaluated at x, is g evaluated at the state after being pushed forward through the system dynamics in one time-step as Fig. 2 reveals. This operator is infinite-dimensional as it acts on a suitably defined space of functions of the states. It is easy to verify that

$$\mathcal{K}(\alpha_1 g_1 + \alpha_2 g_2) = \alpha_1 \mathcal{K} g_1 + \alpha_2 \mathcal{K} g_2 \tag{5}$$

for scalars α_1, α_2 and observables g_1, g_2, and hence, is a linear operator. The spectrum of this operator is rich in information about the dynamics of the system. As we will demonstrate, the indicator function $\bar{\varphi}$ of ROA(x_s) is intimately related to the eigenfunctions of $\mathcal{K}$. Before we establish that relationship, we emphasize that the linear operator description of the dynamical system in the space of observables is distinct from a local linearization around a point in the state space.

Let $\varphi_\lambda(x)$ be an eigenfunction of $\mathcal{K}$ with eigenvalue λ, i.e., $[\mathcal{K}\varphi_\lambda](x) = \lambda\varphi_\lambda(x)$ for all $x \in \mathbb{X}$. In particular, if φ_1 is an eigenfunction with eigenvalue equal to 1,[2] then

$$[\mathcal{K}\varphi_1](x) = \varphi_1(F(x)) = \varphi_1(x). \tag{6}$$

Applying the Koopman operator n times, we get

$$\varphi_1(x) = [\mathcal{K}^n \varphi_1](x) = \varphi_1(F^{(n)}(x)). \tag{7}$$

For any $x \in \text{ROA}(x_s)$, we have $\lim_{n\to\infty} F^n(x) = x_s$, which combined with (6), gives

$$\varphi_1(x) = \varphi_1\left(\lim_{n\to\infty} F^n(x)\right) = \varphi_1(x_s). \tag{8}$$

In other words, φ_1 is constant over ROA(x_s). Said even more differently, φ_1 over ROA(x_s) is a scaled version of the indicator function $\bar{\varphi}$ of ROA(x_s).[3] As a result, one can utilize the eigenfunction of the Koopman operator whose eigenvalue is equal to one to construct the indicator function of ROA(x_s) with which one can then perform transient stability analysis. This analysis then reduces to the estimation of such an eigenfunction from data suitably obtained from system trajectories.

We further remark that the theory presented in this section also applies to systems specified by the algebraic states. This is due to the fact that the information of mapping between the algebraic states and the dynamic states is inherently embedded into the trajectory of the dynamic states [41]. In other words, only the dynamic states need to be sampled and analyzed in order to identify the ROA of post-fault dynamics and the CCT of a fault.

3. Connection between Koopman eigenfunctions and time-domain simulation

The ROA of a stable equilibrium point of the post-fault dynamics has long been estimated using repeated numerical integration of the post-fault dynamical system model, starting from a collection of points around the boundary of the ROA of an equilibrium point. Having studied the relationship between the eigenfunction of the Koopman operator with $\lambda = 1$ and the indicator function of the ROA, it is natural to surmise a connection between these eigenfunctions and the time-domain simulation approach to transient stability analysis. In this section, we precisely establish said connection. As shall become clear, such a relation sheds light on a computational scheme to estimate the eigenfunction with an eigenvalue equal to 1.

Consider a bounded function $\psi : \mathbb{X} \to \mathbb{C}$ on the state space with $|\psi| \leq B$. Then, define the N-step time-averaged version of ψ along a trajectory starting from x as

$$\Psi_N(x) := \frac{1}{N}\sum_{n=1}^{N} \psi(F^n(x)). \tag{9}$$

Then, for $x \in \mathbb{X}$, triangle inequality gives

$$\left|\Psi_N(F(x)) - \Psi_N(x)\right| = \frac{1}{N}\left|\psi(F^{N+1}(x)) - \psi(x)\right| \leq \frac{2B}{N}. \tag{10}$$

For large N, the above bound suggests that

$$[\mathcal{K}\Psi_N](x) = \Psi_N(F(x)) \approx \Psi_N(x). \tag{11}$$

This relation, together with (6), indicates that Ψ_N converges pointwise to the eigenfunction of the Koopman operator whose eigenvalue equals one. In the sequel, we refer to the eigenfunction with $\lambda = 1$ as the unit eigenfunction of the Koopman operator.

[2] The eigenfunction corresponding to eigenvalue one always exists when the system is measure-preserving. Indeed, the set of eigenvalues forms a lattice and the constant function $\varphi = 1$ is always a trivial eigenfunction corresponding to $\lambda = 1$.

[3] Similar conclusions can be drawn about stable orbits and other forward invariant subsets $\mathbb{I} \subset \mathbb{X}$ of the state space, characterized by $F(\mathbb{I}) = \{F(x) : x \in \mathbb{I}\} \subseteq \mathbb{I}$.

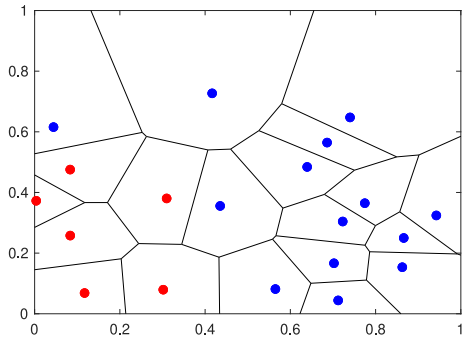


Fig. 3. Conceptual plot of the Voronoi cells for the various points in $\mathbb{D}$ of a 2-D system. The time-averaged value at these points for a well-chosen ψ should be binary that take different values inside and outside the ROA. The two different values are represented by blue and red colors.

The above calculation sheds light on a computational scheme to estimate a unit eigenfunction. Starting with a bounded function ψ, one can estimate Ψ_N by averaging ψ along a trajectory over N time-steps obtained via time-domain simulation.[4]. Then, this estimated unit eigenfunction should be roughly constant within ROA(x_s), using which one can then perform transient stability analysis. While this computational scheme does not require explicit evaluation of the Koopman operator, it reveals the link between the conventional time-domain simulation approach to the estimation of ROA(x_s) and the unit eigenfunction of the Koopman operator. Although Koopman operator learning has been applied to analyze power systems in [34,35], to the best of our knowledge, this paper is the first to draw such a connection between time-domain simulation and Koopman eigenfunctions for transient stability analysis.. We conclude this section with remarks on the practical implementation of estimating an eigenfunction via this approach.

It is time-consuming to estimate Ψ_N everywhere in the state space $\mathbb{X}$. Instead, one can probe Ψ_N at a finite number of points $\mathbb{D} := \{x_1, x_2, \dots, x_m\}$, sampled around the portion of the state space "near" the on-fault trajectory, and then interpolate Ψ_N at any point of interest by assigning the value of Ψ_N at its nearest-neighbor in $\mathbb{D}$. Points in $\mathbb{D}$ thus partition the state space $\mathbb{X}$ into separate regions, often referred to as *Voronoi cells*, as illustrated in Fig. 3, where Ψ_N takes two different values on $\mathbb{D}$. The interpolated value of Ψ_N (call it $\widehat{\Psi}_N$) is then defined by

$$\widehat{\Psi}_N(x) := \Psi_N\left(\arg\min_{x' \in \mathbb{D}} \|x - x'\|\right) \tag{12}$$

for a suitably defined norm $\|\cdot\|$. Fast open-source implementations for computing the nearest neighbor are readily available, e.g., see [42]. In the sequel, we call this approach the time-averaged approach.

Generally, the choice of ψ will affect the quality of ROA estimation using this approach. As our numerical experiments will suggest, a choice of ψ that is positive everywhere, equals one at x_s and that rapidly decays away from x_s tends to yield favorable results. N can be set based on time commitments of time-domain simulation of the post-fault dynamics and where $\Psi_N \approx \Psi_{N-1}$ for all points in $\mathbb{D}$. This time-averaging approach to computing a unit eigenfunction requires collecting post-fault trajectories over a long time horizon, which can be computationally intensive, and each trajectory is analyzed independently. The approach we present next, on the other hand, utilizes multiple one-step propagation of data points that are utilized together to obtain an interpolation.

[4] It was brought to our attention that this computational scheme is a special case of the discretized version of the Fourier average method introduced in [39].

4. Learning the indicator function via the Kernel method

One can approximate a unit eigenfunction of the Koopman operator in a suitable function space. In this section, we present an approximation technique to find a unit eigenfunction within a real-valued reproducing kernel Hilbert space (RKHS), in which the dataset consists of states propagated one time-step via the system dynamics.

We start by introducing the real-valued RKHS. Consider a symmetric, bivariate, positive definite, continuous, and bounded *kernel* function $\kappa : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ that satisfies the following *reproducing property* for $x \in \mathbb{X}$. Then, define $\mathcal{H}$ to be Hilbert space of functions, given by the completion of the span of $\{\kappa(x, \cdot) : x \in \mathbb{X}\}$. In this space, the kernel satisfies the following *reproducing property*,

$$\langle f, \kappa(\cdot, x)\rangle = f(x), \ \forall f \in \mathcal{H}, \tag{13}$$

where $\langle \cdot, \cdot \rangle$ computes the inner product in $\mathcal{H}$. Associated with kernel κ its feature map

$$\nu_\kappa(x) = \kappa(x, \cdot) \in \mathcal{H}. \tag{14}$$

To learn Koopman eigenfunctions in $\mathcal{H}$, one needs to understand how F interacts with functions $f \in \mathcal{H}$, where F describes the system evolution in (3). Assume that $\mathcal{H}$ is *closed* under the system dynamics, i.e.,

$$f \in \mathcal{H} \implies \mathcal{K}f = f \circ F \in \mathcal{H}. \tag{15}$$

The assumption that $\mathcal{H}$ is closed under system dynamics is common in approximation of linear transfer operators, e.g., see [43,44], It holds when the Koopman operator $\mathcal{K}$ is a Hilbert Schmidt operator mapping from $\mathcal{H}$ to $\mathcal{H}$. Under this assumption, the Koopman operator admits a closed-form expression using (cross)-covariance operators, per [37,40], and it can be learned from a collection of data points in $\mathbb{X}$ and their one-step propagation through the system dynamics F. In what follows, we introduce this expression of the operator and present a way to learn a *sparse* variant of this operator with data from which we then compute a unit eigenfunction for transient stability analysis. We also note that recent works in [45,46] hold promise to relax the closeness assumption by considering the Koopman operator as a Hilbert–Schmidt operator mapping from $\mathcal{H}$ to the space of square-integrable functions.

The expression for the Koopman operator interacting with an RKHS given in [37,40] relies on a *stochastic* dynamical system model rather than the deterministic model we introduced in (3). In particular, consider a stochastic dynamical system characterized by a transition kernel $p_F(x_{t+1}|x_t)$ that maps the state x_t at time t to a probability density of states at time $t+1$. The deterministic variant defined by F can be understood as a stochastic kernel that puts all its mass at $F(x_t)$ at time $t+1$, given the state x_t at time t. With this notation, suppose that $\mathbb{P}(X, X^+)$ denotes a joint distribution over $\mathbb{X} \times \mathbb{X}$, where X is sampled according to a sampling distribution and X^+ is the induced states after X is propagated through the system dynamics p_F. Covariance operators can then be defined as

$$\begin{aligned} C_{XX} &:= \mathbb{E}_X[\nu_\kappa(X) \otimes \nu_\kappa(X)], \\ C_{XX^+} &:= \mathbb{E}_{(X,X^+)}[\nu_\kappa(X) \otimes \nu_\kappa(X^+)], \end{aligned} \tag{16}$$

which are elements in the tensor product space $\mathcal{H}_\otimes := \mathcal{H} \otimes \mathcal{H}$, where $\mathbb{E}_{(X,X^+)}$ is the expectation with respect to $\mathbb{P}(X, X^+)$. Under a closedness assumption similar to (15)–precisely that $\mathbb{E}_{(X,X^+)}[f(X^+)|X] \in \mathcal{H}$ for all $f \in \mathcal{H}$–it follows from [47, Theorem 4] that one can define the Koopman operator using the aforementioned covariance operators as

$$\mathcal{K} := C_{XX}^\dagger C_{XX^+}. \tag{17}$$

See [26] for details. The prevalent definition of Koopman operator interacting with an RKHS is via $\mathcal{K} := C_{XX}^\dagger C_{XY}$. However, as pointed out by [48], $C_{XX}^\dagger$ is not globally defined if $\mathcal{H}$ is infinite-dimensional. As such, we consider its regularized variant,

$$\mathcal{K}_\varepsilon := \left(C_{XX} + \varepsilon I\right)^{-1} C_{XX^+}, \tag{18}$$

where I is the identity operator. Owing to the regularization parameter $\epsilon > 0$, the regularized covariance operator is positive definite and thus admits an inverse.

One can construct the operator $\mathcal{K}_\epsilon$ from data as follows. Consider a data-set $\mathbb{M} := \{(x_1, x_1^+), \ldots, (x_m, x_m^+)\}$ of m samples i.i.d. from $\mathbb{P}(X, X^+)$. Then, the empirical estimates of C_{XX} and C_{XX^+} can be computed via sample average as

$$\widetilde{C}_{XX} = \frac{1}{m}\sum_{i=1}^{m} \nu_\kappa(x_i) \otimes \nu_\kappa(x_i), \qquad \widetilde{C}_{XX^+} = \frac{1}{m}\sum_{i=1}^{m} \nu_\kappa(x_i) \otimes \nu_\kappa(x_i^+). \tag{19}$$

While the approximation accuracy improves with more data, data storage and computational issues compound when m grows large. To circumvent this burden, we prune $\mathbb{M}$ to construct a sparse dictionary $\mathbb{M}_\gamma$ based on the notion of *coherency* introduced in [49]. Specifically, we construct $\mathbb{M}_\gamma$ by identifying a subset of $\mathbb{M}$ that satisfies

$$\frac{\left|\kappa_\otimes\left((x_i, x_i^+), (x_j, x_j^+)\right)\right|}{\sqrt{\kappa_\otimes\left((x_i, x_i^+), (x_i, x_i^+)\right)\kappa_\otimes\left((x_j, x_j^+), (x_j, x_j^+)\right)}} \le \gamma, \tag{20}$$

for each i, j such that $(x_i, x_i^+), (x_j, x_j^+)$ are in $\mathbb{M}_\gamma$. This condition ensures that points that are sufficiently "close" to each other in $\mathbb{M}$ are discarded as they do not add to the richness of the representation of the operators in question. Here, $\kappa_\otimes$ is defined as $\kappa_\otimes\left((x_1, x_1^+), (x_2, x_2^+)\right) = \kappa(x_1, x_2)\ \kappa(x_1^+, x_2^+)$. Let $\mathcal{I}_\gamma$ be the indices among $1, \ldots, m$ for which (x_i, x_i^+) are in $\mathbb{M}_\gamma$ and define the feature matrices V_X, V_{X^+} whose ith columns are feature map ν_κ centered at x_i, x_i^+, for $i \in \mathcal{I}_\gamma$. One can then construct the sparse estimator of C_{XX^+} (C_{XX}) as

$$\hat{C}_{XX^+} = V_X A_\alpha V_{X^+}^\top, \quad \hat{C}_{XX} = V_X A_\beta V_X^\top, \tag{21}$$

where A_α (and similarly, A_β) is a diagonal matrix whose entries on the main diagonal minimize[5]

$$\left\| \frac{1}{m}\sum_{i=1}^{m} \nu_\kappa(x_i^+) \otimes \nu_\kappa(x_i) - \sum_{i\in\mathcal{I}_\gamma} \alpha_i \nu_\kappa(x_i^+) \otimes \nu_\kappa(x_i) \right\|_{\mathcal{H}_\otimes}^2 \tag{22}$$

over $\alpha \in \mathbb{R}^{|\mathcal{I}_\gamma|}$. The *sparse kernel Koopman estimator* can then be defined as

$$\hat{\mathcal{K}}_\epsilon := \left(\hat{C}_{XX} + \epsilon I\right)^{-1} \hat{C}_{XX^+}, \tag{23}$$

per [37]. When underlying system is ergodic and the stationary distribution is absolutely continuous with respect to the Lebesgue measure on $\mathbb{X}$, [37] further studies finite-sample convergence behavior of such a sparse estimator.

We next compute the eigenfunction of $\hat{\mathcal{K}}_\epsilon$ whose eigenvalue equals 1. Define the Gram matrices

$$G_{XX} = V_X^\top V_X, \quad G_{X^+X} = V_{X^+}^\top V_X, \tag{24}$$

and construct

$$Y := \left(A_\beta G_{XX} + \epsilon I\right)^{-1} A_\alpha G_{X^+X}. \tag{25}$$

From [40, Proposition 3.1], an operator of the form $\hat{\mathcal{K}}_\epsilon = V_X Y V_{X_+}^\top$ has an eigenvalue equals to one with the corresponding eigenfunction

$$\varphi_1(x) = k(x)w, \quad [k(x)]_i = \kappa(x_i, x), \quad i \in \mathcal{I}_\gamma, \tag{26}$$

if and only if w is a right eigenvector of Y associated with the same eigenvalue. This observation enables us to construct eigenfunctions of $\hat{\mathcal{K}}_\epsilon$ analytically from finite-dimensional Gram matrices G_{XX} and G_{X^+X}.

[5] In practice, one can also choose uniform weights, i.e., $\alpha = \beta = \mathbf{1}/|\mathcal{I}_\gamma|$, to speed up computation, see [37] for details.

However, the unit eigenfunction (26) is not perfectly constant inside the ROA of the stable equilibrium as it is an approximation derived from a finite representation of the Koopman operator instead of the ideal unit eigenfunction. We thus estimate an indicator function from $\varphi_1(x)$ via

$$\bar{\varphi}(x) = \begin{cases} 1, & \text{if } |\varphi_1(x) - \varphi_1(x_s)| \le \epsilon_\circ \\ 0, & \text{otherwise,} \end{cases} \tag{27}$$

where $\epsilon_\circ$ defines a user-specified tolerance. We remark that alternatively, one can also apply clustering techniques to $\varphi_1(x)$ where the number of clusters is two. The RKHS-based approximation to a unit eigenfunction is summarized in Algorithm 1.

Algorithm 1 Approximating Indicator Function in RKHS

Require: kernel κ; sample pairs $\mathbb{M}$; coherence parameter γ
1: Prune $\mathbb{M}$ to get $\mathbb{M}_\gamma$ that satisfies (20)
2: Solve for coefficients α, β as in (22)
3: Construct Y according to (25)
4: Find the right unit eigenvector $\boldsymbol{v}$ of Y and compute unit eigenfunction $\varphi_1(x)$ using (26)
5: Compute $\bar{\varphi}(x)$ via (27)
6: **return** $\bar{\varphi}(x)$.

5. Numerical experiments

In this section, we present numerical results for data-driven transient stability assessment on three power system examples: (1) a two-state single-machine infinite-bus example simulated in Python as an ordinary differential equation (ODE); (2) a four-state two-machine infinite-bus example simulated in Python as an ODE; and (3) the ten-machine New-England system simulated with constant power loads in PSSE. Both the kernel method and the time-averaged method were implemented in Python to estimate the CCT in each test fault scenario. In the description, we use the notation $j = \sqrt{-1}$. We emphasize that our goal in this section is to reveal the efficacy of RKHS-based Koopman approximation and the time-averaged estimation of the unit eigenfunction to perform transient stability analysis. A more comprehensive and scalable implementation of large power system examples is left for future endeavors.

5.1. The Single-Machine Infinite-Bus (SMIB) system

Consider a per unit (pu) system whose one-line diagram is given in Fig. 4(a). Its electromechanical dynamics is described by

$$\dot{\delta} = \omega, \quad \dot{\omega} = -D\omega + P_m - P_e \sin(\delta), \tag{28}$$

with damping constant $D = 1.3\ \text{rad}^{-1}\text{s}$, mechanical power input $P_m = 5$ pu, and electrical power coefficient $P_e = 10$ pu which corresponds to the maximum electrical power that can be transferred from this generator. The stable equilibrium point of this system is $[0.53\ \text{rad}, 0]$. Thus, the system under consideration is measure-preserving with respect to the Dirac delta measure at the equilibrium point. We consider the case where a transmission line fault happens and subsequently, P_e deviates from its value under normal operating conditions. In the most extreme case, a bolted fault corresponds to $P_e = 0$. While the value of P_e is determined by the fault location and impedance, for this experiment, we choose a set of values of P_e. The smaller the value of P_e, the lesser the power deliverable to the infinite bus, indicating a more severe fault condition. In our study, we study three fault-on scenarios with $P_e \in \{0.95, 0.75, 0.45\}$ pu.

In order to construct the indicator function using RKHS, we sampled 1500 initial points $x = [\delta, \omega]$ that are uniformly distributed over $[\delta, \omega] \in [-4, 4] \times [-6, 6]$. We then numerically integrated the ODE starting from each x for a time interval of 0.1 s to obtain x^+. For this 2-D system, we first scaled ω by 1/2 and then used a Gaussian kernel $\kappa(x_1, x_2) = \exp(-\|x_1 - x_2\|_2^2 / 2\times 0.5^2)$. The indicator function was then derived from

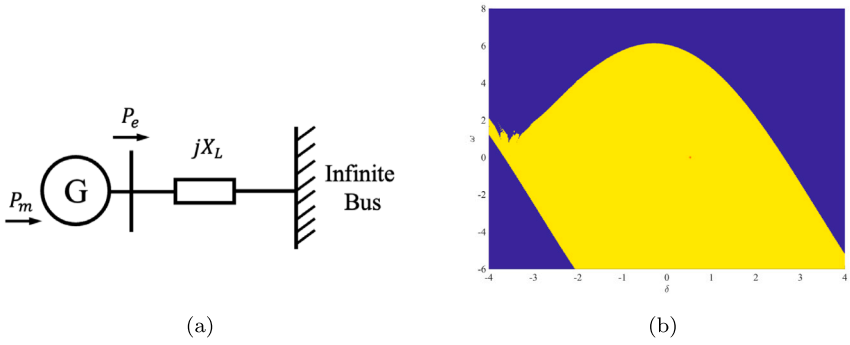


Fig. 4. (a) The SMIB system consists of a single generator connected to an infinite bus through a transmission line. A fault occurs in the transmission line and the electrical power P_e deviates from its normal operating value. (b) A plot of the estimated indicator function which reveals the ROA.

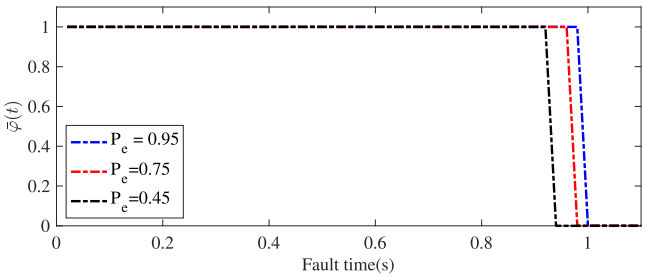


Fig. 5. Value of the indicator function $\bar{\varphi}$ evaluated at fault-on trajectories with $P_e = 0.95,\ 0.75,$ and 0.45.

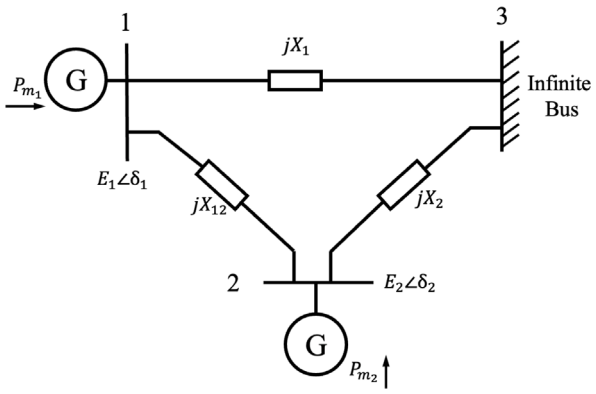


Fig. 6. The TMIB system consists of two generators at buses 1 and 2 connected to the infinite bus at bus 3. The generator malfunctions and the voltages E_1, E_2 deviate from their normal operating values.

φ_1 using (27) with $\epsilon_\circ = 8 \times 10^{-4}$. For the time-averaged indicator function, the same 1500 initial points were simulated for 20 s with a time-step of 0.05 s. The seeding function was the radial basis function (RBF) centered at $[0.53\ \text{rad}, 0]$ with a standard deviation of 0.25 along δ and a standard deviation of 0.5 along ω. $\bar{\varphi}$ was then obtained by the nearest neighbor algorithm in scikit-learn [42]. The resulting $\bar{\varphi}$ is plotted in Fig. 4(b).

To compute the CCTs, we evaluated the indicator function $\bar{\varphi}$ along each fault-on trajectory for $P_e \in \{0.95, 0.75, 0.45\}$, and the results are plotted in Fig. 5. As shown in the figure, the value of $\bar{\varphi}$ remains unity until the fault-on trajectory passes through the boundary of the ROA and it drops to zero. We use the time-stamps of these transitions as estimates of the CCTs that equal 0.98 s, 0.96 s, and 0.92 s, respectively, for the aforementioned choices of P_e. The CCTs from the time-averaging technique matched those obtained via the RKHS, and they both matched the values we obtained from time-domain simulations of the post-fault trajectories starting from various points of the fault-on trajectories.

5.2. The Two-Machine Infinite-Bus (TMIB) system

Consider the 4-state system in Fig. 6 described by

$$\begin{aligned}
\dot{\delta_1} &= \omega_1,\\
M_1\dot{\omega_1} &= P_{m1} - D_1\omega_1 - \frac{E_1}{X_1}\sin(\delta_1) \quad - \frac{E_1E_2}{X_{12}}\sin(\delta_1 - \delta_2),\\
\dot{\delta_2} &= \omega_2,\\
M_2\dot{\omega_2} &= P_{m2} - D_2\omega_2 - \frac{E_2}{X_2}\sin(\delta_2) \quad - \frac{E_1E_2}{X_{12}}\sin(\delta_2 - \delta_1),
\end{aligned} \tag{29}$$

with inertia constant $M_1 = M_2 = 6/(2\pi 60)\ \text{rad}^{-1}\text{s}^2$, damping constant $D_1 = D_2 = 2/2\pi 60\ \text{rad}^{-1}\text{s}$, mechanical power input $P_{m1} = 1.6$, $P_{m2} = 1.4$ pu, line reactances $X_1 = 0.6$, $X_2 = 0.2$, and $X_{12} = 0.2$ pu. The generator terminal voltages are $E_1 = 1$ pu and $E_2 = 1$ pu during normal operating conditions. We consider internal generator faults due to which E_1 and E_2 vary between 0 and 1. We further assume that the pre-fault and the post-fault systems are the same.

To construct $\mathbb{D}$, we densely sampled regions that are more likely to contain fault-on trajectories. Specifically, we sampled points along a few fault-on trajectories with different values of $0 \le E_1, E_2 \le 1$ via odeint() in Python. For this example, $\mathbb{D}$ consists of 9000 points from 900 fault-on trajectories with a sampling interval of 0.01 s.

In order to approximate the indicator function, we implemented the time-averaged approach in Section 3 and the kernel method in Section 4. To execute the time-averaged method, we used the same seeding function as we used for the SMIB system. We then evaluated Ψ at points in $\mathbb{D}$ via numerical simulations in Python, and the time-averaged value was computed with a step size of 0.05 s for a time horizon of 20 s. $\bar{\varphi}$ was then obtained by the nearest neighbor algorithm in scikit-learn [42]. For the kernel method, we rearranged $\mathbb{D}$ into sample pairs and scaled them via StandardScaler() in Python. We utilized a Gaussian kernel $\kappa(x_1, x_2) = 0.71\exp(-\|x_1 - x_2\|_2^2/2 \times 0.42^2) + 0.29\exp(-\|x_1 - x_2\|_2^2/2 \times 0.75^2)$. The indicator function was constructed via Algorithm 1 with $\gamma = 0.7$.

We next determined the CCT of the faulted scenario where $E_1 = E_2 = 0.4$. As illustrated in Fig. 7, the CCT of a fault was estimated as the instant when the $\bar{\varphi}(\cdot)$ drops to zero along the fault-on trajectory. We observed that the kernel method and the time-averaged method generate estimates of 0.29 s and 0.28 s, respectively. To find the actual critical clearing time, time-domain simulations were performed in Python for the same fault-on system for varying fault times followed by the post-fault system. The evolution of ω for the fault clearing times of 0.285 s and 0.29 s are plotted in Fig. 9. The figures indicate that the CCT is closer to 0.29 s, close to the CCTs estimated via the methods we presented.

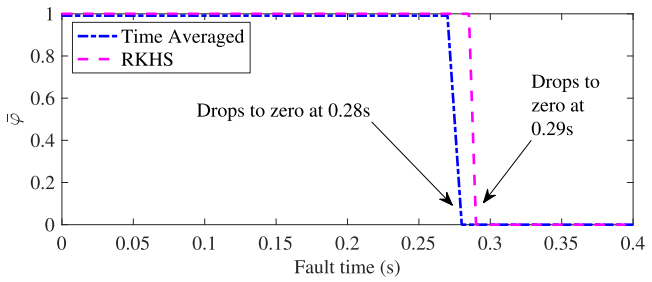


Fig. 7. The value of the indicator function evaluated along the TMIB fault-on trajectory with $E_1 = E_2 = 0.4$. The CCT is estimated to be 0.29 s via the kernel approach and 0.28 s via the time-averaged approach.

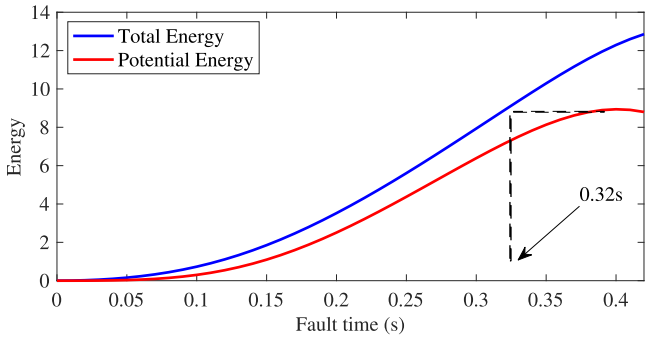


Fig. 8. The total energy and the potential energy of the TMIB faulted system with $E_1 = E_2 = 0.4$. The CCT is estimated to be 0.32 s.

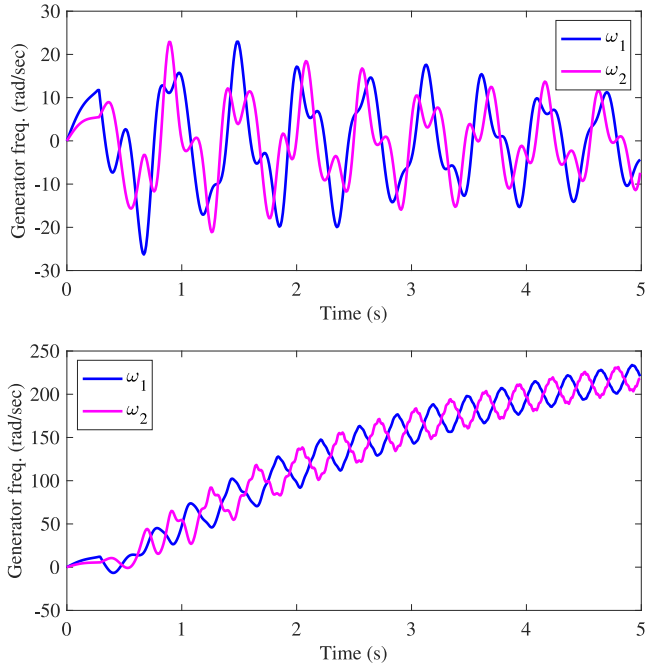


Fig. 9. Evolution of generator frequencies in the TMIB system with a fault $E_1 = E_2 = 0.4$ applied at $t = 0$ s and (a) cleared at $t = 0.285$ s, (b) cleared at $t = 0.29$ s.

We compared the proposed methods to the standard potential energy boundary surface (PEBS) method in [41]. Different from our data-driven approach, the PEBS method requires an analytic expression of the dynamical system in order to derive a potential energy function. Fig. 8 plots the total energy together with the total potential energy of the faulted system with $E_1 = E_2 = 0.4$. The CCT is estimated to be

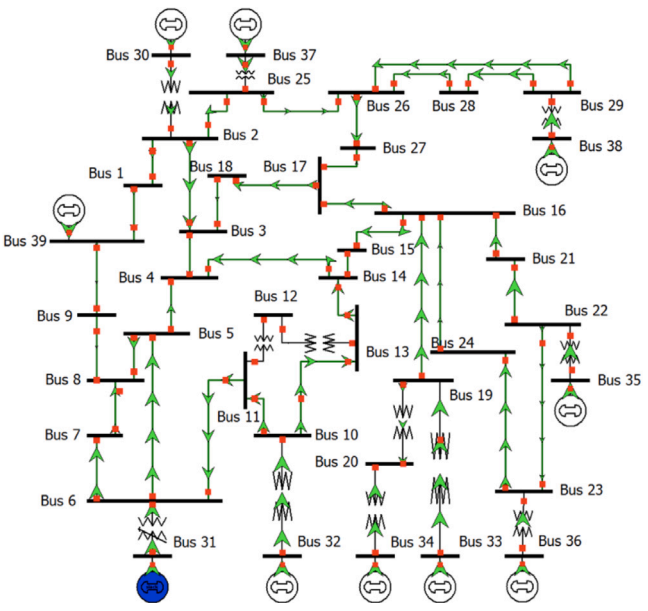


Fig. 10. The ten-machine New-England Power System (also known as the IEEE 39 bus system) consists of 10 generators and 46 lines whose parameters are specified in [50]. *Source:* This depiction is taken from [51].

0.32 s, which is the time when the total energy of the system exceeds the maximum potential energy. These observations suggest that the results from our data-driven approaches better align with the CCT for this system, compared to the PEBS method.

5.3. The ten-machine 39-Bus New-England system

We now apply our data-driven methods to the ten-machine New-England system in Fig. 10, where the machines are represented by the classical generator model, and the loads are constant power types. Table 2 lists the faults we consider. Assume that the pre- and post-fault systems are the same. We simulated the system in Siemens PTI PSSE, and the actual value of CCT was obtained through repeated time-domain simulation implemented in PSSE.

To implement the time-averaging approach,

we generated $\mathbb{D}$ by sampling and perturbing points at all 39 buses along five fault-on trajectories with fault admittances being $[10^2 - 10^2 j, 10^3 - 10^3 j, 10^4 - 10^4 j, 10^5 - 10^5 j, 10^6 - 10^6 j]$ pu. A total of 150 points along each trajectory are collected and perturbed before adding them to $\mathbb{D}$. Thus, the overall number of points in $\mathbb{D}$ is $39 \times 5 \times 150 = 29{,}250$. PSSE was then used to simulate the post-fault system from each point in $\mathbb{D}$ with a time horizon of 20 s and a step size of 0.01 s. We used the same seeding function utilized for prior examples. Then, we estimated $\bar{\varphi}$ using the nearest neighbor function from the scikit-learn package in Python.

To implement the kernel method, we sub-sampled 5000 trajectories of length 0.1 s out of the whole dataset used in the time-averaged approach. We then scaled the samples via StandardScaler() in Python and used the Gaussian kernel $\kappa(x_1, x_2) = 0.7\exp(-\|x_1 - x_2\|_2^2/2 \times 0.45^2) + 0.2\exp(-\|x_1 - x_2\|_2^2/2 \times 0.7^2) + 0.1\exp(-\|x_1 - x_2\|_2^2/2 \times 2^2)$. We calculated $\bar{\varphi}$ via Algorithm 1 with $\gamma = 0.8$ and regularization parameter $\epsilon = 1e^{-13} \times |\mathbb{M}_\gamma|^{-0.2}$.

Fig. 11 and Fig. 12 plot the values of the indicator function evaluated along the fault-on trajectories listed in Table 2. The estimated CCTs are the instants when the indicator function evaluations drop to zero along the trajectories. We compared the estimated CCTs with that obtained from time-domain simulation in Table 2. For illustration, we plotted the generator rotor frequency ω for different fault clearing times

Table 1
Computation times in the different phases for the NE system.

Phase	Time-averaged	RKHS
Data Generation	548 s	23 s
ROA indicator function construction	9 s	54 s
CCT Estimation	0.09 s	0.12 s

Table 2
Fault information of sample cases studied for the NE system.

Case	A	B	C
Fault $Y(\times 10^5)$	$(5-4j)$	$(1-4j)$	$(1-4j)$
Faulted Bus	33	6	24
True CCT	0.29 s	0.32 s	0.51 s
CCT (time-average)	0.29 s	0.33 s	0.49 s
CCT (kernel)	0.28 s	0.35 s	0.54 s

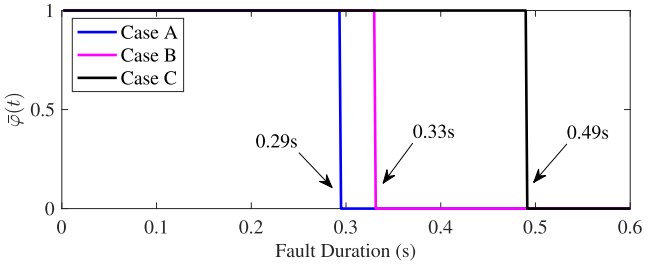


Fig. 11. The value of the time-averaged indicator function evaluated along fault-on trajectories for faults listed in Table 2.

in Fig. 13, which demonstrate the converging/diverging behavior of ω when the fault duration is lesser/greater than the CCTs obtained from the time-domain simulations. Our estimated CCTs are close (within 0.03 s) to that obtained from time-domain simulations for all faults. The methods described are executed on a server equipped with a 3-GHz Intel Xeon 32-core processor. The computation process involves three distinct phases: (a) simulating trajectories using PSSE, (b) analyzing trajectory data to derive an indicator function for the ROA, and (c) using this indicator function to estimate the CCT. The ROA indicator function remains applicable across various faults and perturbations as long as the post-fault system dynamics remain consistent. Therefore, while reducing the time spent on the initial two phases is desirable, the ability to quickly evaluate the ROA indicator function at any point within the state space is crucial.

Table 1 summarizes the computational expenses associated with both methods on the 39-bus NE system. The time-averaged method necessitates a substantial number of trajectories compared to the RKHS method to accurately capture the ROA, resulting in an extended data-generation period. Conversely, the RKHS method involves more complexity in the data-analytics phase, as the dictionary needs to be built from the data and the Koopman operator needs to be constructed before the unit eigenfunction, corresponding to the ROA indicator function, can be established. Ultimately, the indicator functions generated by both methods can be evaluated with minimal overhead (≈ 0.1 s) during fault simulations to identify the CCT, thus demonstrating their use for power system planners and operators (see Fig. 12).

6. Conclusions

In this paper, we developed data-driven methods to analyze transient stability in power systems that do not require analytical equations of system dynamics. We showed that the unit eigenfunction of the Koopman operator can identify the ROA of a stable equilibrium point of the post-fault system. In particular, we establish the connection between the unit Koopman eigenfunction and numerical integration, where the latter is widely adopted in industry. We then leveraged this

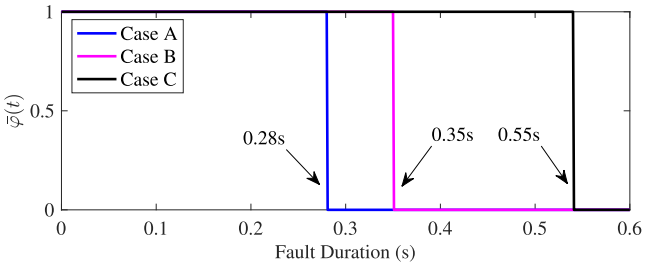


Fig. 12. The value of the kernel-based indicator function evaluated along fault-on trajectories for faults listed in Table 2.

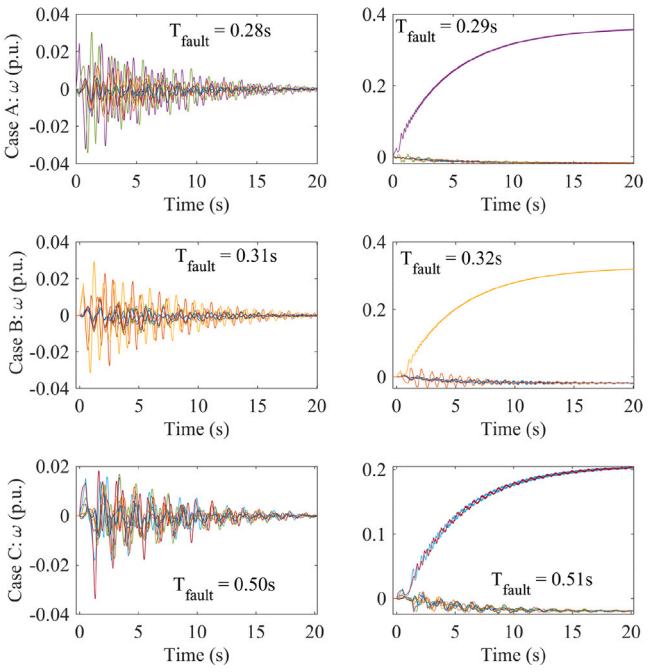


Fig. 13. The generator ω trajectories for fault scenarios listed in Table 2 with fault durations lesser and greater than the corresponding CCT.

property to estimate the critical clearing time of a fault. To this end, we proposed the kernel method and the time-averaged method which estimate the unit function from simulation data. Both approaches eliminate the need for explicit modeling of the system dynamics. Instead, they only require access to a black-box simulator. This property is useful for the emerging power system in which most renewable resources are modeled as input–output black-box models and almost no information about the underlying component dynamics is available. Finally, we validated the efficacy of our approach on three power system examples.

The time-averaging approach requires longer sampling trajectories, but the computation of the unit eigenfunction is typically computationally light. The kernel method on the other hand requires short snapshot bursts, but the computation of the eigenfunction is more involved compared to the time-averaging approach. The compression mechanism serves to lighten the computational burden.

There are two important directions for future work. First, we want to scale our techniques to larger power system examples and possibly automate the process of choosing kernel and seeding functions. Second, we want to extend our methods to study transient stability with a collection of inverter-based resources whose timescales for evolution may be significantly faster than the mechanical timescales of the transmission network.

CRediT authorship contribution statement

Amar Ramapuram Matavalam: Writing – original draft, Validation, Methodology, Conceptualization. **Boya Hou:** Writing – original draft, Methodology, Formal analysis. **Hyungjin Choi:** Conceptualization. **Subhonmesh Bose:** Writing – review & editing, Funding acquisition, Conceptualization. **Umesh Vaidya:** Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Subhonmesh Bose reports financial support was provided by National Science Foundation. Amar Ramapuram Matavalam reports financial support was provided by Power Systems Engineering Research Center. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partly supported by the grants NSF EPCN-2031570, USA and PSERC S-106, USA.

Data availability

Data will be made available on request.

References

[1] Varaiya P, Wu FF, Chen R-L. Direct methods for transient stability analysis of power systems: Recent results. Proc IEEE 1985;73(12):1703–15.
[2] Pai MA. Survey of practical direct methods of stability analysis in power systems. Electr Mach Power Syst. 1984;9(2–3):131–43.
[3] Sauer PW, Pai MA, Chow JH. Power system dynamics and stability: with synchrophasor measurement and power system toolbox. John Wiley & Sons; 2017.
[4] Stevens RJ, Meneveau C. Flow structure and turbulence in wind farms. Annu Rev Fluid Mech 2017;49:311–39.
[5] Fouad A-A, Vittal V. Power system transient stability analysis using the transient energy function method. Pearson Education; 1991.
[6] Chiang H-D. Direct methods for stability analysis of electric power systems: theoretical foundation, BCU methodologies, and applications. John Wiley & Sons; 2011.
[7] Pai M. Power system stability: analysis by the direct method of Lyapunov, vol. 3, North-Holland; 1981.
[8] Pai M. Energy function analysis for power system stability. Springer Science & Business Media; 1989.
[9] Khalil HK. Nonlinear systems third edition, vol. 115, Patience Hall; 2002.
[10] Parrilo PA. Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization. California Institute of Technology; 2000.
[11] Zou Y, Yin M-H, Chiang H-D. Theoretical foundation of the controlling UEP method for direct transient-stability analysis of network-preserving power system models. IEEE Trans Circuits Syst I 2003;50(10):1324–36.
[12] Treinen RT, Vittal V, Kliemann W. An improved technique to determine the controlling unstable equilibrium point in a power system. IEEE Trans Circuits Syst I 1996;43(4):313–23.
[13] Chiang H-d, Thorp JS. The closest unstable equilibrium point method for power system dynamic security assessment. In: 26th IEEE conference on decision and control. vol. 26, IEEE; 1987, p. 72–6.
[14] Liu C-W, Thorp JS. A novel method to compute the closest unstable equilibrium point for transient stability region estimate in power systems. IEEE Trans Circuits Syst I 1997;44(7):630–5.
[15] Chiang H-D, Tong J, Tada Y. On-line transient stability screening of 14,000-bus models using TEPCO-BCU: Evaluations and methods. In: IEEE PES general meeting. IEEE; 2010, p. 1–8.
[16] Kakimoto N. Transient stability analysis of electric power system via lur'e type Lyapunov function. Trans IEE Jpn E 1978;98(5):63–79.
[17] Anghel M, Milano F, Papachristodoulou A. Algorithmic construction of Lyapunov functions for power system stability analysis. IEEE Trans Circuits Syst I Regul Pap 2013;60(9):2533–46.
[18] Tan W, Packard A. Stability region analysis using polynomial and composite polynomial Lyapunov functions and sum-of-squares programming. IEEE Trans Autom Control 2008;53(2):565–71.
[19] Franze G, Famularo D, Casavola A. Constrained nonlinear polynomial time-delay systems: A sum-of-squares approach to estimate the domain of attraction. IEEE Trans Autom Control 2012;57(10):2673–9.
[20] Choi H, Seiler PJ, Dhople SV. Robust power systems stability assessment with sum of squares optimization. In: 2015 IEEE power & energy society general meeting. IEEE; 2015, p. 1–5.
[21] North American Electric Reliability Corporation. Electromagnetic transient modeling for BPS connected inverter-based resources—Recommended model requirements and verification practices. In: NERC reliability guidelines. 2023.
[22] Koopman BO, Neumann Jv. Dynamical systems of continuous spectra. Proc Natl Acad Sci 1932;18(3):255–63.
[23] Williams MO, Kevrekidis IG, Rowley CW. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. J Nonlinear Sci 2015;25(6):1307–46.
[24] Tu JH. Dynamic mode decomposition: theory and applications. (Ph.D. thesis), Princeton University; 2013.
[25] Kutz JN, Brunton SL, Brunton BW, Proctor JL. Dynamic mode decomposition: data-driven modeling of complex systems SIAM; 2016.
[26] Hou B, Bose S, Vaidya U. Sparse learning of kernel transfer operators. In: 2021 55th asilomar conference on signals, systems, and computers IEEE; 2021, p. 130–4.
[27] Hou B, Sanjari S, Dahlin N, Bose S. Compressed decentralized learning of conditional mean embedding operators in reproducing kernel Hilbert spaces. In: Proceedings of the AAAI conference on artificial intelligence vol. 37, 2023, p. 7902–9.
[28] Hou B, Matavalam ARR, Bose S, Vaidya U. Propagating uncertainty through system dynamics in reproducing kernel Hilbert space. Physica D 2024;134168.
[29] Lasota A, Mackey MC. Chaos, fractals, and noise: stochastic aspects of dynamics, vol. 97, Springer Science & Business Media; 1998.
[30] Vaidya U, Mehta PG. Lyapunov measure for almost everywhere stability. IEEE Trans Autom Control 2008;53(1):307–23.
[31] Mezić I. Spectral properties of dynamical systems, model reduction and decompositions. Nonlinear Dynam 2005;41(1):309–25.
[32] Mauroy A, Mezic I. Global stability analysis using the eigenfunctions of the Koopman operator. IEEE Trans Autom Control 2016;61(11):3356–69.
[33] Budisic M, Mohr R, Mezic I. Applied koopmanism. Chaos 2012;22(4):047510.
[34] Susuki Y, Mezić I. Nonlinear koopman modes and power system stability assessment without models. IEEE Trans Power Syst 2013;29(2):899–907.
[35] Susuki Y, Mezic I, Raak F, Hikihara T. Applied Koopman operator theory for power systems technology. Nonlinear Theory Appl IEICE 2016;7(4):430–59.
[36] Xu Y, Netto M, Mili L. Propagating parameter uncertainty in power system nonlinear dynamic simulations using a Koopman operator-based surrogate model. IEEE Trans Power Syst 2022;37(4):3157–60.
[37] Hou B, Sanjari S, Dahlin N, Bose S, Vaidya U. Sparse learning of dynamical systems in RKHS: An operator-theoretic approach. In: Proceedings of the 40th international conference on machine learning ICML 2023, 2023.
[38] Di Natale L, Lian Y, Maddalena ET, Shi J, Jones CN. Lessons learned from data-driven building control experiments: Contrasting gaussian process-based mpc, bilevel deepc, and deep reinforcement learning. In: 2022 IEEE 61st conference on decision and control CDC, IEEE; 2022, p. 1111–7.
[39] Mauroy A, Mezić I. On the use of Fourier averages to compute the global isochrons of (quasi) periodic dynamics. Chaos 2012;22(3).
[40] Klus S, Schuster I, Muandet K. Eigendecompositions of transfer operators in reproducing kernel Hilbert spaces. J Nonlinear Sci 2020;30(1):283–315.
[41] Sauer PW, Pai MA. Power system dynamics and stability. 2006.
[42] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: Machine learning in python. J Mach Learn Res 2011;12:2825–30.
[43] Yeung E, Kundu S, Hodas N. Learning deep neural network representations for koopman operators of nonlinear dynamical systems. In: 2019 American control conference ACC, IEEE; 2019, p. 4832–9.
[44] Nandanoori SP, Sinha S, Yeung E. Data-driven operator theoretic methods for global phase space learning. In: 2020 American control conference ACC, IEEE; 2020, p. 4551–7.
[45] Li Z, Meunier D, Mollenhauer M, Gretton A. Optimal rates for regularized conditional mean embedding learning. Adv Neural Inf Process Syst 2022;35:4433–45.
[46] Hou B, Sanjari S, Koppel A, Bose S. Compressed online learning of conditional mean embedding. 2024, arXiv preprint arXiv:2405.07432.
[47] Song L, Huang J, Smola A, Fukumizu K. Hilbert space embeddings of conditional distributions with applications to dynamical systems. In: Proceedings of the 26th annual international conference on machine learning 2009, p. 961–8.
[48] Klebanov I, Schuster I, Sullivan TJ. A rigorous theory of conditional mean embeddings. SIAM J Math Data Sci 2020;2(3):583–606.
[49] Richard C, Bermudez JCM, Honeine P. Online prediction of time series data with kernels. IEEE Trans Signal Process 2008;57(3):1058–67.
[50] Athay T, Podmore R, Virmani S. A practical method for the direct analysis of transient stability. IEEE Trans Power Appar Syst 1979;(2):573–84.
[51] Illinois Center for a Smarter Electric Grid (ICSEG). IEEE 39-bus system. 2023, [Online]. Available: https://icseg.iti.illinois.edu/ieee-39-bus-system/.