Nonlinearity-aware sub-model combination in trajectory based methods for nonlinear Mor

S.A.Nahvi¹,∗, M.Nabi¹, S.Janardhanan¹

Electrical Engineering Department, IIT Delhi, New Delhi, India 110016

Abstract

Trajectory based methods approximate nonlinear dynamical systems by superposition of dimensionally reduced linear systems. The linear systems are obtained by linearisations at multiple points along a state-trajectory. They are combined in a weighted sum and the combinations are switched appropriately to approximate the dynamic behaviour of the nonlinear system. Weights assigned at a specimen point on the trajectory generally depend on the euclidean distance to the linearisation points. In this work, limitations of the conventional weight-assignment scheme are pointed out. It is shown that the procedure is similar across all nonlinearities, and hence ignores the nonlinear vector field curvature for superposition. Additionally, it results in an inadequate assessment of the linear systems when they are equidistant from the specimen point. An improved method for weight-assignment, which uses state-velocities in addition to state-positions is proposed. The method naturally takes into account the system nonlinearity and is hence called Nonlinearity-aware Trajectory Piece-wise Linear (Ntpwl) method. Further, a computationally efficient procedure for estimating the state-velocity is introduced. The new strategy is illustrated and assessed with the help of case studies and it is shown that the Ntpwl model substantially improves the approximation of the nonlinear systems considered. Increased robustness to training and negligible stretching of the computational resources is also obtained.

Keywords: Large dynamical systems, Model order reduction, Nonlinear systems, Trajectory piecewise linear, Weight-assignment

1. Introduction

Dynamical systems are used to model many physical and artificial processes. Realistic and accurate description of such systems tend to be of large dimensions, and their simulation is not possible without expenditure of considerable amounts of computational resources and time. Approximation is thus crucial for cost-effective simulation. Model Order Reduction (Mor) techniques help realize this objective in a computationally efficient way. Mor results in a Reduced order model (Rom) with input-output mapping similar to the large-scale system it approximates.

*Corresponding author

Email addresses: san@ee.iitd.ac.in (S.A.Nahvi), mnabi@ee.iitd.ac.in (M.Nabi), janas@ee.iitd.ac.in (S.Janardhanan)
Recently developed Mor techniques for Lti systems can be considered to be Projection based. They construct a Rom of order $r \ll n$ that approximates the original system from a subspace spanned by a reduced basis of dimension $r$ in $\mathbb{R}^n$. Broadly, the Projection based methods can be classified into two categories – Krylov subspace projection and methods based on Truncated balanced realizations (Tbr).

Krylov based methods provide a numerically robust and cheap alternative for constructing the dominant subspace. Originally introduced in [23, 25] they were proposed as an alternative to the numerically ill-conditioned, explicit moment matching technique called Asymptotic Waveform Evaluation (Awe) [37]. These methods have found widespread applications in areas as diverse as Machine tool simulation [22], coupled field-circuit problems [33] and Micro-Electromechanical Systems (Mems) [7]. The drawbacks are that the resulting reduced order systems generally have no guaranteed error bound, the procedure is not automatic and features like stability and passivity are not retained. However, modified algorithms like Prima [35] (Passive reduced-order interconnect modelling algorithm) that exploit system structure to preserve passivity have been developed.

Tbr based methods, first described in [32], require the solution of Lyapunov equations for the computation of controllability and observability grammians $W_c$ and $W_o$. The Rom is obtained by a projection of the original system onto the dominant Eigen-space of the product $W_c W_o$. Although the exact solution of Lyapunov equations requires dense computations limiting the application of these methods to systems of dimension in thousands [4], modifications of these methods which reduce computational complexity [3, 27] have also been developed. Tbr based methods have a number of desirable properties, a global error bound exists and stability is preserved.

Recent thrust-areas in Mor methods for Lti systems include combining Krylov and Tbr based methods [26], structure preservation in Roms [24, 29] and Parametric Mor (Pmor) [19].

Mor of nonlinear systems is an area in its infancy [42]. However, including model nonlinearities in simulation and control is becoming increasingly important with the thrust in areas of mixed-signal design like Mems. Some of the techniques which have been applied successfully for Mor of nonlinear systems include, local approximations of the system nonlinearity, Proper orthogonal decomposition (Pod) and methods based on approximations on the nonlinear system trajectory. Nonlinearities have been locally approximated by linear or polynomial expansion [17], bi-linearisation [6], or Volterra series expansion [36]. This is followed by projection. However, these methods generate Roms valid only around one operating point of the system, and can be applied only to weakly nonlinear systems and small input disturbances.

POD was developed independently by several scientists [9], and has been one of the most popular tools for nonlinear model reduction, with varied applications. It obtains the projection basis from the time domain simulations of the nonlinear dynamical system. In addition to the obvious drawback that the nonlinear system has to be simulated, the cost of evaluating the projected nonlinearity remains high. This is because it requires operations of computational complexity dependant on dimension $n$ of the original system [16], and is unlike the Lti case wherein reducing $n$ reduces complexity. Hence the pertinent observation that model complexity is not strictly tied to the model order in the nonlinear case [13]. Efforts made for solving this problem include, Missing point estimation (Mpe) [5] and Discrete empirical interpolation method (Deim) [16]. Both of these methods compute the projection over a subset of the spatial domain, hence reducing the computational cost of evaluating the projected nonlinearity. There have been additional extensions, modifications and attempts to combine Pod with other techniques. The method of empirical balanced realization [28] combines the features of Pod and Tbr. This method has been extended and modified in other works like [18, 50].
A relatively recent development are Trajectory based methods, wherein the nonlinear system is approximated by superposition of dimensionally reduced linear systems. The linear systems are obtained by linearisations at points spread across a nonlinear system trajectory and each of them is subsequently reduced. The reduced order representation of the nonlinear system at intermediate points is given by the weighted sum of these reduced linear systems, henceforth referred to as sub-models. Since, instead of the original nonlinear system, linear systems are projected onto the dominant subspace, the computational cost of projection and evaluation of the Rom becomes comparable to that of projecting an LTI system, hence offering considerable computational savings as compared to POD. Further, this method also offsets the drawback of local linearisations and creates a model with improved fidelity, as long as the nonlinear system stays near training trajectory. The original formulation is called the Trajectory piecewise-linear (Tpwl) method [40]. Shortly after its introduction, Tpwl was found to hold promise, and was quickly picked up by researchers for application. There are many instances of successful application of the Tpwl technique to various types of nonlinear systems, e.g., Circuits [8, 20], Computational fluid dynamics [15], Power Electronics [38], Nonlinear Electromagnetic devices [1] and Mems [48, 51].

In addition to applications, there have been efforts at improving Tpwl. Using different schemes for dominant subspace extraction [30, 47], employing multiple training trajectories [45, 46], improving the small signal response of the Tpwl model [21], extending the scheme to nonlinear parameterized models [10, 11] and ensuring stability of the Tpwl model [12] are some of them. For a detailed review of the research directions in which Tpwl has evolved see [34].

Our endeavour in this work, is to identify the limitations of the conventional superposition scheme and suggest an alternative strategy that address them. In the conventional scheme, at an intermediate point on the state trajectory, weights to sub-models are assigned based on their ‘closeness’ to it. The ‘closeness’ measure is generally the euclidean distance between the intermediate point to the linearisation points. However, it is shown here that this scheme is insensitive to the nonlinearity involved, although it is conceptually and computationally simple. Further, if an intermediate point is approximately equidistant from multiple linearisation points and this ‘closeness’ measure alone fails to suggest decisive choices of weights.

In this work, we propose assigning weights on the basis of difference in state-velocities as an additional criteria for assessment of the sub-models. This leads to a useful discrimination amongst the sub-models and naturally takes into account the nonlinear vector field curvature for superposition, making the process nonlinearity-aware. A computationally efficient procedure is introduced for making the state-velocity estimation viable. The new method, hence called nonlinearity-aware Tpwl, or Ntpwl, leads to a significant improvement in the reproduction of the nonlinear system dynamics with little increase in the simulation time.

In the subsequent section we briefly review the Tpwl method and point out the limitations of the conventional weight-assignment strategy. Ntpwl is introduced and formulated in Section-3, and the complete Mor strategy using it is given in Section-4. Finally, to demonstrate the efficacy of the new method we give detailed case studies in Section-5, which is followed by conclusion.

2. Review of the Tpwl method

In Tpwl the first step is to find the nonlinear system trajectory in response to a training input, the trajectory maybe exact or approximate. Appropriate points, called Linearisation Points (Lp’s) are then selected on the trajectory for linearising the nonlinear system. The linear systems hence
formed, are then dimensionally-reduced, generally using Krylov subspace projection. This step
gives the sub-models at all Lp’s which are combined in a weighted sum to give a reduced-order
piece-wise linear approximation of the nonlinear system. The complete idea can be mathemati-
cally expressed as follows.

Consider the nonlinear dynamical system in the following state space form

\[ \dot{x} = f(x) + Bu \]
\[ y = Cx \]

(1)

where \( x \in \mathbb{R}^n \) is a vector of system states, \( f: \mathbb{R}^n \rightarrow \mathbb{R}^n \) is the nonlinear vector field, \( B \in \mathbb{R}^{n \times p} \) is the input matrix, \( C \in \mathbb{R}^{n \times q} \) is the output matrix and \( y \in \mathbb{R}^q \) is the output. The treatment in this
paper is restricted to systems in the input-affine form given in (1), which describes [40] a large
class of nonlinear dynamical systems.

Assuming \( m \) Lp’s \((x_0, \ldots, x_{i}, \ldots, x_{m-1})\) are selected along a training trajectory, the Taylor series
first-order approximation of \( f(x) \) about \( x_i \) is given by

\[ \tilde{f}(x) = f(x_i) + A_i(x - x_i) \]

(2)

Where \( A_i \) is the Jacobian of \( f(x) \) evaluated at \( x_i \). The linear system at \( x_i \) is hence given by:

\[ \dot{x} = f(x_i) + A_i(x - x_i) + Bu \]
\[ y = Cx \]

(3)

The dynamics are restricted to the dominant subspace by the state transformation \( x = Vz \). \( V \) is the
projection matrix with its columns spanning the dominant subspace of dimension \( r \), where \( r \ll n \) and \( V^TV = I \). With this, the reduced sub-model at \( x_i \) is:

\[ \dot{z} = A_{ir}z + V^T(f(x_i) - A_ix_i) + B_ru \]
\[ y = Crz \]

(4)

where \( A_{ir} = V^TA_iV, B_r = V^TB \), and \( C_r = CV \). The final Tpwl model is expressed as the weighted
sum:

\[ \dot{z} = \sum_{i=0}^{m-1} w_i(z)(A_{ir}z + V^T(f(x_i) - A_ix_i)) + B_ru \]
\[ y = Crz \]

(5)

Since each sub-model (4) is a good local approximation of the nonlinear system, the Tpwl model
(5) is expected to be a good approximation of the nonlinear system for that part of the state-space
which has been covered by the training trajectory. It is also expected that the Tpwl model would
remain a valid approximation for any other input, called the evaluation input, which does make
the nonlinear system leave this region.

Weight \( w_i(z) \) assigned to the sub-model at \( x_i \) at the intermediate point \( x (x = V^Tz) \) on the
trajectory is a function of the distance \( \|z - z_i\| \) in the reduced space. The weighing functions
used are hard switching functions. Weights with smoother profiles [20, 45] and time dependant
weights [2, 49] have also been used. Recent works [31, 43, 44] have pointed towards the need
for development of better weight-assignment strategies.

2.1. Limitations of the conventional weight assignment approach

It has been recognized that sub-model combination strategies should be improved upon [43,
p. 300] to address inadequacies of the Tpwl method, but their exact limitations have not been
clearly spelled out. In this section we try to give a deeper insight into the limitations of the conventional scheme and try build a foundation for the subsequent introduction of a new strategy.

1) **Ignoring the nonlinear vector field curvature**: For dynamical systems governed by (1), the error \( e(x) \) in the approximation (2) is:

\[
e(x) = f(x) - \tilde{f}(x)
\]

The norm of \( e(x) \) in this approximation is bounded by Taylor’s inequality:

\[
\|e(x)\| \leq \frac{1}{2} \sup_{x \in B(x_i, \epsilon)} \|W(x)\| \|x - x_i\|^2
\]

where \( W(x) \) is the system Hessian and \( \sup_{x \in B(x_i, \epsilon)} \|W(x)\| \) is calculated in the interval between \( x_i \) and \( x \). In [39, p. 40], based on the bound on the error growth (7) weights are assigned proportional to \( \|x - x_i\| \) (and in turn proportional to \( \|z - z_i\| \) if \( x_i \in colspan(V) \) [39, p. 41]). The evident drawback of this method is that it ignores the contribution of \( W(x) \) in deciding the error growth and is thus only partially based on (7). Hence it also implies that the curvature of the nonlinear vector field \( f(x) \) is ignored for calculating weights. The result is that even with sharply switching weights this results in an equal weight assignment, especially when the distances \( \|x - x_i\| \) are comparable.

In a similar vein in [41] it is claimed that a linearisation of \( f \) at \( x_i \) accurately approximates the nonlinear function at \( x \), provided \( \|x - x_i\| < \epsilon \), i.e., \( x \) is ‘close enough’ to \( x_i \). The point missed here is that \( \epsilon \) would not be the same at all \( L_p \)’s, its magnitude would be strongly decided by quality of approximation of \( f \) at \( x_i \). Hence \( x \) maybe equidistant from two sub-models, but one of the two may be a better approximation for \( f(x) \), and assigning equal weights to the two may not be correct.

This problem in Tpwl though recognized, has not been addressed well. The authors in [39, p. 43] suggested computing \( \|W(x)\| \) at each \( L_p \) and then taking scaled distances given by

\[
d_i = \|W(x_i)\|^{1/2} \|z - z_i\|
\]

for weight assignment. But, the exact computation of \( \|W\| \) is not possible unless its analytical expression is known, and even if it is known somehow, the evaluation at all \( L_p \)’s is expensive and increases computational burden. Secondly, in case \( \|W\| \) is estimated numerically at all \( L_p \)’s, the weighing strategy would be based on the assumption that \( \sup_{x \in B(x_i, \epsilon)} \|W(x)\| = \|W(x_i)\| \), which may not be justifiable.

2) **Inadequate discrimination between linear sub-models**: There are instances when the sub-models obtained are such that during evaluation of the reduced order representation at intermediate points, the distances to all or most of the \( L_p \)’s are almost equal. The result is that it is not possible to appraise the linear sub-models correctly and this leads to an equal weight distribution and significant errors in reproduction of the nonlinear system behaviour. The errors are especially pronounced in the region where the discrimination between the linear sub-models is not adequate, as shown later in Section-5.1. It seems that an additional criteria to judge the sub-models may solve the problem. We show in Section-3 that state velocities at \( L_p \)’s are a viable alternative, and they also result in an incorporation of system nonlinearity into the superposition scheme.

3. **Nonlinearity-aware Tpwl**

3.1. **Dynamics inspired weight-assignment**

From a dynamical system viewpoint, assessing similarities between linearisations based on the nearness of both their state \( x \) as well as simultaneous nearness of their state-velocities \( \dot{x} \) seems
more logical. This is obvious since the characteristics of the dynamical system are present in the relationship between the state \( x \) and its velocity \( \dot{x} \) at any point. Further, \( \dot{x} \) carries significant information about the system [14]. Consequently two linear systems created out of linearisations at \( x_1 \) and \( x_2 \) can be said to be ‘close’ if \( \|x_1 - x_2\| < \epsilon \) and \( \|\dot{x}_{x=x_1} - \dot{x}_{x=x_2}\| < \delta \), i.e., \( x_1 \) is close to \( x_2 \) and \( \dot{x}_{x=x_1} \) is close to \( \dot{x}_{x=x_2} \). Besides, if two linearisations are called ‘close’ because the Lp’s at which they have been created are close, then their dynamic behaviour at Lp’s (represented by \( \dot{x} \)) should also be similar.

Based on this observation, we propose that sub-models be appraised on the basis of state-distances as well as state-velocities. We show next that this proposal is consistent with a new error-growth bound.

A simple rewriting of the error definition (6) yields

\[
e(x) = f(x) - \tilde{f}(x) = f(x) - (f(x_i) + A_i(x - x_i))
\]  

By simple re-arrangements and using standard inequalities we have:

\[
e(x) = (f(x) - f(x_i)) - A_i(x - x_i)
\]

\[
\|e(x)\| = \|(f(x) - f(x_i)) - A_i(x - x_i)\|
\]

\[
\|e(x)\| \leq \|(f(x) - f(x_i))\| + \|A_i(x - x_i)\|
\]

Hence the new bound on error growth:

\[
\|e(x)\| \leq \|(f(x) - f(x_i))\| + \|A_i\|\|(x - x_i)\|\]

Since \( |A| \) is constant, it is clear from (10) that the error in the approximation of \( f(x) \) by the linear system at \( x_i \) is bounded by the increase in the differences of the nonlinear vector fields \( \|(f(x) - f(x_i))\| \) and the states \( \|x - x_i\| \). The vector field \( f(x) \) at \( x \) is equal to the self-dynamic component of the state velocity. Hence together, these differences can be used for weight assignment in the form suggested by (10). It is pertinent to mention here that this treatment is analogous to (7), because it is also based on an error-growth bound. Further, unlike the previous case wherein the error-growth bound relationship (7) is considered only partially, all components that lead to the growth of error in (10) are considered for assigning weights.

The new criteria is stronger than the previous one on two counts. One, the sub-models are appraised on the basis of states-distances as well as state-velocities, which leads to useful discrimination when the sub-models are equidistant from the evaluation point, a claim that we substantiate in Section-5. Second, the new criteria, unlike the previous one, is not based partially on an error-growth bound, but we have shown that it is consistent completely with a new error-growth bound. The additional advantage is that since \( \dot{x} \) carries information about the nonlinearity, the new criteria makes the weight-assignment scheme non-linearity aware.

There are two important observations though. First, the state-velocities at Lp’s being input dependant can make the new criteria sensitive to input variations, which is not wanted, especially when the training and evaluation inputs are different. Since the nonlinear system in question allows an easy separation of the self-dynamics (\( \dot{x} = f(x) \)), this problem is resolved because only the self-dynamic component of the state velocities is considered. Secondly, it may seem that the new criteria may give higher weights to systems which are ‘away’ in the state-sense but ‘close’ in the velocity-sense. Now, large-dissimilar systems have a small chance of having similar state velocities, because it implies that for say a 100-dimension system, all the components of \( \dot{x} \) are approximately equal, which is not very probable. And even if such a thing happens, it can be taken care of, by making weights zero for systems at distances beyond a threshold, irrespective of state-velocities.
3.2. Composite participation measure

The evaluation of the sub-models at \( x_i \) that best approximate the non-linear system at \( x \), has been made to depend on two criteria, and (10) shows how the two can be combined. Now \( \|x - x_i\| \) can be replaced by \( \|z - z_i\| \) as shown in Section-2.1, and \( f(x_i) \) is known at \( L_p \)’s \( x_i \). \( f(x) \) which represents the non-linear vector field at the evaluation point \( x \) being unknown, it is replaced by an approximation \( \hat{f}(x) \). Based on these observations we introduce the concept of a composite participation measure for assigning weights to the sub-models. At an evaluation point \( z = V^T x \) in the reduced space, the participation measure \( p_i \) of the sub-model formed at \( x_i \) is defined as:

\[
  p_i = \eta d_i + (1 - \eta)v_i \tag{11}
\]

where \( d_i \) and \( v_i \) are normalised differences given by:

\[
  d_i = (\|z - z_i\|)/ \sum_{i=1}^{m-1} (\|z - z_i\|) \quad v_i = (\|\hat{f}(x) - f(x_i)\|)/ \sum_{i=1}^{m-1} (\|\hat{f}(x) - f(x_i)\|) \tag{12}
\]

Here \( \eta \) decides the relative importance of the two criteria, and can vary depending on the application. Normally giving equal importance to the two criteria (\( \eta = 0.5 \)) works well, as we show in results shown in Section-5. We show next that \( \hat{f}(x) \) which is the approximation of \( f \) at \( x \) can be obtained in a computationally inexpensive manner.

3.3. Evaluating \( \hat{f}(x) \)

The numerical approximation of \( f(x) \) is based on the premise that just as the evolution of \( x \) can be reconstructed from the evolution of \( z \), an estimate of \( f(x) \) can be also obtained from the same information. Consider the Tpwl model given by (5), it can be re-written as:

\[
  \dot{z} = \hat{A}z + Bu + \hat{K} \quad y = Cz \tag{13}
\]

where \( \hat{A} = \sum_{i=1}^{m-1} w_i(z)(A_{x_i}) \) and \( \hat{K} = \sum_{i=0}^{m-1} w_i(z)(V^T(f(x_i) - A_{x_i})) \). Since the dynamics are restricted by \( x \approx Vz \), which means \( z \approx V_x \), from (1) and (13) we have:

\[
  f(x) + Bu \approx V(\hat{A}z + Bu + \hat{K}) \tag{14}
\]

This implies that

\[
  f(x) \approx V(\hat{A}z + \hat{K}) + VV^T Bu - Bu \tag{15}
\]

If \( B \in \text{colspan}(V) \) we can write \( B = VR \) for some \( R \), and pre-multiplying by \( VV^T \) we have,

\[
  VV^T B = VV^T(VR) = V(V^T V)R
\]

\[
  VV^T B = VR
\]

\[
  VV^T B = B
\]

Using this in (15) yields:

\[
  f(x) \approx V(\hat{A}z + \hat{K}) = \hat{f}(x) \tag{16}
\]

Thus an estimate of the non-linear function \( f(x) \) can be made throughout the trajectory while simulating the Tpwl model which is of dimension \( r \), where \( r \ll n \). This estimate (16) involves computations in the reduced space and only one matrix-vector multiplication, and thus does not tax the computational resources at all. However, it requires that \( B \in \text{colspan}(V) \). This
condition can be easily ensured at the time when \( V \) is being evaluated, as shown in Algorithm-2. Finding \( \hat{f}(x) \) and evaluating the participation measure \( (11) \) leads to a minor increase in the simulation time, but the new algorithm leads to a considerable improvement in results, as reported in Section-5.

We want to remark here that \( (16) \) and \( (11) \) are used for assigning weights so that at a particular time-step \( t_j \) and state \( z(t_j) \), the Tpwl model is evaluated, i.e., \( \hat{A} \) and \( \hat{K} \) are found. But \( (16) \) requires a-priori knowledge of \( \hat{A} \) and \( \hat{K} \). We use \( \hat{A} \) and \( \hat{K} \) evaluated at \( t_{j-1} \) in \( (16) \), based on the assumption that the Tpwl model would not change significantly in one time step.

4. Overall Mor strategy using Ntpwl

Lp selection is done using Algorithm-1. The criteria for selecting \( \delta \) is unmodified from [40]. However, the nonlinear vector field magnitudes \( f(x_i) \) obtained during Lp selection are stored separately (They are subsequently used for evaluation of the participation measure in \( (11) \)).

\textbf{Algorithm 1 Lp Selection}

1: \( i \leftarrow 0, \ j \leftarrow 1 \). \( x_0 \) is the initial state, \( T \) the number of simulation time-steps and \( \delta \) is an appropriately selected constant.
2: \textbf{while} \( j < T \) \textbf{do} \hspace{1cm} \textbf{Compute } \( z \) for \( t = j \) till \( Vz_j \) is close to any of the Lp’s.
3: \textbf{repeat}
4: \hspace{1cm} Simulate \((4)\)
5: \hspace{1cm} \textbf{until} \( \min_{0 \leq k \leq i} \left\| \frac{||Vz_j - x_k||}{||x_k||} \right\| \leq \delta \)
6: \hspace{1cm} \( j \leftarrow j + 1 \)
7: \textbf{end while}
8: \( i \leftarrow i + 1 \). \( x_i = Vz_j \) is taken as the next Lp.

The projection basis \( V \) is obtained using the linearisation at \( x_0 \) as in [40]. However, any other method for Mor of Lti systems, some of which were briefly reviewed in the Introduction, can be used for finding \( V \). The pseudo-code is given in Algorithm-2. By including \( B \) in \( V \) in step-5, it is ensured that \( B \in \text{colspan}(V) \).

\textbf{Algorithm 2 Finding the projection basis } \( V \)

1: \( \dot{x} = f(x_0) + A_0(x - x_0) + Bu, \ y = Cx \) \hspace{1cm} \textbf{Linear System at } \( x_0 \).
2: If \( K_p \) is the \( p \)-th order Krylov subspace for this system, we have:
3: \( \text{span}(V_1) = K_p[A_0^{-1}, A_0^{-1}B] \)
4: \( \text{span}(V_2) = K_p[A_0^{-1}, A_0^{-1}(f(x_0) - A_0x_0)] \)
5: \( V = [V_1, V_2, x_0, B] \)
6: Orthogonalize the columns of \( V \) using Svd, and retain columns corresponding to singular values larger than some \( \epsilon \).

The pseudo-code for simulating the Ntpwl model is given in Algorithm-3.
Algorithm 3 Simulating the Ntpwl model

1: For $i = (0, 1, \ldots, m - 1)$ obtain $L_p$'s $x_i$ and the corresponding sub-models at $x_i$
   \[ z = A_r z + V^T (f(x_i) - A_i x_i)) + B_i u \\
   y = C_r z \quad (17) \]

2: Time-steps $k$ with $0 \leq k \leq T$; Initial conditions $x = x_0$ (translated as $z = z_0$)
3: $k \leftarrow 0$
4: The Rom at $z_k$ is the sub-model at $x_0$ (given by (17) for $i = 0$). \[ \triangleright \text{Assume that } x_0 \text{ is an } L_p \]
5: while $k \leq T$ do
6: Simulate the Rom at $z_k$ for one time step, i.e., evaluate $z_{k+1}$.
7: Obtain $\hat{f}(x_{k+1})$ (16) using $\hat{A}$ and $\hat{K}$ at $k$.
8: for $i = 0, 1, \ldots, m - 1$ do
9: Find the performance measure $p_i$ for each sub-model at $z_{k+1}$ using (11).
10: end for
11: $q = \min_{i(0,1, \ldots, m-1)} p_i$
12: for $i = 0, 1, \ldots, m - 1$ do
13: Compute $\tilde{w}_i = e^{-\beta p_i/q}$ \[ \triangleright \beta \text{ decides weight variation and is preselected as in [40].} \]
14: if $q = 0$ (because some $p_j = 0$) then $w_j = 1$ and $w_i = 0$ for $i \neq j$.
15: end if
16: end for
17: Normalize $\tilde{w}_i$. $w_i = \tilde{w}_i / \sum_{j=0}^{m-1} \tilde{w}_j$
18: end for
19: Find the Rom at $z_{k+1}$. \[ \triangleright \text{As a weighted sum of sub-models, use (5).} \]
20: $k \leftarrow k + 1.$
21: end while

![Nonlinear Transmission line model](image)

Figure 1: Nonlinear Transmission line model

5. Case studies

Three nonlinear circuits, a transmission line model, an RC-ladder circuit and a Memswitch that serve as benchmark models for Tpwl are used for illustrating and assessing Ntpwl.

5.1. Case-1: Nonlinear transmission line model

The circuit is shown in Fig. 1. It consists of resistors, capacitors and diodes. The constitutive equation for the diodes is $i_d(v) = e^{40v} - 1$. We have taken assumed unit resistance and capacitance,
so $R = 1\Omega, C = 1F$. Input $u(t)$ is the current source at node-1, $i(t)$, and output is the voltage at node-1, $y(t) = v_1(t)$. For the complete description of the model refer to [17].

Transmission line, Test-1: As pointed out in Section-2.1, sometimes the standard Tpwl strategy fails to give decisive weights to the sub-models. This is illustrated for the transmission line model of size $n = 100$. Lp’s are selected using Algorithm-1 with the input $u(t)$ being the Heaviside function $H(t)$. The order of the reduced system selected is $r = 15$. The nonlinear system state-trajectory is then obtained for the evaluation input $u(t) = 0.5(1 + \cos(\pi t/5))$. The distances of the Lp’s from the state-trajectory are plotted in Fig. 2 for 5 linear submodels. It is seen that close to $t = 4s$ the distances between the nonlinear system state and the 5 Lp’s are almost equal. Increasing the number of linearisations does not solve the problem as shown in Fig. 3 for 15 Lp’s. The fact that the sub-models are not appraised adequately is also obvious when these sub-models are used in Tpwl and their weight-distribution is plotted, as shown in Test-2.

![Figure 2: Transmission line, Test-1 (5 Lp's): Distances from the evaluation point $x(t)$ on the trajectory to the Lp's $x_i$ as the model evolves in the state space](image)

Transmission line, Test-2: The 5 sub-models obtained in test-1 are combined using Tpwl and Ntpwl. Results for Tpwl are strikingly similar to the ones given in [43]. Fig. 4 shows the results for the transient and Fig. 5 for the steady-state simulations. The weight variation of Tpwl is shown in Fig. 6. A comparison of Figures 2, 4, and 6 shows that the response is poorer, and weights are equal in the region where the assessment of sub-models is inadequate. Fig. 7 shows that sharply switching weights are obtained using the new algorithm. The important computational aspects are enumerated in Table-1.

Figure 2: Transmission line, Test-1 (5 Lp's): Distances from the evaluation point $x(t)$ on the trajectory to the Lp's $x_i$ as the model evolves in the state space
Figure 3: Transmission line, Test-1 (15 Lp’s): Distances from the evaluation point $x(t)$ on the trajectory to the Lp’s $x_i$ as the model evolves in the state space.

Figure 4: Transmission line, Test-2: Transient response of the three models.

Figure 7: Transmission line, Test-2: Weight variation of Ntpwl model.
Figure 5: Transmission line, Test-2: Steady-state response of the three models

Figure 6: Transmission line, Test-2: Weight variation of Tpwl model

Table 1: Transmission line, Test-2: Comparison of simulation time and errors.

<table>
<thead>
<tr>
<th>Measure</th>
<th>Nonlinear</th>
<th>Tpwl</th>
<th>Ntpwl</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation time</td>
<td>4.07 s</td>
<td>0.23 s</td>
<td>0.29 s</td>
</tr>
<tr>
<td>% Error (output) (Transient)</td>
<td>-</td>
<td>10.7</td>
<td>2.21</td>
</tr>
<tr>
<td>% Error (all states) (Transient)</td>
<td>-</td>
<td>7.71</td>
<td>1.89</td>
</tr>
<tr>
<td>% Error (output)(Steady-state)</td>
<td>-</td>
<td>12.44</td>
<td>2.5</td>
</tr>
<tr>
<td>% Error (all states) (Steady-state)</td>
<td>-</td>
<td>7.19</td>
<td>0.97</td>
</tr>
</tbody>
</table>

It is obvious that the performance improvement is considerable, the Ntpwl approximation of the nonlinear system output and states is better than Tpwl. Further, the increase in simulation time for the Ntpwl model is negligible.

Transmission line, Test-3: Training and evaluation inputs are both changed. Training is done using $u(t) = 0.5(1 + sin(\pi/5))$ and the models are evaluated by the test input $u(t) = 0.25(1 + cos(\pi/5)) + 0.25(1 + cos(\pi))$. The new evaluation input has two components, one of them is of
higher frequency than the training input. The results for the transient and steady state responses are shown in Figures 8 and 9 respectively. The computational aspects are enumerated in Table-2.

![Graph showing transient and steady state responses for different models](image)

Figure 8: Transmission line, Test-3: Transient Response of the three models.

![Graph showing transient and steady state responses for different models](image)

Figure 9: Transmission line, Test-3: Steady State Response of the three models.

<table>
<thead>
<tr>
<th>Measure</th>
<th>Nonlinear</th>
<th>Tpwl</th>
<th>Ntpwl</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation time</td>
<td>2.46 s</td>
<td>0.11 s</td>
<td>0.18 s</td>
</tr>
<tr>
<td>%Error (output) (Transient)</td>
<td>-</td>
<td>7.78</td>
<td>1.45</td>
</tr>
<tr>
<td>%Error (all states) (Transient)</td>
<td>-</td>
<td>6.29</td>
<td>2.06</td>
</tr>
<tr>
<td>%Error (output) (Steady-state)</td>
<td>-</td>
<td>11.89</td>
<td>1.51</td>
</tr>
<tr>
<td>%Error (all states) (Steady-state)</td>
<td>-</td>
<td>8.06</td>
<td>1.29</td>
</tr>
</tbody>
</table>

Table 2: Transmission line, Test-3: Comparison of simulation time and errors.

Hence, the better approximation qualities of Ntpwl are evident even after a change in the training and evaluation inputs.
5.2. Case-2: RC ladder circuit

The second model is an RC ladder shown in Fig. 10. This model is taken from [40]. The circuit is nonlinear because the resistor \(v\) characteristics are \(i_\text{v}(v) = \frac{1}{R}\sum\text{sgn}(v)v^2\). For further tests we have also varied the severity of the non-linearity, by taking \(i_\text{v}(v) = \frac{1}{R}v^d\), where \(d = 3, 5\). Heaviside function \(u(t) = H(t)\) is used for training and Lp selection. The evaluation input is \(u(t) = 0.5(1 + \cos(\pi t/5))\). Figures 11 and 12 show the transient and steady-state results for cubic nonlinearity. Figures 13 and 14 show the transient and steady-state results for quintic nonlinearity. The computational aspects are enumerated in Tables 3 and 4.

![Figure 10: RC ladder circuit](image)

**Figure 10: RC ladder circuit**

![Figure 11: RC ladder (cubic nonlinearity): Transient Response of the three models.](image)

**Figure 11: RC ladder (cubic nonlinearity): Transient Response of the three models.**
Figure 12: RC ladder (cubic nonlinearity): Steady-state Response of the three models.

Figure 13: RC ladder (quintic nonlinearity): Transient Response of the three models.

Figure 14: RC ladder (quintic nonlinearity): Steady-state Response of the three models.
### Table 3: RC ladder: Comparison of the errors as the severity of the nonlinearity is increased

<table>
<thead>
<tr>
<th>Measure</th>
<th>$x^2$</th>
<th>$x^3$</th>
<th>$x^4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>%Error (output) Tpwl</td>
<td>3.47</td>
<td>7.37</td>
<td>18.66</td>
</tr>
<tr>
<td>%Error (output) Ntpwl</td>
<td>4.87</td>
<td>3.24</td>
<td>2.05</td>
</tr>
</tbody>
</table>

### Table 4: RC ladder: Comparison of simulation time (seconds) for varying nonlinearity

<table>
<thead>
<tr>
<th>Nonlinearity</th>
<th>Nonlinear $T_pwl$</th>
<th>Nonlinear $N_twl$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x^2$</td>
<td>0.95</td>
<td>0.27</td>
</tr>
<tr>
<td>$x^3$</td>
<td>1.04</td>
<td>0.29</td>
</tr>
</tbody>
</table>

It is evident that Tpwl does not work well for severe non-linearities, although it gives good results when the non-linearity is quadratic, because in such a case $||W(x)||$ remains constant as $x$ is varied [39, p. 58]. Ntpwl works well throughout, although the improvement is better as the nonlinearity becomes more severe.

5.3. Case-3: Mems switch

The Mems switch (Fig. 15) has a Poly-silicon beam suspended over a silicon substrate. Voltage between the beam and the substrate causes it to deflect towards the substrate with the thin layer of air in between acting as a damper. The process is modelled with 1D Euler’s beam equation and 2D Reynold’s squeeze film damping equation. For a complete description and the governing equations see [39].

Spatial discretization of the model using standard Finite-difference scheme leads to a large nonlinear dynamical system in the form (1) with $u = v^2$ (where $v$ is the voltage applied between the beam and the substrate) and size $n = 880$. The output $y(t)$ is the deflection of the centre of the beam from the equilibrium point.

In this section we show a comparison in the performance of the Tpwl and Ntpwl models of the Mems device. We also show the computational savings obtained using trajectory based methods in comparison to the standard POD approach.

**Mems switch, Test-1:** Linear sub-models are obtained for the training input $u(t) = (9H(t))^2$ and evaluated for the input $u(t) = (7H(t))^2$. The order of each sub-model is $r = 101$, and 80 such
sub-models are obtained. The results of the test is shown in Fig. 16 and the quantitative description is given in Table-5. It is clear that the Ntpwl model provides an improved reproduction of the nonlinear system dynamics.

![Figure 16: Mem switch, Test-1: Response of the three models](image)

Table 5: Mem switch, Test-1: Comparison of errors.

<table>
<thead>
<tr>
<th>Measure</th>
<th>Tpwl</th>
<th>Ntpwl</th>
</tr>
</thead>
<tbody>
<tr>
<td>%Error (output)</td>
<td>0.402</td>
<td>0.204</td>
</tr>
<tr>
<td>%Error (all states)</td>
<td>0.23</td>
<td>0.12</td>
</tr>
</tbody>
</table>

**Mem switch, Test-2:** Linear sub-models are obtained for the training input $u(t) = (7H(t))^2$ and evaluated for the input $u(t) = (7\cos(2\omega t) + 3\cos(0.5\omega t))^2$ (with $f = 10kHz$). The order of each sub-model is $r = 101$, and 41 such sub-models are obtained. The results of the test is shown in Fig. 17 and the quantitative description is given in Table-6.
Figure 17: Mems switch, Test-2: Response of the three models.

Table 6: Mems switch, Test-2: Comparison of errors.

<table>
<thead>
<tr>
<th>Measure</th>
<th>Tpwl</th>
<th>Ntpwl</th>
</tr>
</thead>
<tbody>
<tr>
<td>%Error (output)</td>
<td>0.622</td>
<td>0.559</td>
</tr>
<tr>
<td>%Error (all states)</td>
<td>0.37</td>
<td>0.33</td>
</tr>
</tbody>
</table>

Memswitch, Comparison with POD: As mentioned in the Introduction, the primary advantage of Trajectory based methods in comparison to Pod is the improved computational performance of the former. This is illustrated here using Numerical results from the Mems model. In Table-7, the simulation time for Pod and Trajectory based piece-wise linear Roms are compared. The inputs used for evaluating these models remain unchanged from the previous test.

Table 7: Mems switch: Simulation time of the Roms in seconds and comparison with the original System.

<table>
<thead>
<tr>
<th>Test</th>
<th>Nonlinear system</th>
<th>POD model</th>
<th>Tpwl</th>
<th>Ntpwl</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memswitch, Test-1</td>
<td>1821.73</td>
<td>1335.6</td>
<td>3.2</td>
<td>5.2</td>
</tr>
<tr>
<td>Memswitch, Test-2</td>
<td>1750.51</td>
<td>1327.9</td>
<td>3.3</td>
<td>5.2</td>
</tr>
</tbody>
</table>

It is very evident that the Pod model does not lead to significant computational savings as compared to the full-order, nonlinear problem. The piece-wise linear models, however, lead to significant savings. Further, as observed earlier, the Ntpwl model is slightly more expensive to simulate as compared to the Tpwl model.

5.4. Improved robustness

The dependance of the Tpwl model output on the kind of training has been cited as one of the drawbacks of this method, which makes it less robust to a changing input environment. Although we cannot expect very robust models to be generated using one training input, less variation in model behaviour with change in training is always welcome. Our algorithm shows an increased robustness which is reflected in a lesser change in model behaviour with two different types of training. We have tested this for the two nonlinear models, with the $RC$ ladder having quintic
nonlinearity. The two training inputs are $u_1(t) = H(t)$ and $u_2(t) = 0.5(1 + \sin(2\pi t)/10)$. The evaluation input is $u(t) = 0.45(1 + \cos(\pi t)) + 0.05(1 + \cos(10\pi t))$. The results are shown in Figures 18 and 19. The Tables 8 and 9 enumerate the numerical aspects.

![Figure 18: Transmission line model: Comparison of variation in output for different training inputs.](image)

![Figure 19: RC ladder circuit: Comparison of variation in output for different training inputs.](image)

<table>
<thead>
<tr>
<th>Measure</th>
<th>% Error-$u_1(t)$</th>
<th>% Error-$u_2(t)$</th>
<th>% Diff. in response to $u_1(t)$&amp;$u_2(t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tpwl (output)</td>
<td>14.84</td>
<td>14.53</td>
<td>2.843</td>
</tr>
<tr>
<td>Ntpwl (all states)</td>
<td>3.19</td>
<td>2.13</td>
<td>1.7285</td>
</tr>
<tr>
<td>Tpwl (output)</td>
<td>10.79</td>
<td>10.85</td>
<td>1.483</td>
</tr>
<tr>
<td>Ntpwl (all states)</td>
<td>1.96</td>
<td>1.78</td>
<td>1.3022</td>
</tr>
</tbody>
</table>

Table 8: Nonlinear transmission line: Errors for Tpwl and Ntpwl models obtained using different training inputs for the test input $u(t) = 0.45(1 + \cos(\pi t)) + 0.05(1 + \cos(10\pi t))$
Measure   & % Error-$u_1(t)$ & % Error-$u_2(t)$ & % Diff. in response to $u_1(t)\& u_2(t)$ \\ 
Tpwl (output) & 14.88 & 2.27 & 11.563 \\ 
Ntpwl (output) & 2.14 & 1.32 & 0.9365 \\ 
Tpwl (all states) & 14.09 & 4.01 & 10.312 \\ 
Ntpwl (all states) & 3.85 & 3.56 & 0.8255 \\

Table 9: RC ladder circuit: Errors for Tpwl and Ntpwl models obtained using different training inputs for the test input $u(t) = 0.45(1 + \cos(\pi t)) + 0.05(1 + \cos(10\pi t))$

6. Conclusion

An attempt is made to address an outstanding issue in the Tpwl method. The causes of error in effective reproduction of nonlinear system dynamics are analysed and it is found that the conventional weight-assignment strategies are inadequate for good superposition of sub-models. A solution based on dynamical-systems viewpoint which proposes assessing sub-models using state-positions and state-velocities is introduced. It is shown that this helps in taking the system nonlinearity into account for superposition. A new procedure is given for quick and computationally light estimation of state-velocities. Case studies on nonlinear circuits demonstrate efficacy of the new method, called Ntpwl. All this is achieved without increasing the computational burden significantly. However, Ntpwl was proposed and applied to input-affine nonlinear dynamical systems. Its application to more general nonlinear systems requires additional work as in such systems the self-dynamic component of the state-velocity cannot be separated. Such an exercise can be taken up in the future.

References

40. M. Rewienski, J. White, A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices, IEEE Transactions on Computer-Aided Design of Integrated Cir-


