Trajectory Piecewise Linear approach for nonlinear differential-algebraic equations in circuit simulation

Thomas Voß
Delft University of Technology
Mekelweg 2, 2628 CD Delft, The Netherlands
t.voss@tudelft.nl

***Abstract*—**
In this paper we present the Trajectory Piecewise Linear (TPWL) method for differential algebraic equations (DAE). The TPWL method is based on combining several linear reduced models, which are created on a typical trajectory, to approximate the full nonlinear model.
We discuss how to select of the equlibiria for linearisation and how we can create a gobal reduced subspace. Then we study show how to combine the local linearised reduced systems to create a global TPWL model. Finally, we show a numerical result of the TPWL method.

I. INTRODUCTION

Nowadays a lot of nonlinear circuits which are used in many fields are a mixture of analogue and digital parts. The digital parts contain several sub-circuits, that have the same dynamics and only have different inputs. So simplifying these parts gives a good speed up for the transient analysis.

To do this we could use methods which are based on linear or quadratic reduction [1] or other methods, e.g. proper orthogonal decomposition (POD) [2], but these methods are mostly developed for weak nonlinear systems. This makes these methods not so useful in circuit simulation, which often deals with highly nonlinear circuits. To overcome this issue, a Trajectory Piecewise Linear (TPWL) [3] approach for ordinary differential equations (ODE) was developed. We will show how we can adapt these method to DAEs.

In the next section we present our TPWL approach for nonlinear DAEs. In Section III. we show how the method performs in practice. Finally in Section IV. we draw our conclusions.

II. TRAJECTORY PIECEWISE LINEAR MODEL ORDER REDUCTION

The idea behind the TPWL method is to linearise the system several times along a typical trajectory. Then we use the local linearised systems to create a global reduced subspace and project each of them into this global reduced subspace. The final TPWL model is then a weighted sum of all local linearised reduced systems. In the following subsections we show how we apply the described steps.

The DAE system where we want to apply the TPWL method is

$$\frac{d}{dt}\mathbf{q}(t,\mathbf{x}) + \mathbf{j}(t,\mathbf{x}) + \tilde{B}\tilde{\mathbf{u}}(t) = \mathbf{0}$$

where $\mathbf{q}, \mathbf{j} : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$, $B \in \mathbb{R}^{n\times m}$ and $\mathbf{u} : \mathbb{R} \rightarrow \mathbb{R}^m$.

A. Creating the local linearised models

The disadvantage of the standard linearisation methods is that the we can only trust in the results, if the solutions stay close to the linearisation tuple (LT) around which we have created the linearised model. To overcome this disadvantage the idea is to take several linearised models to create the TPWL model. Then we can trust in the results as long as the solution stays close to one of the LT.

Because we also have to calculate a trajectory of the system, e.g. via a backward differential formula (BDF) approach, it is a good idea to include the selection of the LTs directly in such a solver. We use the following strategy for selecting new LTs.

We know that we will project the linearised system to global reduced systems. So we reduce our local system locally. And then simulate the local linearised additionally to the original system. If the distance between both solutions is getting too large we set a new LT. So we get the following procedure to find $i+1$-th LT.

1. Set an absolute accuracy factor $\varepsilon > 0$, set $i = 1$
2. Linearise the system around the i-th LT (x_i, t_i). So we get
$$C_i\dot{\mathbf{x}} + G_i\mathbf{x} + B_i\mathbf{u}(t) = 0$$
where $C_i = \frac{\partial}{\partial x}\mathbf{q}(t,\mathbf{x})\big|_{\mathbf{x}_i,t_i}$ and $G_i = \frac{\partial}{\partial x}\mathbf{j}(t,\mathbf{x})\big|_{\mathbf{x}_i,t_i}$ Save C_i, G_i and B_i.
3. Reduce the linearised system to dimension $r \ll n$ with a linear model reduction method, e.g. 'Poor Mans' TBR (PMTBR) [4] or a Krylov approach [5].
$$C_i^r\dot{\mathbf{y}} + G_i^r\mathbf{y} + B_i^r\mathbf{u}(t) = 0$$
where $C_i^r = P_i^\top C_i P$, $G_i^r = P_i^\top G_i P_i$, $B_i^r = P_i^\top B$ with $P_i \in \mathbb{R}^{r\times n}$. $\mathbf{y} \in \mathbb{R}^r$ is the approximation to $\mathbf{x}$ with $\mathbf{x} \approx P_i\mathbf{y}$. Save P_i.
4. Simulate the reduced system with $\mathbf{y}_0 = P_i^\top\mathbf{x}_i$ and the original system. If the absolute distance between the two solutions $\frac{||P\mathbf{y}-\mathbf{x}||}{||\mathbf{x}||}$ is bigger than ε then set the $i+1$-th LT to $(\mathbf{x}, t)$ and go to step 2.

We continue with this procedure until we have reached the end of the given trajectory.

B. Creating the global reduced subspace

Now we can construct the global subspace. The idea is to merge all local reduced subspace to get the global reduced subspaces. To do this we create $\tilde{P} := [P_1, \ldots, P_p] \in \mathbb{R}^{n \times rp}$ which spans then the union of all local reduced subspaces. From our construction we see that the columns of $\tilde{P}$ are in general not linear invariant. Therefore we use a singular value decomposition (SVD) to create the final global subspace. Then we project each of the local linearised systems to the global subspace. We use the following procedure

1. Define $\tilde{P} = [P_1, \ldots, P_p]$.
2. Calculate the SVD of $\tilde{P}$. So $\tilde{P} = U \Sigma V^\top$ with $U = [u_1, \ldots, u_n] \in \mathbb{R}^n$,$\Sigma \in \mathbb{R}^{n \times rp}$ and $V \in \mathbb{R}^{rp \times rp}$.
3. Define P as $[u_1, \ldots, u_r]$.
4. Create the p local linearised reduced systems given as $C_{ir}\dot{\mathbf{y}} + G_{ir}\mathbf{y} + B_{ir}\mathbf{u}(t) = \mathbf{0}$ with $C_{ir} = P^\top C_i P$, $G_{ir} = P^\top G_i P$ and $B_{ir} = P^\top B_i$

C. Creating the TPWL model by weighting

Now we need to combine the local linearised reduced systems to get the global TPWL model. We do this by a weighted sum of local models

$$\sum_{i=1}^{p} w_i C_{ir}\dot{\mathbf{y}} + \sum_{i=1}^{p} w_i G_{ir}\mathbf{y} + \sum_{i=1}^{p} w_i B_i \mathbf{u}(t) = 0.$$

The weights w_i represent the influence of the i-th local system to the global system. A way of choosing the weights is to make them distance depending. This means w_i is large if the solution $\mathbf{y}$ is close to the i-th LT, because then it is in the accuracy region of the LT. Is the solution far away from the i-th LT then the weight has to be small. After calculating the weights we normalise them to get a convex combination of the local linearised reduced systems, which is described by the following procedure

1. Given actual state $\mathbf{y}$, actual time t, p LT $(\mathbf{x}_i, t_i)$ and $\alpha_{\mathbf{y}}, \alpha_t \geq 0$ with $\alpha_{\mathbf{y}} + \alpha_t = 1$
2. For $i = 1, \ldots, p$ compute $d_i = \alpha_{\mathbf{y}} \left\| \mathbf{y} - P^\top \mathbf{x}_i \right\| + \alpha_t \left| t - t_{l_i} \right|$
3. For $i = 1, \ldots, p$ calculate $\tilde{w}_i = e^{-\frac{d_i \beta}{m}}$ with $m = \min_{i=0,\ldots,s-1} d_i$, $\beta > 0$
4. Normalise the weights:
   $w_i = \frac{\tilde{w}_i}{s}$ with $s = \sum_{i=0}^{s-1} \tilde{w}_i$

III. Example

Now we want to show how the TPWL method performs in practice. As a test circuit we have chosen a chain of inverters, which consists of 100 inverters which are connected in series. The circuit behaves nonlinearly so it is a good test for the TPWL method. Also we have dependencies between all nodes which is also not an optimal behaviour for a model reduction process. The DAE which is describing the dynamics of the circuit has 104 states. For selecting the LTs we have used the proposed method, the linear model reduction technique we used is PMTBR.

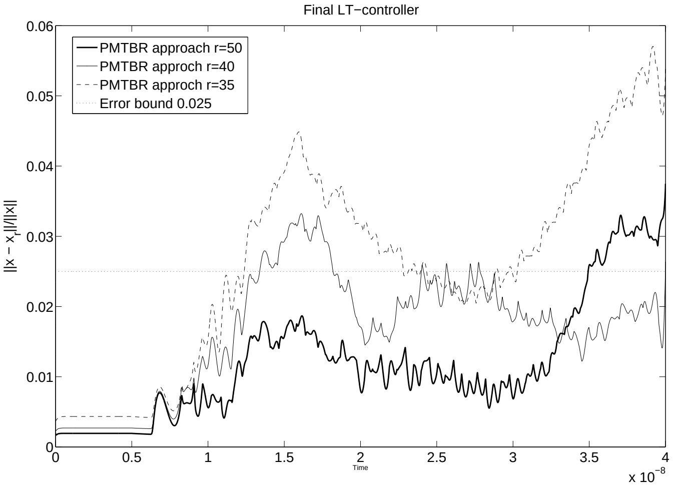

Fig. 1. Relative error of the TPWL method for different orders

In Figure 1 on page 2 we see that the relative error is most of the time lower then the given error bound. For all orders we have to use the same number of LTs (62), this comes from the fact that the local systems only need relatively small subspaces to get the desired accuracy. The resulting speed up is between 5.4 and 8.3 compared to a BDF method.

IV. Conclusion

The TPWL method applied to nonlinear DAEs is a promising technique to reduce the simulation time. It has several advantages compared to other methods. First of all we can get a big speed up in simulation time. Next we can use the well-developed linear model reduction techniques. A TPWL method is also scalable. This means that by using different linearisation tuple controllers, linear model reduction techniques and weighting methods, we can change the method from a fast to a much more accurate method.

Acknowledgment

Thanks go to Dr. Jan E.W. ter Maten, Philips research laboratories Eindhoven, Netherlands and Prof. Dr. Michael Günther, University of Wuppertal, Germany who both have supported my research.

References

[1] J.R. Phillips, “Projection frameworks for model reduction of weakly nonlinear systems,” 2000.
[2] S. Volkwein, “Proper orthogonal decomposition and singular value decomposition,” 1999.
[3] M.J. Rewieński, *A trajectory piecewise-linear approch to model order reduction of nonlinear dynamical systems*, PhD Thesis, Massachutes Institute of Technology, 2003.
[4] L.M. Silvera J. Phillips, “Poor man’s tbr: A simple model reduction scheme,” 2005.
[5] L.T. Pileggi A. Odabasioglu, M. Celik, “Prima: Passive reduced-order interconnect macromodeling algorithm,” pp. 645–654, 1998.