

RESEARCH ARTICLE

Open Access



Reduced order modeling for transient simulation of power systems using trajectory piece-wise linear approximation

Muhammad Haris Malik^{1*} , Domenico Borzacchiello¹, Francisco Chinesta¹ and Pedro Diez²

*Correspondence:
muhammad.malik@ec-nantes.fr
¹Institut de Calcul Intensif (ICI),
École Centrale de Nantes, 1 rue
de la Noë, BP 92101, 44321
Nantes Cedex 3, France
Full list of author information is
available at the end of the article

Abstract

This paper concerns the application of reduced order modeling techniques to power grid simulation. Swing dynamics is a complex non-linear phenomenon due to which model order reduction of these problems is intricate. A multi point linearization based model reduction technique trajectory piece-wise linearization (TPWL) method is adopted to address the problem of approximating the nonlinear term in swing models. The method combines proper orthogonal decomposition with TPWL in order to build a suitable reduced order model that can accurately predict the swing dynamics. The method consists of two stages, an offline stage where model reduction and selection of linearization points is performed and an online stage where the reduced order multi-point linear simulation is performed. An improvement of the strategy for point selection is also proposed. The TPWL method for a swing dynamics model shows that the method provides accurate reduced order models for non-linear transient problems.

Keywords: Model order reduction, Power grid, Swing dynamics, POD, TPWL

Background

Transient stability of the power grids is a keen area of research because of its implications in the power system planning, operation and control. Energy based methods for the transient analysis of power grids were developed originally by Mangnusson [1] and Aylett [2]. The substantial size of power grids makes the transient analysis computationally costly to simulate and therefore a need of model order reduction arises. Reduced order models need to be computationally cost-effective while retaining considerable accuracy of the full model in large network grids simulations.

A number of approximation schemes are available for model reduction and selection of an appropriate scheme depends upon the problem to be solved so that a suitable reduced order model is achieved [3]. Some of the earliest methods in the domain of model order reduction are Truncated Balance Realization proposed by Moore in 1981, Hankel-norm reduction published in 1984 by Glover [4], proper orthogonal decomposition (POD) [5], asymptotic waveform evaluation, PRIMA [4] and a more recent proper generalized decomposition (PGD) [6].

The techniques defined by Bai et al. [3] includes Krylov-subspace techniques, Lanczos based methods such as MPVL algorithm and SyMPVL among others for the reduced order modeling in the electromagnetic applications. The study by Parillo et al. [7] presents the use of POD to reduce the hybrid, nonlinear model of a power network. The authors used the “Swing Equations” which are the differential-algebraic equations of second order to simulate the cascading failures in power systems. They employed model order reduction in parts of the system that remains unaffected by the failures. POD based model order reduction has found applications in diverse fields and is the preferred method in electrical engineering applications such as in the study by Montier et al. [8]. In their study, POD is applied in combination with discrete empirical interpolation method.

Kashyap et al. [9] have used model order reduction for the purpose of state estimation of phasor measurement units (PMUs). The authors have proposed an algorithm based on reduced-dimension matrices which operate separately on PMU measurements and on conventional measurements. The proposed scheme is applicable to distributed implementation and is reported to be numerically stable. The algorithm was applied on a IEEE 14-bus system and compared with existing schemes in the literature and demonstrated good accuracy. Wille-Hausmann et al. [10] used symbolic reduction approach to model lower order grid segments. The study shows reduction by a factor of 2 for a typical grid.

The main hurdle in the effective model order reduction of the power grids is the strong nonlinearity appearing in swing dynamics models. Trajectory Piecewise-Linear method (TPWL) is a well-defined method for the model order reduction of nonlinear time varying applications [11]. This method proposes a suitable strategy for treatment of nonlinearities which presents the real bottleneck of model order reduction. This method has been applied on several nonlinear problems especially to electronics engineering applications [12–19]. A similar method to the one adopted in this paper is found in the work of Bugard et al. [20] and Panzer et al. [21] who have proposed a parametric model order reduction. The main idea presented in these works is to reduce several local models and then produce a parametric reduced order model using a suitable interpolation strategy. Compared to these methods, TPWL has one global reduced basis and uses interpolation of locally linearized models just to represent the nonlinear term in the reduced variables space. More than one training trajectories can be added together to form the single global reduced basis similar to the concept of POD.

TPWL method has been implemented in non-linear control of integrated circuits and MEMS [14]. Xie and Theodoropoulos [22] have used the capability of TPWL of reducing large scale non-linear dynamic models and demonstrated it through the stabilisation of the oscillatory behavior tubular reactors as the case study.

Trajectory piecewise-linear methods are not limited to just power electronics and control systems applications, indeed there are vast areas of research where the application of TPWL based model order reduction will be beneficial [23,24]. The study by He et al. [23] involves the implementation of TPWL macromodeling for subsurface flow simulations. In another study by Cardoso and Durlofsky [24] the work on model order reduction using TPWL methods for subsurface flow simulations is presented.

In the current study, we implemented TPWL method to accurately obtain a reduced order model for the nonlinear transient dynamics of power grids, mathematically modeled by swing dynamics. The swing dynamics model is highly non-linear and it is very difficult to have accurate results with linearized reduced order models and the nonlinear POD is

inefficient with respect to time consumption. Therefore, the adoption of TPWL method in the current study is suitable for the model order reduction of power grids.

The article has been divided into three sections, where “Swing dynamics equations” section discusses about the mathematical model of the power grid. “Model order reduction” section deals with the model reduction techniques of POD and introduces the method of TPWL adapted for the swing model. “Numerical experiments” section provides a couple of test cases as an example and the robustness of the application of TPWL method.

Swing dynamics equations

Problem statement

A mathematical model to describe the transient dynamics of power systems is the “Swing Dynamics” [25]. It involves a second order differential equation representing the generator node or bus which originates from the rotor dynamics of the generator and an algebraic equation associated with the load bus. The differential equation for the i th bus:

$$m_i \ddot{\delta}_i + d_i \dot{\delta}_i = p_i^m - p_i \quad \text{for } i = 1, \dots, N \tag{1}$$

The unknown in the Eq. (1) is $\delta_i(t)$ and i varies from $i = 1, \dots, N$ where N represents the number of nodes in the system. The variables δ_i represent the generator rotor angle derivations with respect to a synchronously rotating frame, while $\dot{\delta}_i$ and $\ddot{\delta}_i$ are respectively the first and the second time derivatives of the rotor angle. The quantities p_i^m and p_i are the mechanical power input and the electrical power output and are given. The parameters m_i and d_i are the i th generator’s normalized inertia and damping coefficients.

The expression for the electrical power output is given by:

$$p_i = \sum_{k=1}^N |V_i| |V_k| b_{ik} \sin(\delta_i - \delta_k) \quad \text{for } i = 1, \dots, N \tag{2}$$

In Eq. (2) $V_i = |V_i|e^{i\delta_i}$, and $y_{ik} = g_{ik} + ib_{ik}$ represents the complex admittance matrix with b_{ik} is the line susceptance and g_{ik} is the line conductance, it is assumed that voltage magnitudes $|V_i|$ do not change and the transmission line losses are negligible, i.e. y_{ik} is purely imaginary ($g_{ik} = 0$). Using Eq. (2) in Eq. (1) gives,

$$m_i \ddot{\delta}_i + d_i \dot{\delta}_i = p_i^m - \sum_{k=1}^N |V_i| |V_k| b_{ik} \sin(\delta_i - \delta_k) \quad \text{for } i = 1, \dots, N \tag{3}$$

Equation (3) describes the transient dynamics of the power system under the assumption that the lines are purely reactive and voltage magnitudes are kept constant [7] since all the nodes are considered as PV nodes.

In the current study, POD is employed for the purpose of model order reduction. It is to be noted in the framework of POD mathematical manipulations of system (3) are more easily handled using matrix and vector representations. Hence, we present the system of equations in matrix form:

$$[M]\{\ddot{\delta}\} + [D]\{\dot{\delta}\} = \{p^m\} - \{p(\{\delta\})\} \tag{4}$$

where, $[M]$ and $[D]$ are $N \times N$ diagonal matrices, while $\{\ddot{\delta}\}$, $\{\dot{\delta}\}$ and $\{\delta\}$ are vectors in \mathbb{R}^N , N being the number of nodes in the grid. Note that, the notation $[M]\{\ddot{\delta}\}$ stands for the matrix-vector product. The vector $\{p(\{\delta\})\}$ is a nonlinear function of $\{\delta\}$. Eq. (4) is the high fidelity model of the swing dynamics in contrast to the reduced order model that is developed based on the high fidelity model.

Numerical integration of the high fidelity model

Swing equations are numerically integrated in the commercial software MATLAB. A built-in function ‘ode15s’ has been used in the evaluation of the time dependent ODE represented in Eq. (4), because of the potential stiffness of the problem this is the most effective ODE solver available in MATLAB. It is based on the numerical differentiation formulas (NDF) and optionally use backward differentiation formulas (BDF) and is a multistep solver. A detailed description of this method and its incorporation in MATLAB environment is provided in the article by Shampine and Reichelt [26].

Model order reduction

Proper orthogonal decomposition

Proper Orthogonal Decomposition is an “a posteriori” method for model order reduction. The objective is to find an orthonormal basis considerably smaller as compared to the high fidelity model using the information extracted from previously computed simulations. Say δ is a vector of dimension N containing all the state variables of the system, the objective of POD is to reduce the dimension from N to q where $q \ll N$. The mapping from the original to the reduced coordinates is expressed by the linear application:

$$\{\delta\} = [\tilde{U}]\{z\} \tag{5}$$

where $\{\delta\}$ is a $N \times 1$ vector, $[\tilde{U}]$ is a $N \times q$ matrix, and $\{z\}$ is a $q \times 1$ vector. The goal of POD is to compute matrix $[\tilde{U}]$ from the analysis of the principal components of the available solutions.

The matrix $[\tilde{U}]$ can be calculated using several techniques. In essence, POD is similar to the Karhunen–Loeve decomposition (KLD) and it is often referred to as KLD, principal component analysis (PCA) or the singular value decomposition (SVD) [27]. In the current study, SVD interpretation has been used to obtain a reduced order model.

Here, we will briefly describe the SVD reduction procedure which is available as a built-in function in MATLAB.

A selection of solution “snapshots” $\{\delta\}_k$, with $k = 1, 2, \dots, n$, are arranged into the columns of the matrix $[Q] \in \mathbb{R}^{N \times n}$,

$$[Q] = [\{\delta\}_1, \{\delta\}_2, \dots, \{\delta\}_n] \tag{6}$$

The number of snapshots must guarantee that they represent the complete set of solutions, i.e., n must be large enough. The set of n snapshots contains redundant information that have to be suppressed by keeping only the pertaining remaining modes q .

The factorization under SVD is given as:

$$\begin{aligned} [Q] &= [U][\Sigma][V]^* \\ &= \sum_{i=1}^N \sigma_i \{U_i\} \{V_i\}^T \approx \sum_{i=1}^q \sigma_i \{U_i\} \{V_i\}^T \end{aligned} \tag{7}$$

where, $[U]$ is a $N \times N$ matrix, $[\Sigma]$ is a $N \times n$ diagonal matrix with non-negative real numbers on the diagonal, and $[V]^*$ is a $n \times n$, unitary matrix, $[V]^*$ is the conjugate transpose of the $n \times n$ unitary matrix $[V]$. The left hand side of Eq. (7) accurately estimate the full $[Q]$ matrix for $i = 1, \dots, N$. The last sum is the truncation of first terms that sufficiently approximates the full $[Q]$ matrix.

To obtain a reduced order model which retains minimum energy required to accurately capture the behavior of the high fidelity model, it is truncated at q where $q \ll N$. Number

of modes q are selected such that for any $j > q$, the quotient η is within some defined tolerance and $\sigma_j \ll \sigma_1$, where quotient η is defined as the difference of the relative energy retained, where $\eta = 0$ means that the total energy of the system is retained. It is usually the case that only few of the larger σ_j contain the most energy and the rest σ_j for $j = q, \dots, N$ can be dropped from the reduced basis.

$$\eta = \left| \frac{\sum_{j=1}^q \sigma_j}{\sum_{k=1}^N \sigma_k} - 1 \right| \tag{8}$$

The quotient η is employed to find the size of the reduced basis which accurately mimics the original basis, typical values are between 10^{-1} and 10^{-5} . The matrix $[\tilde{U}]$ is given as:

$$[\tilde{U}] = [\{U\}_1, \{U\}_2, \dots, \{U\}_q], \quad q < N < n \tag{9}$$

The columns of $[\tilde{U}]$ correspond to vectors $\{U\}_i$ representing the most characteristic *modes* in the solution, that is, the most recurrent structures.

For detailed insight into the method and the variations in the above mentioned procedures of KLD, PCA and SVD, the author refers to the studies by Liang et al. [27]. Additionally one can also refer to Kerschen et al. [28] and Berkooz et al. [29].

The reduced order model for the governing equations is obtained by replacing $\{\delta\}$ with the relation given by Eq. (5) in Eq. (4),

$$[M][\tilde{U}]\{\ddot{z}\} + [D][\tilde{U}]\{\dot{z}\} = \{p^m\} - \{p\} \tag{10}$$

and using Galerkin method to project the residual on the reduced basis

$$[\tilde{U}]^T [M][\tilde{U}]\{\ddot{z}\} + [\tilde{U}]^T [D][\tilde{U}]\{\dot{z}\} = [\tilde{U}]^T \{p^m\} - [\tilde{U}]^T \{p(\{\delta\})\} \tag{11}$$

Defining the following notations

$$\begin{aligned} [\tilde{M}] &:= [\tilde{U}]^T [M] [\tilde{U}] \\ [\tilde{D}] &:= [\tilde{U}]^T [D] [\tilde{U}] \\ \{\tilde{p}^m\} &:= [\tilde{U}]^T \{p^m\} \\ \{\tilde{p}(\{z\})\} &:= [\tilde{U}]^T \{p(\{\delta\})\} \end{aligned} \tag{12}$$

we obtain the governing Eq. (4) in the reduced basis as:

$$[\tilde{M}]\{\ddot{z}\} + [\tilde{D}]\{\dot{z}\} = \{\tilde{p}^m\} - \{\tilde{p}(\{z\})\} \tag{13}$$

Note that $\{\tilde{p}\}$ is a nonlinear function of $\{z\}$.

Although it is possible to use POD on nonlinear problems for model order reduction, the necessity of evaluating the nonlinear function renders it less practical in terms of computational complexity. In the current study, we are using MATLAB based ODE solver ‘ode15s’, which requires that the nonlinear function to be defined analytically. Recall the Eq. (2) describing the nonlinear function

$$p_i = \sum_{k=1}^N |V_i| |V_k| b_{ik} \sin(\delta_i - \delta_k) \quad \text{for } i = 1, \dots, N$$

The nonlinear function here depends upon the original basis δ , in the reduced basis it has to be defined as:

$$\begin{aligned} p_i(\tilde{U}z) &= \sum_{k=1}^N |V_i| |V_k| b_{ik} \sin \left(\sum_{j=1}^q (\tilde{U}_{ij} - \tilde{U}_{kj}) z_j \right) \quad \text{for } i = 1, \dots, N \\ \tilde{p}_l(z) &= \sum_{i=1}^N \tilde{U}_{il} p_i(\tilde{U}z) \quad \text{for } l = 1, \dots, q \end{aligned} \tag{14}$$

The nonlinear function $\tilde{p}(z)$ in the reduced basis therefore becomes,

$$\tilde{p}_l(z) = \sum_{i=1}^N \tilde{U}_{il} \sum_{k=1}^N |V_i| |V_k| b_{ik} \sin \left(\sum_{j=1}^q (\tilde{U}_{ij} - \tilde{U}_{kj}) z_j \right) \quad \text{for } l = 1, \dots, q \quad (15)$$

It is evident from the Eq. (15), it requires $\mathcal{O}(N^2 \times q^2)$ operations, which is counter-productive to the reduced order modeling. To save computation costs and truly exploit the benefits of reduced order modeling, it would be necessary to eliminate the N^2 number of operations which is the dimension of original basis. Therefore, a method of trajectory piece-wise linear method (TPWL) has been proposed that is described in the following section.

As an example of the problem with increased computational cost, we performed a simulation with POD based reduced model, the same simulation is presented later in detail. The full simulation without the model reduction required around 202 s, with POD and the nonlinear function as defined in Eq. (15) the simulation took about 290 s.

Trajectory piece-wise linear method

An approach based on TPWL method has been adopted in the current study to reap the benefits of reduced order modeling while also maintaining a good approximation of the nonlinear function.

Evaluation of $\{\tilde{p}(z)\}$ requires $N^2 \times q^2$ operations which results in similar time consumption as high fidelity model. The objective of introducing TPWL method is to construct a locally affine mapping $\{\tilde{L}_p(z)\}$ from \mathbb{R}^q to \mathbb{R}^q at some time steps s where $s \ll n$ which involves less operations and such that $\{\tilde{L}_p(z)\} \approx \{\tilde{p}(z)\}$.

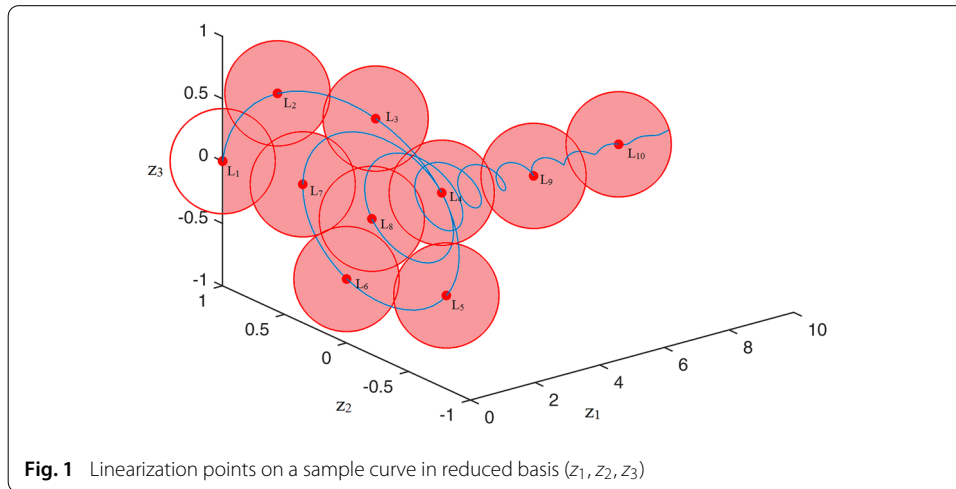
Trajectory piece-wise linear method is a method combining the model order reduction and the linearization of the non-linear functions. The system in the current study given by Eq. (1) has strong non-linear characteristic and as described in earlier sections, nonlinear reduced order model does not reduce the time consumption. The TPWL method provides a combination of linearized models obtained at selected snapshots.

To accurately capture the behavior of nonlinear function, it is important that the points selected for the linearization should be such that they span the whole manifold in which the system trajectories evolve. As an example of this, Fig. 1 shows the typical trajectory in the space of reduced variable $\{z\}$ and in this the selection of linearization points is made to ensure that the trajectory is completely covered.

Steps of TPWL Simulation

TPWL method can be separated into an offline and an online stage. The steps during the offline phase are below:

- Step 1: Simulation of the high fidelity model used as the training trajectories (See details in “Selection of training trajectories” section).
- Step 2: Generation of reduced basis using POD.
- Step 3: Linearization of the nonlinear function and construction of the set of linearization points S (See details in “Selection of linearization points” section). During the construction of set S , weights have to be computed for the combination of linear functions which is detailed in the “Weighting function” section.



- Step 4: Reduction of the linearized system and the storage of $\{\delta\}^j, \{\bar{p}\}^j, [\bar{J}]^j$ and S , where j is the element in the set of linearization points S .

Once the above steps are performed and we have a reduced linear system with $\{\delta\}^j, \{\bar{p}\}^j, [\bar{J}]^j$ and S available, we can perform the simulation of the reduced system in an online phase of TPWL. The following steps comprise the online phase of TPWL.

- Step 1: Load the stored reduced basis.
- Step 2: Calculate weighting functions.
- Step 3: Combine the linearized systems in a convex combination.
- Step 4: Solve the reduced linearized system.

In the following sections we will present the methodology to select the linearization points and also the weighting procedure for the combination of the linearization points.

Selection of training trajectories

Training trajectories form an integral part of the TPWL method which theoretically, should be able to cover all the domain of the nonlinear function. The selection of training trajectories, therefore, requires careful selection of initial conditions upon which the trajectories of the swing model depend. One may consider it is inefficient to compute so many nonlinear functions to cover the whole domain. However, in practice there are only a few possible conditions a system can achieve in real time applications. Training trajectories provide the points at which the nonlinear system has to be linearized (see “Selection of linearization points” section). It is to be stressed that the TPWL method can interpolate between the training trajectories but not to extrapolate. Therefore, it is necessary to include all the trajectories in the training set that are considered to be visited by the nonlinear function [30].

Selection of linearization points

Selection of linearization points has been done during the offline stage of the TPWL method and is performed on the full original dimension of the system. This is due to the reason that the construction of the linear approximation $\{\tilde{L}_p(z)\}$ is done with the

information already available from the training trajectories. Therefore, we present the equations in terms of the original basis $\{\delta\}$, the nonlinear function $\{p(\delta)\}$ in Eq. (4) is linearized at certain points in time. This function at a generic snapshot is given by local approximation $\{L_p^j(\delta)\}$ as:

$$\{L_p^j(\delta)\} \approx \{p\}^j + [J]^j (\{\delta\} - \{\delta\}^j) \quad \text{for } j = 1, \dots, s \tag{16}$$

This is the first step in the TPWL simulation after the full nonlinear solutions have been obtained at the predefined training trajectories. The initial conditions are represented by $\{\delta\}^0$ and it is by default the first point selected for the linearization. The Jacobian is given by matrix $'[J]'$ as:

$$[J] = \frac{\partial \{p\}}{\partial \{\delta\}} = \begin{bmatrix} \frac{\partial \{p\}}{\partial \delta_1} & \dots & \frac{\partial \{p\}}{\partial \delta_N} \end{bmatrix} \tag{17}$$

Note that, the jacobian can be derived analytically for the swing equations.

The linearized function approximates the actual nonlinear function using the Jacobian at the specified linearization points, represented by j in the superscript. The idea behind the TPWL method is that a number of linearization points are selected and the linearized functions at those snapshots are summed up by a weighting function given a global approximation as presented in Eq. (18). A detail on the selection of linearization points has been presented in Algorithm 1

$$\{L_p(\delta)\} \approx \sum_{j=1}^s w^j \cdot (\{p\}^j + [J]^j (\{\delta\} - \{\delta\}^j)) \tag{18}$$

where, L_p is the linear approximation, $\{p\}^j$, $[J]^j$ are the nonlinear function and Jacobian matrix evaluated at the j th linearization point, and $\{\delta\}^j$ is the j th linearization point. The weight \hat{w}^j is given by (19), which depends on the distance d^j between $\{\delta\}$ and linearization point $\{\delta\}^j$, and the normalized form is denoted by w^j which appears in Eq. (18).

$$\hat{w}^j = e^{-\beta d^j / d^{min}} \tag{19}$$

where β is a positive constant and it can be adjusted to reduce the error and smooth the affine function $\{L_p\}$, d^j is the distance between $\{\delta\}$ and linearization point $\{\delta\}^j$ and d^{min} is the minimum among d^j .

The strategy to select the linearization points is traditionally based on the difference between the phase differences between the successive time steps, i.e., $(\delta_i - \delta_{i-1})^j - (\delta_i - \delta_{i-1})^{j-1}$. If the phase difference is greater than some angle, e.g. 10° , than this point j will be added to the set of linearization points. This method is widely used in the studies referenced in our work as presented in the works of Albunni [30] and Rewiński et al. [31]. The limitation in this method is that when the selection is based on the distance between points, there is no control over the error. This can lead to significant error especially if the Jacobian is close to singular. The alternative method adopted in this study is to place the linearization points adaptively based on an error indicator. This is obtained by the difference between the nonlinear function and the approximation obtained by TPWL method. When a point is encountered for which the error is greater than a given threshold, the TPWL model is enriched with a new point. This procedure is applied recursively to all the snapshots of the training simulations, starting with a single point that is the initial condition.

A simple nonlinear trigonometric function and its linear approximation based on the concept of fixed distance in the context of its norm is plotted in Fig. 2a while a new method developed in the current study is shown in Fig. 2b . As it is observable from Fig. 2a and b , the number of linearization points are comparable as 5 in the first case to 7 in the current case has reduced the approximation error by about 40%. Generally, the increase in number of linearization points is of no significant loss in computation time as the selection is done during the offline phase while in the online phase the computation of linear functions is very quick.

This approach is similar to the method proposed in the study of Liu et al. [12] in which the authors have used a global maximum error for linearization point selection.

Algorithm 1 Build set of linearization points

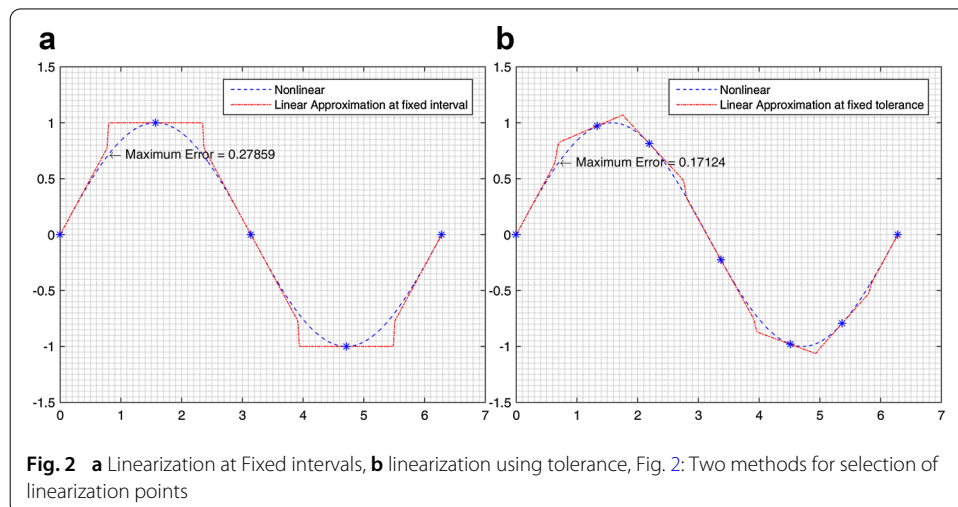
```

Input  $\{\delta\}^i, \{p\}^i, [J]^i, i = 1, \dots, n$ 
Initialize set of linearization points with initial condition as the first point  $\{\hat{\delta}\}^1 = \{\delta\}^0$ 
 $S = \{\hat{\delta}\}^1, s = 1$ 
 $\{\hat{p}\}^1 = \{p\}^1, [\hat{J}]^1 = [J]^1$ 
Set tolerance  $\epsilon_s$  for the selection of new point of linearization
 $\epsilon_s = 0.005$ 
while  $i < n$  do
  Linearize  $\{p\}$  and combine in a weighted sum
  for  $j = 1$  to  $s$  do
    Compute  $\hat{w}^j$  according to eq. (19) and normalize to get  $w^j$ 
     $\{L_p(\{\delta\}^i)\} = \sum_{j=1}^s w^j \cdot \{p\}^j + [J]^j \{\delta\}^i - \{\delta\}^j$ 
    if  $\{p(\{\delta\}^i)\} - \{L_p(\{\delta\}^i)\} > \epsilon_s$  then
      Increase the size of set of linearization points
       $\{\delta\}^s = \{\delta\}^i, \{\hat{p}\}^s = \{p\}^i, [\hat{J}]^s = [J]$ 
       $S \leftarrow S \cup \{\hat{\delta}\}^s$ 
    end if
  end for
end while
Output  $S = \{\{\hat{\delta}\}^1, \dots, \{\hat{\delta}\}^s, s = \text{card}(S), \{\hat{p}\}^j, [\hat{J}]^j, j = 1, \dots, s$ 

```

The modified version of the selection of linearization points is more time consuming than the original method proposed by Albunni [30]. However, selection of linearization points is performed during the offline phase where time is not a constraint, the proposed method in the current study has higher accuracy with the problem discussed here.

A very important note that the nonlinear function $\{\hat{p}\}^j$ and the Jacobian matrix $[\hat{J}]^j$ are stored in the reduced basis.



Weighting function

TPWL method combines the linearized model in a convex combination approximating the original nonlinear system. In a convex combination all the coefficients are greater or equal to zero with the sum of all the coefficients equal to one. If there are ‘s’ linearized models and ‘q’ is the order of the linearized system, then the computation of these weights is in the order of $O(sq)$ (for detailed study on the weighting function refer to thesis of Rewiński [11]). The calculation of weights is carried out during both the online and offline phases in the current study.

Given the set of linearization points S and β , the weights can be calculated for any point with respect to the points in the linearization set. The value of β should be a positive constant, in the current study its value is 25 . A smaller value smooths the function and make its appearance continuous, while a higher value results in kinks in the function. The necessity of adjusting β is that it helps to reduce the error between the nonlinear function and its approximation. The first step is to compute the distance d^j between a point $\{\delta\}$ and all the points in the set S using,

$$d^j = ||\{\delta\} - \{\delta^j\}||_2 \quad \text{for } j = 1, \dots, s \tag{20}$$

The weights are then calculated as,

$$\hat{w}^j = e^{-\beta d^j / d^{min}} \quad \text{for } j = 1, \dots, s \tag{21}$$

where, $d^{min} = \min_{j=1, \dots, s} d^j$

Once, the weights are calculated with respect to all the points in the set S , the weights are normalized as:

$$w^j = \frac{\hat{w}^j}{\sum_{i=1}^s \hat{w}^i}, \quad \text{for } j = 1, \dots, s, \quad i = 1, \dots, s \tag{22}$$

During the online phase, in place of $\{\delta\}$ reduced basis $\{z\}$ is used to calculate the distance and the points in the set S consists of linearization points in the reduced basis as well.

Numerical integration of the reduced model

Once a linear model has been obtained from the training trajectories during the offline phase, the values of the nonlinear function and the Jacobian matrix evaluated at the selected linearization points are reduced and stored to be used during the online phase along with the set of linearization points. The nonlinear function and the Jacobian is projected in the reduced space as:

$$\begin{aligned} \{\tilde{p}(\{z\})\} &= [\tilde{U}]^T \{p(\{\delta\})\} \\ [\tilde{J}] &= [\tilde{U}]^T [J] [\tilde{U}] \end{aligned} \tag{23}$$

Since, we replace the nonlinear function with an approximation containing sum of linearized functions, we have

$$\begin{aligned} \{\tilde{L}_p(\{z\})\} &= [\tilde{U}]^T \{L_p([\tilde{U}]\{z\})\} \\ \Rightarrow \{\tilde{L}_p(\{z\})\} &\approx \sum_{j=1}^s w^j \cdot \left(\{\tilde{p}\}^j + [\tilde{J}]^j (\{z\} - \{z\}^j) \right) \end{aligned} \tag{24}$$

In the above equation, the weights w^j are evaluated afresh in the reduced dimension. The weights are calculated exactly as described in the “Weighting function” section with $\{z\}$ replacing the $\{\delta\}$.

With the above information, we now have a system in reduced basis which fully exploits the benefits of reduced order modeling. As it can be observed from Eq. (24), the number of operations now depend on s rather than N^2 .

Numerical experiments

The network grid studied here is termed as the “Ring Grid” consisting of only generators with one reference node connected to all the generators, represented in Fig. 3. The mathematical model describing the ring grid is the swing dynamics given by the Eq. (1). The grid in this study is a ring grid containing all the generator nodes and a slack node in a topology such that the slack node is connected with all the generators. A slack node or bus in electrical power system is a bus where both $|V|$ and δ are known and is used to balance the power losses or power demand shortage while performing a power flow study [32]. Here, the slack bus is modeled as an infinite bus which is a simplifying assumption that the voltage at this bus is always constant and it has infinite power capacity as the impedance is zero for this bus. A list of assumptions for the grid in the current study are:

- The power grid is loss-less
- The generators are small and the ratio between the length of transmission line joining generators to the infinite bus and the length of transmission line joining two consecutive generators is much bigger. Hence, the interaction between a generator and infinite bus is much smaller than the interaction between two neighboring generators
- Transmission lines joining two consecutive generators is shorter than the line joining the generators with the infinite bus
- Transmission lines between the infinite bus and all the generators are of same length
- Transmission lines connecting the generators are of same length

The nonlinear function p_i in Eq. (1) is different from its form given in Eq. (2) due to the assumptions described earlier and is given as:

$$p_i = b \sin(\delta_i) + b_{int}[\sin(\delta_i - \delta_{i+1}) + \sin(\delta_i - \delta_{i-1})] \quad \text{for } i = 1, \dots, N \tag{25}$$

where, b is line susceptance between generators and the reference node and b_{int} is the line susceptance between two connected generators.

The Eq. (1) takes the following form

$$m_i \ddot{\delta}_i + d_i \dot{\delta}_i = p_i^m - b \sin(\delta_i) - b_{int}[\sin(\delta_i - \delta_{i+1}) + \sin(\delta_i - \delta_{i-1})] \tag{26}$$

for $i = 1, \dots, N$

The grid studied consists of 1000 generators and one slack node connected to all the generators. The data used in the study is given in Table 1, all the values are given in per-unit system.

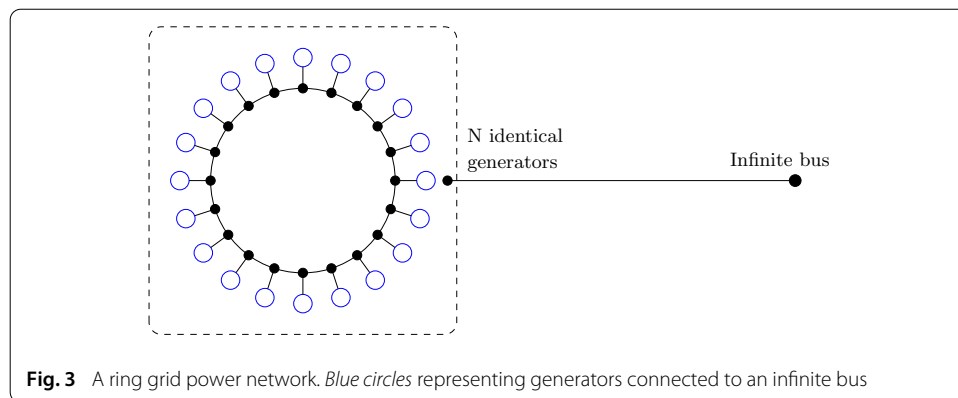


Fig. 3 A ring grid power network. Blue circles representing generators connected to an infinite bus

Table 1 Grid data

Symbol	Description	Value
m_i	Mass of the generators	1 (p.u.)
d_i	Damping of the generators	0.25 (p.u.)
p^m	Power generated by the generators	0.95 (p.u.)
b	Susceptance between generator and slack node	1 (p.u.)
b_{int}	Susceptance between consecutive generators	100 (p.u.)
N	Number of generators	1000

The values used in the current study are adapted from the study of Susuki et al. [33] with the addition of damping to ensure the steady state stability of the power grid. Also, the number of generators in the study of Susuki et al. are only 20 and the focus of their study is to demonstrate the coherent swing instabilities. Although different from the study by Susuki et al., the grid loop as described in their study presented a good opportunity to showcase the ability of TPWL method for the model order reduction and fast simulation of electrical power grids.

Training trajectories and reduced order models

In the current study, there are three different scenarios of initial conditions that must be taken into account for the training trajectories, these are listed in Table 3. The dependence of the trajectories on the initial conditions is evident since the trajectory will be different in each case.

As described in “Selection of Training Trajectories” section, multiple training trajectories have been employed for the generation of reduced order models. These training trajectories were categorized into three different types. The total time of simulation and the step size are listed in Tables 2 and 3 lists the perturbation amount and the node for each case.

1. Initial conditions at the equilibrium point for all the generators bar one.
2. All the generators start from a non-equilibrium point and one generator out of synchronicity.
3. All the generatros start from the same non-equilibrium state and are synchrononous.

Table 2 Training trajectories data

Symbol	Description	Value (s)
T	Total time of simulation	50
ΔT	Time step	0.005

Table 3 Initial conditions

Training trajectory	δ_2	$\delta_i \quad \forall i \neq 2$
Trajectory 1	1.45	1.25
Trajectory 2	1.1	1
Trajectory 3	0.8	0.8

The Jacobian is a tri-banded diagonal matrix in the case under study here.

$$\begin{aligned}
 J_{k,l} &= \frac{\partial p_k}{\partial \delta_l} = \frac{\partial}{\partial \delta_l} [b \sin(\delta_k) + b_{int} \sin(\delta_k + \delta_{k+1}) + b_{int} \sin(\delta_k - \delta_{k-1})] \\
 &= \hat{\delta}_{k,l} \{b \cos(\delta_k)\} + \{b_{int} \cos(\delta_k - \delta_{k+1})\}(\hat{\delta}_{k,l} - \hat{\delta}_{k+1,l}) \\
 &\quad + \{b_{int} \cos(\delta_k - \delta_{k-1})\}(\hat{\delta}_{k,l} - \hat{\delta}_{k-1,l})
 \end{aligned} \tag{27}$$

It is intuitive that the more the number of modes and the number of linearization points the more time the reduced model takes to simulate. The time consumption data along with number of modes and linearization points of the training trajectories is listed in Table 4. It is to be noted that all the simulations were run on a laptop with a i5-4200 CPU with a 1.6GHz processor and 8 GB RAM.

The comparison between the reduced linearized model from TPWL and the full non-linear model was carried out on the average δ , which is termed as collective-phase variable and its time derivate is ω . These are defined for loop power systems as:

$$\delta = \frac{1}{N} \sum_{i=1}^N \delta_i \tag{28}$$

$$\omega = \frac{d\delta}{dt} = \frac{1}{N} \sum_{i=1}^N \omega_i \tag{29}$$

The variables are well-known in power system stability analysis as the Center of Angle (COA) or Center of Inertia (COI) [33]. These variables demonstrate the collective dynamics of the system and are useful in the study of stability of the power grids.

The modes and the linearization points from the training trajectories were saved and then used with cases which are slightly different from the training cases. The results we obtained are encouraging for this kind of model order reduction for the nonlinear functions.

First test case: single node perturbation in non-equilibrium conditions

This is the case which is closely related to the second training trajectory. We have all the nodes starting from a non-equilibrium point $\delta_i = 1 \forall i \neq 2$ and in addition one node was perturbed by about 0.12 radians, i.e., $\delta_2 = 1.12$. The initial conditions used for this test case are presented in Table 5.

The results are very promising and the TPWL simulation is considerably faster as the full simulation in a similar case takes about 240 s while the TPWL simulation took about

Table 4 Time consumption data

Training trajectory	Modes 'q'	Lin Pts 's'	Time (s)		
			Full model	POD	Lin Pts Sel
1	312 ($\eta < 10^{-4}$)	26 ($\epsilon_s = 0.01$)	202	33	276
	312 ($\eta < 10^{-4}$)	47 ($\epsilon_s = 0.005$)			632
	203 ($\eta < 10^{-2}$)	47 ($\epsilon_s = 0.005$)			580
2	312 ($\eta < 10^{-4}$)	16 ($\epsilon_s = 0.005$)	191	27	155
	199 ($\eta < 10^{-2}$)	16 ($\epsilon_s = 0.005$)			150
	199 ($\eta < 10^{-2}$)	49 ($\epsilon_s = 0.001$)			850
3	130 ($\eta < 10^{-4}$)	8 ($\epsilon_s = 0.005$)	60	30	105
	62 ($\eta < 2 \times 10^{-2}$)	8 ($\epsilon_s = 0.005$)			100
	62 ($\eta < 2 \times 10^{-2}$)	17 ($\epsilon_s = 0.001$)			200

Table 5 Initial conditions for test cases

Test cases	δ_2	$\delta_i \quad \forall i \neq 2$
Test case 1	1.12	1
Test case 2	1.15	1.15

18 s with good accuracy. The errors are listed in the Table 6. The comparison of the average δ between the original and reduced order models are presented in Fig. 4.

Second test case: synchronous non-equilibrium

This is a case similar to the third training trajectory where we had all the nodes starting from a synchronous non-equilibrium position, in this case we gave the initial conditions of 1.15 rather than 0.8 in the training case $\delta_i = 1.15 \forall i$. The initial conditions used for this test case are presented in Table 5.

The results are excellent and the TPWL simulation is considerably faster as the full simulation in this case takes about 60 s while the TPWL simulation takes about 4 s with very high accuracy. The errors are listed in the Table 6 and the comparison of the average δ between the original and reduced order models are presented in Fig. 5.

Convergence analysis

It is important to analyse the source of error in the current study. As the results from the previous section shows that the error is small but not small enough as to compare with the machine precision level. In order to understand how does the error diminish and its dependency on various factors, we studied different factors namely, the number of modes, number of linearization points and the time step size. It is evident from the previous tables that the number of modes used has a little impact on error but considerable one on time

Table 6 Test cases of TPWL simulations

Test case	Modes 'q'	Lin Pts 's'	Time (s)		Error in δ	
			Full model	TPWL	Abs	Rel
1	199	49	240	18	8.75×10^{-4}	6.75×10^{-4}
2	62	17	60	4	6.62×10^{-4}	5.27×10^{-4}

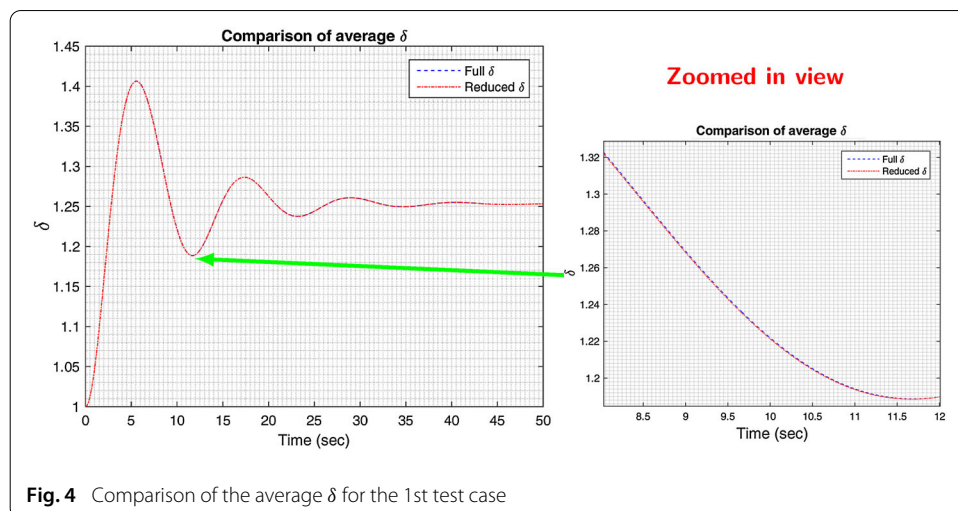
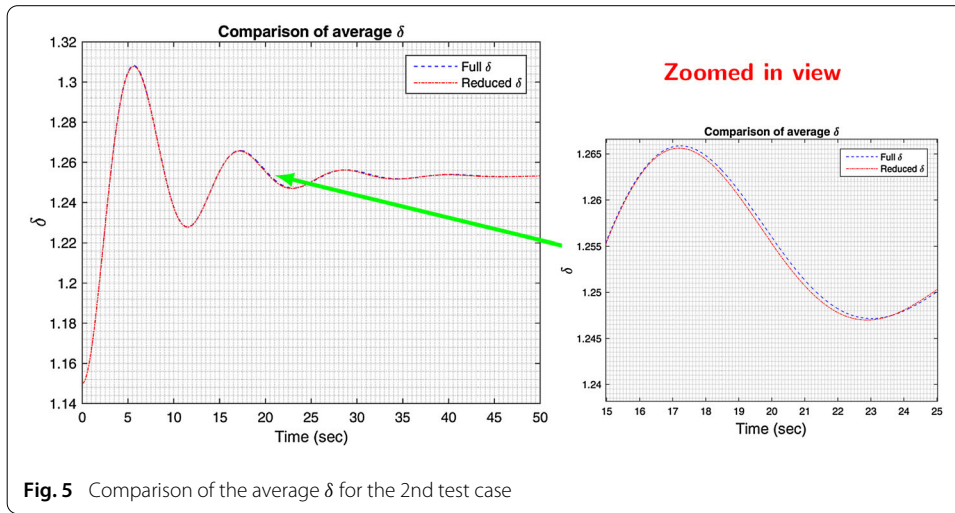
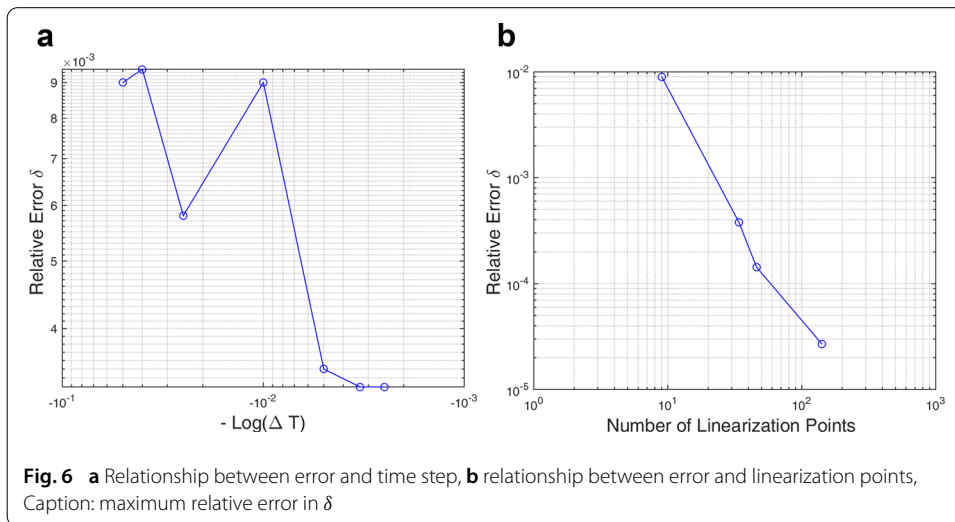


Fig. 4 Comparison of the average δ for the 1st test case



consumption during the simulation. The behavior of the maximum relative error in the reduced linearized variable δ with respect to the time step and the number of linearization points is shown in the Fig. 6a and b , respectively. The non-monotonic behavior of the relative error as a function of time step in Fig. 6a is due to the fact that for a larger time step compared to the one adopted for the training trajectory ($\Delta T = 0.005$) the reduced order model might skip some of the linearization points. Hence, it is inferred that the time step used for the reduced model simulation should be at least equal to the time step used in the training simulation. For smaller time steps the error is bounded by the truncation error of the reduced basis.

It can be conclusively said that the major impact on the accuracy of the reduced order linearized model depends on having more linearized point. This is quite intuitive, since adding more points around which the linearization of the non-linear function is performed it will be able to capture the non-linear behavior more effectively and hence reducing the error.



Confidence interval

In order to give an idea of how the reduced model generated using TPWL compares with the high fidelity model, we carried out a simple analysis similar to sensitivity analysis. For this analysis, all the parameters were kept constant bar one. This is repeated for all the parameters. This analysis is performed using reduced basis generated using one training trajectory. The details of the training trajectory are similar to as listed in Table 1 and the initial values of the phase angles are listed in Table 7. The number of linearization points (s) in the reduced basis for this case is 205 and the modes (q) selected are 199 out of 1000.

In Table 8 we presented the upper limit and the lower limit wherever applicable that we can use with confidence having a maximum relative error in $\{\delta\}$ of 5% or less. Due to the nature of the problem, the lower limit in the case of p^m and upper limit of d are open-ended as the error is always bounded on these limits. The number of generators N and the susceptances b are not included in this exercise because they depend upon the network topology and changes in these will have to be incorporated through a new reduced model. The mass of the generator is also not included since the equation is always normalized with the mass of the generator such that m_i is one. The variation in the phase angle δ_2 depends upon the initial condition, the performance of the reduced model simulation improves with the inclusion of more than one training trajectory and hence, the range for δ_2 increases from what is given in Table 7.

As we have described earlier that in this method several training trajectories can be included in the offline phase to make the reduced basis. This implies that for values over the limits in the Table 8, we can add another training trajectory so that the error remains acceptable.

Conclusions

TPWL proves to be a very robust method for model order reduction of models containing nonlinear functions. It has been proved as a fast, reliable and accurate MOR technique as observed from the results presented by the test cases in “Numerical experiments” section. The method as described is separated into offline and online phases, where in the offline phase the selection of linearization points is carried out. That is the only time consuming part of the method and as it is performed only once during offline phase the time penalty on the overall procedure is not severe.

Table 7 Initial conditions used in the build up for confidence interval

Trajectory for confidence interval	δ_2	$\delta_i \quad \forall i \neq 2$
Trajectory 1	1.5	1

Table 8 Confidence interval for parameters with training trajectory of Table 7

Parameter	Upper limit	Lower limit
p^m	0.97 (p.u.)	–
d_i	–	0.15 (p.u.)
δ_2	2.25	1
b_{int} (p.u.)	105	95

For the confidence interval, some other values of disturbances in $\{\delta\}$, and different values of p^m , d_i , b_{int} were trialed and the results as shown in “Confidence interval” section proves the robustness of the method considering the large variations possible for simulation with reduced model. Note that, the confidence interval’s simulation were performed with only one training trajectory and if more trajectories are included in the reduced basis the interval where the method can be applied increases and the results improves.

The method is very well adapted to the problem discussed in this study and more application for example, for the differential-algebraic equations (DAEs) of the network grids containing both generators (PV nodes) and the loads (PQ nodes), can further consolidate the current method as a well established model reduction method.

Authors’ contributions

All the authors contributed in the development and adaptation of the method for the current study. Malik performed the coding in MATLAB and drafted the manuscript. Borzacchiolo, Diez and Chinesta supervised the study and advised on the draft and the corrections in the manuscript. All authors read and approved the final manuscript.

Author details

¹Institut de Calcul Intensif (ICI), École Centrale de Nantes, 1 rue de la Noë, BP 92101, 44321 Nantes Cedex 3, France,

²Laboratori de càlcul numèric (LaCàN), Polytecnic University of Catalunya, C2 Campus Nord, Calle Jordi Girona, 31, 08034 Barcelona, Spain.

Competing interests

The authors declare that they have no competing interests.

Received: 18 July 2016 Accepted: 15 November 2016

Published online: 05 December 2016

References

- Magnusson PC. The transient-energy method of calculating stability. *Trans Am Inst Electr Eng.* 1947;66(1):747–55. doi:[10.1109/T-AIEE.1947.5059502](https://doi.org/10.1109/T-AIEE.1947.5059502).
- Aylett PD. The energy-integral criterion of transient stability limits of power systems. *Proc IEE Part C Monogr.* 1958;105(8):527–36. doi:[10.1049/pi-c.1958.0070](https://doi.org/10.1049/pi-c.1958.0070).
- Z. Bai PD, Freund R. Reduced-order modeling. In: Schilders EJW, Maten W, editors. *Handbook of numerical analysis Vol XIII, numerical methods in electromagnetics vol 13*, 2nd edn. North Holland : Elsevier; 2005. p. 825–95.
- Schilders W. Introduction to model order reduction. In: Schilders WHA, van der Vorst HA, Rommes J, editors. *Model order reduction: theory, research aspects and applications. Mathematics in industry*, vol 13. Berlin: Springer; 2008. p. 3–32. doi:[10.1007/978-3-540-78841-6](https://doi.org/10.1007/978-3-540-78841-6).
- Pinnau R. Model reduction via proper orthogonal decomposition. In: Schilders, WHA, van der Vorst HA, Rommes J, editors. *Model order reduction: theory, research aspects and applications. Mathematics in industry*, vol 13. Berlin: Springer; 2008. p. 95–109. doi:[10.1007/978-3-540-78841-6](https://doi.org/10.1007/978-3-540-78841-6).
- Chinesta F, Leygue A, Bordeu F, Aguado JV, Cueto E, Gonzalez D, Alfaro I, Ammar A, Huerta A. Pgd-based computational vademecum for efficient design, optimization and control. *Arch Comput Methods Eng.* 2013;20(1):31–59. doi:[10.1007/s11831-013-9080-x](https://doi.org/10.1007/s11831-013-9080-x).
- Parrilo PA, Lall S, Paganini F, Verghese GC, Lesieutre BC, Marsden JE. Model reduction for analysis of cascading failures in power systems. In: *American control conference, 1999. Proceedings of the 1999*, vol 6. 1999. p. 4208–42126. doi:[10.1109/ACC.1999.786351](https://doi.org/10.1109/ACC.1999.786351).
- Montier L, Henneron T, Clénet S, Goursaud B. Transient simulation of an electrical rotating machine achieved through model order reduction. *Adv Model Simul Eng Sci.* 2016;3(1):1–17. doi:[10.1186/s40323-016-0062-z](https://doi.org/10.1186/s40323-016-0062-z).
- Kashyap N, Werner S, Riihonen T, Huang Y-F. Reduced-order synchrophasor-assisted state estimation for smart grids. In: *Smart grid communications (SmartGridComm), 2012 IEEE third international conference.* 2012. p. 605–10. doi:[10.1109/SmartGridComm.2012.6486052](https://doi.org/10.1109/SmartGridComm.2012.6486052).
- Wille-Haussmann B, Link J, Wittwer C. Simulation study of a smart grid approach: Model reduction, reactive power control. In: *Innovative smart grid technologies conference Europe (ISGT Europe), 2010 IEEE PES.* 2010. p. 1–7. doi:[10.1109/ISGTEUROPE.2010.5638971](https://doi.org/10.1109/ISGTEUROPE.2010.5638971).
- Rewieński MJ. A trajectory piecewise-linear approach to model order reduction of nonlinear dynamical systems. PhD thesis, Massachusetts Institute of Technology. 2003.
- Liu Y, Yuan W, Chang H. A global maximum error controller-based method for linearization point selection in trajectory piecewise-linear model order reduction. *IEEE Trans Comp Aided Des Integr Circuits Syst.* 2014;33(7):1100–4. doi:[10.1109/TCAD.2014.2307000](https://doi.org/10.1109/TCAD.2014.2307000).
- Chen Y, White J. A quadratic method for nonlinear model order reduction. In: *Technical proceedings of the 2000 international conference on modeling and simulation of microsystems.* 2000. p. 477–80. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.19.8951>.
- Vasilyev D, Rewieński M, White J. A tbr-based trajectory piecewise-linear algorithm for generating accurate low-order models for nonlinear analog circuits and mems. In: *Design automation conference, 2003. Proceedings.* 2003; p. 490–5. doi:[10.1109/DAC.2003.1219055](https://doi.org/10.1109/DAC.2003.1219055).

15. Qu L, Chapman PL. A trajectory piecewise-linear approach to model order reduction for nonlinear stationary magnetic devices. In: Computers in power electronics, 2004. Proceedings. 2004 IEEE workshop. 2004; p. 15–19. doi:[10.1109/CIPE.2004.1428113](https://doi.org/10.1109/CIPE.2004.1428113).
16. Rewiński M, White J. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. *IEEE Trans Comp Aided Des Integr Circuits Syst.* 2003;22(2):155–70. doi:[10.1109/TCAD.2002.806601](https://doi.org/10.1109/TCAD.2002.806601).
17. Farooq MU, Xia L, Hussin FAB, Malik AS. Automated model generation of analog circuits through modified trajectory piecewise linear approach with Chebyshev Newton interpolating polynomials. In: Intelligent systems modelling simulation (ISMS), 2013 4th international conference. 2013; p. 605–9. doi:[10.1109/ISMS.2013.28](https://doi.org/10.1109/ISMS.2013.28).
18. Long C, Simonson LJ, Liao W, He L. Floorplanning optimization with trajectory piecewise-linear model for pipelined interconnects. In: Proceedings of the 41st annual design automation conference. 2004. p. 640–5.
19. Farooq MU, Xia L. Local approximation improvement of trajectory piecewise linear macromodels through chebyshev interpolating polynomials. In: Design automation conference (ASP-DAC), 2013 18th Asia and South Pacific. 2013. p. 767–72. doi:[10.1109/ASPDAC.2013.6509693](https://doi.org/10.1109/ASPDAC.2013.6509693).
20. Burgard S, Farle O, Klis D, Dyczij-Edlinger R. Order-reduction of fields-level models with affine and non-affine parameters by interpolation of subspaces. *IFAC PapersOnLine.* 2015;48(1):170–5. doi:[10.1016/j.ifacol.2015.05.111](https://doi.org/10.1016/j.ifacol.2015.05.111).
21. Panzer H, Mohring J, Eid R, Lohmann B. Parametric model order reduction by matrix interpolation. *Automatisierungstechnik.* 2010;58(8):475–84. doi:[10.1524/aut.2010.0863](https://doi.org/10.1524/aut.2010.0863).
22. Xie W, Theodoropoulos C. An off-line model reduction-based technique for on-line linear MPC applications for nonlinear large-scale distributed systems. In: Pierucci S, Ferraris GB, editors. In: 20th European symposium on computer aided process engineering. Computer aided chemical engineering, vol 28, . Amsterdam: Elsevier; 2010. p. 409–14. doi:[10.1016/S1570-7946\(10\)28069-0](https://doi.org/10.1016/S1570-7946(10)28069-0). <http://www.sciencedirect.com/science/article/pii/S1570794610280690>.
23. He J, Sætrom J, Durlofsky LJ. Enhanced linearized reduced-order models for subsurface flow simulation. *J Comput Phys.* 2011;230(23):8313–41. doi:[10.1016/j.jcp.2011.06.007](https://doi.org/10.1016/j.jcp.2011.06.007).
24. Cardoso MA, Durlofsky LJ. Linearized reduced-order models for subsurface flow simulation. *J Comput Phys.* 2010;229(3):681–700. doi:[10.1016/j.jcp.2009.10.004](https://doi.org/10.1016/j.jcp.2009.10.004).
25. Bergen AR, Vittal V. Power system analysis. 2nd ed. New Jersey: Prentice Hall; 1999.
26. Shampine LF, Reichelt MW. The MATLAB ODE suite. *SIAM J Sci Comput.* 1997;18:1–22. doi:[10.1214/aoms/1177729959](https://doi.org/10.1214/aoms/1177729959).
27. Liang YC, Lee HP, Lim SP, Lin WZ, Lee KH, Wu CG. Proper orthogonal decomposition and its applications—part I: theory. *J Sound Vib.* 2002;252(3):527–44. doi:[10.1006/jsvi.2001.4041](https://doi.org/10.1006/jsvi.2001.4041).
28. Kerschen G, Golinval J-C, Vakakis AF, Bergman LA. The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview. *Nonlinear Dyn.* 2005;41(1–3):147–69. doi:[10.1007/s11071-005-2803-2](https://doi.org/10.1007/s11071-005-2803-2).
29. Berkooz G, Holmes P, Lumley JL. The proper orthogonal decomposition in the analysis of turbulent flows. *Annu Rev Fluid Mech.* 1993;25(1):539–75. doi:[10.1146/annurev.fl.25.010193.002543](https://doi.org/10.1146/annurev.fl.25.010193.002543).
30. Alburni MN. Model order reduction of moving nonlinear electromagnetic devices. PhD thesis, Technical University of Munich. 2010.
31. Rewiński M, White J. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. In: Computer aided design, 2001. ICCAD 2001. In: IEEE/ACM international conference on. 2001. p. 252–7. doi:[10.1109/ICCAD.2001.968627](https://doi.org/10.1109/ICCAD.2001.968627).
32. Acha E, Agelidis VG, Anaya-Lara O, Miller TJE. 4 - power flows in compensation and control studies. In: Acha E, Agelidis VG, Anaya-Lara O, Miller TJE, editors. Power electronic control in electrical systems. Newnes Power Engineering Series. Oxford: Newnes; 2002. p. 106–52. doi:[10.1016/B978-075065126-4/50004-3](https://doi.org/10.1016/B978-075065126-4/50004-3). <http://www.sciencedirect.com/science/article/pii/B9780750651264500043>.
33. Susuki Y, Mezić I, Hikiyama T. Coherent swing instability of power grids. *J Nonlinear Sci.* 2011;21(3):403–39. doi:[10.1007/s00332-010-9087-5](https://doi.org/10.1007/s00332-010-9087-5).

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com
