QLMOR: A New Projection-Based Approach for Nonlinear Model Order Reduction

Chenjie Gu
University of California, Berkeley
gcj@eecs.berkeley.edu

ABSTRACT

We present a new projection-based nonlinear model order reduction method, named QLMOR (MOR via quadratic-linear systems). QLMOR employs two novel ideas: (1) we show that DAEs (differential-algebraic equations) with many commonly-encountered nonlinear kernels can be re-written equivalently into a special format, QLDAEs (quadratic-linear differential algebraic equations, *i.e.*, DAEs that are quadratic in their state variables and linear in their inputs); (2) we adapt the moment-matching reduction technique of NORM[1] to reduce these QLDAEs into QLDAEs of much smaller size.

Because of the generality of the QLDAE form, QLMOR has significantly broader applicability than Taylor-expansion based methods [2, 3, 1]. Importantly, QLMOR, unlike NORM, totally avoids explicit moment calculations (A^iB terms), hence it has improved numerical stability properties as well. Because the reduced model has only quadratic nonlinearities (*i.e.*, no cubic and higher-order terms), its computational complexity is less than that of similar prior methods[2, 3, 1]. We also prove that QLMOR-reduced models preserve local passivity, and provide an upper bound on the size of the QLDAEs derived from a polynomial system.

We compare QLMOR against prior methods [2, 3, 1] on a circuit and a biochemical reaction-like system, and demonstrate that QLMOR-reduced models retain accuracy over a significantly wider range of excitation than Taylor-expansion based methods [2, 3, 1]. Indeed, QLMOR is able to reduce systems that Taylor-expansion based methods fail to reduce due to passivity loss and impractically high computational costs. QLMOR therefore demonstrates that Volterra-kernel based nonlinear MOR techniques can in fact have far broader applicability than previously suspected, possibly being competitive with trajectory-based methods (*e.g.*, TPWL [4]) and nonlinear-projection based methods (*e.g.*, maniMOR [5]).

1. INTRODUCTION

Model order reduction (MOR) has been an active research area for the past two decades. It refers to the procedure of reducing a large dynamical system to a much smaller one which captures the dominant dynamics of the original system. MOR thus enables higher-level modeling and simulation of large systems, and is important in many engineering domains, *e.g.*, circuit design and analysis, chemical kinetics, mechanical systems (or MEMS), ecosystems, building design, and so forth.

Most applications are fundamentally nonlinear in nature (although linear time-invariant (LTI) or linearized systems constitute an important special case on which much research has been conducted [6, 7, 8, 9, 10, 11, 12]). However, MOR for nonlinear systems is much more difficult than for LTI systems. One source of the difficulty stems from that nonlinear systems (even very simple ones) can exhibit complex behaviors that linear systems are incapable of, which means that reduced models for nonlinear systems need to preserve much richer behaviors than for the linear case. For example, multiple equilibria (*e.g.*, bistability) and stable oscillations can only appear in nonlinear systems; the Lorenz system [13] (which only has quadratic nonlinearities) and Chua's circuit [14] (which has only one piecewise-linear nonlinear element), exhibit surprisingly complicated chaotic behaviors – they are hard to characterize and reduce.

The second difficulty is that nonlinear systems lack a general, universal input-output representation analogous to transfer functions for LTI systems. Transfer functions $H(s)$ of LTI systems constitute a powerful tool for MOR – for example, a fundamental underlying concept in LTI MOR, moment-matching [6, 7, 8, 9], relies centrally on LTI transfer functions. (*Moments* refer to the coefficients of $1, s, s^2, \cdots$ of the Taylor expansion of $H(s)$.)

For nonlinear systems, useful generalization of the transfer function concept is only possible for special cases. Specifically, for polynomial nonlinear systems, Volterra theory [15] establishes that the I/O relationship can be represented by a series of *Volterra kernels*, or *high-order transfer functions*, as detailed in Section 2.1. Given these high-order transfer functions, the concepts of moments and moment-matching have been extended to reduce polynomial systems [2, 3, 1]. In particular, NORM [1] provides a systematic way to construct the minimum subspace for matching high-order moments.

In order to derive a polynomial system from a general nonlinear system, [2, 3, 1] perform a Taylor approximation of the original system around its steady state solution. Due to the local convergence of Taylor approximation, these methods are only applicable to "weakly nonlinear" systems, or systems with "small" inputs. Moreover, Taylor expansions can lead to stability/passivity loss, and can generate reduced systems that are more computationally expensive than the original one. These issues frequently preclude wide applicability of Taylor-expansion based methods, as discussed in Section 2.1.

To avoid the weakly nonlinear constraint, a class of nonlinear MOR methods known as *trajectory-based* methods (*e.g.*, POD[16], TPWL[4, 17, 18, 19], maniMOR[5]) has been proposed. The term *trajectory* refers to a path in state space when the system is excited by a given input. Along each section of a trajectory, the system is locally approximated as a linear/polynomial system, and is reduced using methods for linear/polynomial systems. The local reduced models are then joined together to form an overall nonlinear reduced model. Evaluation of the reduced model can still be expensive since it depends not only on the number of local approximations used, but also on the polynomial order of each local reduced model.

In this paper, we propose a new method, named QLMOR (model order reduction via quadratic-linear systems), to reduce nonlinear systems that are *quadratic-linearizable*. The method first converts a nonlinear system into an equivalent *quadratic-linear differential algebraic equation* (QLDAE) system, in which the differential equations are quadratic in their state variables and linear in their inputs. QLMOR then reduces this QLDAE into a smaller QLDAE system by matching moments of high-order transfer functions.

A key difference between QLMOR and prior Taylor-expansion based methods is that we can re-write the original system in an exact *equivalent* QLDAE representation, *i.e.*, *without any approximations*. Therefore, the QLDAE representation serves as sort of a "canonical form" of the original system, except for its non-uniqueness. The utility of QLMOR stems from that many common nonlinear

ICCAD'09, November 2–5, 2009, San Jose, California, USA.

kernels appearing in dynamical systems are quadratic-linearizable – *e.g.*, e^x is common in electronic device models (such as diodes and BJTs) as well as biochemical ion channel models; polynomial nonlinearities (particularly quadratic nonlinearities) are common in MOSFET models and are also basic to chemical rate equations; $\frac{x}{1+x}$ is useful for modeling smoothing functions, as well as in Michaelis-Menten kinetics.

This key difference makes QLMOR-reduced models valid for "large" inputs. It also circumvents the passivity/stability loss that results from Taylor approximation, and makes the passivity/stability preservation problem more tractable. Indeed, we prove that the reduced model, when locally linearized, is a congruence transformation of the corresponding linearization of the full system. Therefore, passivity of the linearized model is preserved if certain conditions are satisfied (Section 5.4).

The "price paid" for this equivalent QLDAE representation is that the system size must be increased before it is reduced. This is similar to MOR for linear periodic time-varying systems [2] and MOR based on bilinear forms [3]. We prove an upper bound on the size of the equivalent QLDAE representation derived from a polynomial system. However, in applications, the size of QLDAEs often does not hit the upper bound, and is, indeed, much smaller than that of a reasonable bilinear representation.

We reduce the QLDAE system by extending and generalizing the NORM technique [1]. Since QLDAEs have terms that are products of state variables and inputs, we need to re-derive and modify both the theory and the algorithm in [1] so that they can apply to QLDAEs. As a result, we identify the *minimum* subspace that guarantees matching any number of specified moments. Also, unlike [1], we are able to completely avoid explicit evaluation of A^iB terms, leading to better numerical stability over [1]. Our derivation also unifies methods based on bilinear and quadratic forms, which turn out to be just special forms of QLMOR.

We show that projection of the original system onto the identified subspace results in a reduced system that is also in QLDAE form. This represents another advantage of QLMOR over methods based on Taylor expansion. The computational cost of the QLMOR-reduced model is dominated by terms of the form $\hat{G}_2 z \otimes z$, which is $O(q^3)$ – q is the (small) size of the reduced model. In contrast, prior polynomial reduced models typically have cubic and higher order terms, resulting in a computational complexity of at least $O(q^4)$.

The remainder of the paper is organized as follows. In Section 2, we review model order reduction methods for weakly nonlinear and general nonlinear systems. We then provide an overview of the QLMOR method (Section 3), and discuss details of the quadratic-linearization procedure (Section 4) and the model reduction algorithm of QLDAEs (Section 5). We validate QLMOR on a nonlinear transmission line circuit and a biochemical reaction-like system, and provide comparisons to prior methods in Section 6.

2. BACKGROUND

In this paper, we focus on SIMO dynamical systems in the form of

$$C\dot{x}(t) = f(x(t)) + Bu(t), \qquad y(t) = E^T x(t), \tag{1}$$

where $x \in \mathbb{R}^N$ are the state variables, $y \in \mathbb{R}^{n_o}$ are the outputs, and $u \in \mathbb{R}$ is the input. Note that all the derivations can be straightforwardly extended to MIMO systems, time-varying systems, parameterized systems and the general DAE formulation $\frac{d}{dt}q(x(t)) = f(x(t), u(t))$.

2.1 Previous Volterra-Series Based Methods

Previous Volterra-series based MOR methods [2, 3, 1] are derived from the theorem that the response of a nonlinear system can be decomposed into responses of a series of homogeneous nonlinear systems, *i.e.*, the system response $x(t)$ can be written as the summation of responses all n-th order homogeneous nonlinear systems $x_n(t)$, as shown in (2). In (3) and (4), $h_n(\sigma_1, \cdots, \sigma_n)$ is the *Volterra kernel of order n*, and $H_n(s_1, \cdots, s_n)$, the *transfer function of order n*, is the multi-variable Laplace transform of $h_n(\sigma_1, \cdots, \sigma_n)$ [15].

$$x(t) = \sum_{n=1}^{\infty} x_n(t), \tag{2}$$

$$\begin{aligned} x_n(t) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} & h_n(\sigma_1, \cdots, \sigma_n) \\ & u(t-\sigma_1) \cdots u(t-\sigma_n)\, d\sigma_1 \cdots d\sigma_n, \end{aligned} \tag{3}$$

$$\begin{aligned} H_n(s_1, \cdots, s_n) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} & h_n(\sigma_1, \cdots, \sigma_n) \\ & e^{-(s_1\sigma_1 + \cdots + s_n\sigma_n)}\, d\sigma_1 \cdots d\sigma_n. \end{aligned} \tag{4}$$

Volterra-series based methods generally consist of four steps:
(1) Perform a Taylor expansion of the nonlinear function $f(x)$, *i.e.*, expand $f(x)$ in a series of multidimensional polynomials, and truncate the expansion to a predefined order (*e.g.*, 3, as shown in (5), where symbol $\otimes$ denotes the Kronecker product[1]).

$$C\dot{x} = G_1 x + G_2 x \otimes x + G_3 x \otimes x \otimes x + Bu \tag{5}$$

(2) For methods based on bilinear forms, perform a Carleman bilinearization on (5) to obtain a bilinear approximation (6).

$$C\dot{y} = Gy + Dyu + Bu \tag{6}$$

(3) Construct the projection matrix $V \in \mathbb{R}^{N \times q}$ ($q \ll N$ is the size of the reduced system), to match the moments of H_n, $n = 1, 2, 3, \cdots$ up to a given order. *E.g.*, to match up to the second order moments of $H_2(s_1, s_2)$ means to match the coefficients of terms $1, s_1, s_2, s_1^2, s_1 s_2, s_2^2$ in the Taylor expansion of $H_2(s_1, s_2)$ [1].
(4) Construct the projected reduced system. *E.g.*,

$$\hat{C}\dot{z} = \hat{G}_1 z + \hat{G}_2 z \otimes z + \hat{G}_3 z \otimes z \otimes z + \hat{B}u \tag{7}$$

by defining $x = Vz$, $\hat{C} = V^T C V$, $\hat{B} = V^T B$, $\hat{G}_1 = V^T G_1 V$, $\hat{G}_2 = V^T G_2 (V \otimes V)$, $\hat{G}_3 = V^T G_3 (V \otimes V \otimes V)$.

The common step in [2, 3, 1] is Taylor approximation. This results in three major drawbacks that limit the wide applicability of these methods. Firstly, the convergence of the Volterra system representation is only valid when $|u| \leq \epsilon$, for some small $\epsilon > 0$, *i.e.*, for "small" inputs, or the "weakly nonlinear" case. This constraint stems from the fact that the Taylor series only converges in a small region around the expansion point. Incorrect behavior of the reduced model due to large inputs was reported in [3], and similar results are reproduced in Section 6.2.

Secondly, Taylor expansion typically destroys important global properties of the system, such as passivity and stability[2]. For example [3], the quadratic approximation for e^x (part of the I-V function of a diode) can generate energy. As a result, the reduced model, which is created using the Taylor approximated system as a starting point, will typically not be passive/stable.

Thirdly, the computational cost for third-order terms such as $\hat{G}_3\, z \otimes z \otimes z$ and higher order terms dominates the computational cost of the reduced model. Since $z \in \mathbb{R}^q$, $\hat{G}_3 \in \mathbb{R}^{q \times q^3}$, and with $\hat{G}_3$ usually dense, the computational cost is $O(q^4)$. (It is possible to make $\hat{G}_3$ of a smaller size by combining terms in $z \otimes z \otimes z$, but the computational complexity does not change.) Moreover, it is generally difficult to accurately extract the high-order matrices $G_2, G_3, \cdots$, which also constrains applications of Taylor-expansion based methods.

2.2 Trajectory and Nonlinear Projection Based Methods

To reduce strongly nonlinear systems, trajectory-based methods [16, 4, 17, 18, 19] have been proposed. These methods follow the linear projection framework[20] (*i.e.*, finding a projection $x = Vz$, and projecting the original system onto the subspace defined by the column span of V.). A basic concern is the computational cost of $V^T f(Vz)$ which can be so expensive that no computational

[1] For notational simplicity, we will sometimes use $x^{\textcircled{2}}$ to denote $x \otimes x$, $x^{\textcircled{3}}$ to denote $x \otimes x \otimes x$, *etc.*.

[2] Definitions for passivity and stability are not universally agreed upon, but broadly speaking, a device is passive if it does not generate energy, and a system is input-output stable if the outputs of the system are bounded given that the inputs are bounded.

efficiency over the original model is obtained, despite the smaller model size.

To address this problem, special forms for $f(\cdot)$ can be exploited. For example, TPWL uses piecewise linear approximations to $f(\cdot)$ (*i.e.*, compute $f(\cdot)$ on a per region basis) such that the projections V^T and V can be "passed through" $f(\cdot)$, *i.e.*, pre-computed during the reduction step. However, with the increase of the number of piecewise regions, the computational cost increases. Even if local projection[17] is used, the computational time for distance evaluation and nearest-neighbor search can be large.[3]

An extension of this projection framework for nonlinear systems is to construct a nonlinear manifold in the state space, and project the original system onto that manifold – this has been proposed in maniMOR [5] recently. However, since the manifold and projections are also computed on a per region basis, it shares a similar model computation problem as in trajectory-based methods.

Note that MOR based on bilinear forms [3] actually defines a nonlinear projection – *e.g.*, by explicitly writing out the differential equations for $x \otimes x, x \otimes x \otimes x$, the final projection is defined by $z = V^T[x^T, x^T \otimes x^T, x^T \otimes x^T \otimes x^T]^T$, which is a polynomial projection function. It will be seen in Section 4 that QLMOR indeed allows a richer set of projections than Carleman bilinearization.

To briefly summarize this section, we list in Table 1 the main drawbacks and difficulties that have not been solved in existing MOR methods. We also indicate how well QLMOR addresses these drawbacks.

Table 1: Main Drawbacks of Existing Methods

Existing methods	Drawbacks	QLMOR
Volterra series based	Valid only for small-signal response	Solved
	Stability, passivity loss	Partially solved
	Expensive computation if higher-order terms are included	Improved
Trajectory based	Expensive $V^T f(Vz)$ computation	Improved

3. QLMOR OVERVIEW

In this section, we outline the key ideas of QLMOR. Details are provided in Section 4 and Section 5.

From the discussion in Section 2, the drawbacks of previous Volterra series based methods are mainly caused by Taylor expansion step, in which "information" about the system is lost.

In contrast, QLMOR achieves a polynomial system that is equivalent to the orignal system without losing any information. By doing so, QLMOR avoids the "small" input assumption, and preserves properties of the original system in the first step.

We further show that any polynomial system can be converted into *quadratic-linear differential algebraic equations* (QLDAE)

$$C\dot{x} = G_1 x + G_2 x \otimes x + D_1 x u + D_2 (x \otimes x) u + Bu \tag{8}$$

i.e., the differential equations are quadratic in state variables x and linear in the input u. This QLDAE system is another equivalent representation of the original system, and therefore no approximation is involved in this step, either.

Then QLMOR reduces this QLDAE system to produce a smaller model that is also in the QLDAE form, as opposed to the polynomial form in prior methods. The important advantage here is that higher-order terms beyond quadratic terms are absent in QLMOR-reduced model. Therefore, it avoids the expensive computational cost (*e.g.*, $z \otimes z \otimes z$) in a high-order polynomial reduced model, and quadratic terms dominates the computational complexity.

To generate the reduced model, we use a projection-based approach. To obtain the *minimum* subspace in the sense of moment matching, we analyze the transfer functions of systems in the QLDAE form, and results related to prior methods are derived. For example, it turns out that the moments of $H_2(s_1, s_2)$ of a QLDAE system are a summation of the moments of $H_2(s_1, s_2)$ of a quadratic system and a bilinear system – direct application of existing codes are possible. In our algorithm, we also avoid explicit moment evaluations, and therefore the algorithm has better numerical stability properties.

Another point to note is that QLMOR serves as a new core method that can be combined with any trajectory-based methods. For example, in TPWL method, to generate reduced models along a trajectory, previous Krylov-subspace methods or TBR methods can be replaced with QLMOR straightforwardly, potentially leading to a more accurate model and less number of regions along the trajectory. For another example, POD method can be directly applied to the transformed QLDAE system, which results in a reduced model whose computational cost is lower than the usual POD reduced model if the reduced model size remains the same.

We chart in Fig. 1 the basic flow of QLMOR, as well as the flow of MOR based on polynomial systems and bilinear systems. In what follows, Section 4 presents the quadratic-linearization procedure, and Section 5 discusses the reduction step, *i.e.*, how to generate the minimum subspace.

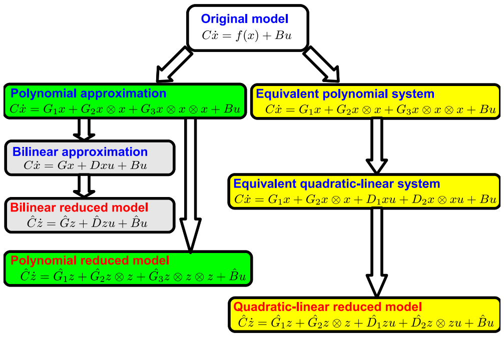

Original model
$C\dot{x} = f(x) + Bu$
Polynomial approximation
$C\dot{x} = G_1x + G_2x \otimes x + G_3x \otimes x \otimes x + Bu$
Equivalent polynomial system
$C\dot{x} = G_1x + G_2x \otimes x + G_3x \otimes x \otimes x + Bu$
Bilinear approximation
$C\dot{x} = Gx + Dxu + Bu$
Equivalent quadratic-linear system
$C\dot{x} = G_1x + G_2x \otimes x + D_1xu + D_2x \otimes xu + Bu$
Bilinear reduced model
$\hat{C}\dot{z} = \hat{G}z + \hat{D}zu + \hat{B}u$
Polynomial reduced model
$\hat{C}\dot{z} = \hat{G}_1z + \hat{G}_2z \otimes z + \hat{G}_3z \otimes z \otimes z + \hat{B}u$
Quadratic-linear reduced model
$\hat{C}\dot{z} = \hat{G}_1z + \hat{G}_2z \otimes z + \hat{D}_1zu + \hat{D}_2z \otimes zu + \hat{B}u$

Figure 1: QLMOR flow (yellow), with comparison to MOR based on polynomial form (green) and bilinear form (gray).

4. QUADRATIC-LINEARIZATION

There are two steps in the quadratic-linearization procedure:

1. convert the original system (1) into a polynomial system;
2. convert the polynomial system into QLDAEs (8).

To begin with, the polynomial system of order M is defined as follows:

Definition 4.1 (Polynomial system of order M). *A polynomial system (9) is said to be of order M, if the highest order of monomial of x is M, i.e., $M = \max_{k,i}\left(\sum_{l=1}^{N} \beta_{i,k,l}, \sum_{l=1}^{N} \gamma_{i,k,l}\right)$.*

$$\sum_k c_{i,k}\dot{x_k} = \sum_k \alpha_{i,k} x_1^{\beta_{i,k,1}} \cdots x_N^{\beta_{i,k,N}} + \left(\sum_k \eta_{i,k} x_1^{\gamma_{i,k,1}} \cdots x_N^{\gamma_{i,k,N}}\right) u$$
$$i = 1, \cdots, N, \beta_{i,k,j} \geq 0, \gamma_{i,k,j} \geq 0, \beta_{i,k,j} \in \mathbb{Z}, \gamma_{i,k,j} \in \mathbb{Z}, \forall k, j \tag{9}$$

We now present two approaches to convert a nonlinear system where $f(x)$ can be written as a linear combination of some set of simple functions, *i.e.*,

$$\dot{x} = a_0 x + a_1 g_1(x) + a_2 g_2(x) + \cdots + a_k g_k(x) + Bu, \tag{10}$$

into a polynomial system in the form of (9). We call this procedure *polynomialization*.

Notice that this form of differential equations is prevalent in circuit simulation, as well as other engineering problems. For example, in circuit MNA equations, each $g_i(x)$ represents the current flowing into a node from a specific device; in chemical rate equations, each $g_i(x)$ represents the reaction rate according to a specific reaction; in mechanical applications, $g_i(x)$ are the forces to the system.

4.1 Polynomialization by Adding Polynomial Algebraic Equations

The first approach is only applicable to certain nonlinear functions, *e.g.*, $\frac{1}{k+x}$. The procedure is quite straightforward – one just

[3] However, apple-to-apple comparisons of trajectory-based methods to Volterra-series based methods are complicated, not only because there are heuristics involved in TPWL (*e.g.*, in function evaluations and nearest neighbor search), but also because definitions of accuracy vary. (*E.g.*, TPWL is usually accurate in regions that are modeled, and Volterra-series based methods are accurate in modeling the first few harmonic distortions.)

needs to make a variable change $y_i = g_i(x)$, and add the polynomial algebraic equation derived from $y_i = g_i(x)$ into the original system.

For example, if $\dot{x} = x + x^2 + \frac{1}{k+x}$, we make the variable change $y = \frac{1}{k+x}$, and then the system is converted to

$$\dot{x} = x + x^2 + y, \qquad 0 = -ky - xy + 1. \tag{11}$$

The assumption is that the variable change can lead to a polynomial algebraic equation. For example, any variable change to a rational function leads to a polynomial algebraic equation. Constrained by this assumption, this approach cannot deal with nonlinear functions such as $y = e^x$.

4.2 Polynomialization by Taking Lie Derivatives

To polynomialize a larger set of nonlinear functions, one can also add differential equations, instead of algebraic equations.

Given (10), if we make the variable change $y_i = g_i(x), i = 1, \cdots, k$, and write out the differential equations for y_i by taking its Lie derivative with respect to $\dot{x}$, we obtain an expanded system

$$\begin{aligned} \dot{x} &= a_0 x + a_1 y_1 + \cdots + a_k y_k + Bu, \\ \dot{y_i} &= \mathfrak{L}_{\dot{x}} g_i(x) = g_i'(x)(a_0 x + a_1 y_1 + \cdots + a_k y_k + Bu), \end{aligned} \tag{12}$$

where $g_i'(x) = \frac{dg_i(x)}{dx}$. Hence, if $g_i'(x)$ is a polynomial function in x and y_is, (12) is in the form of (9).

It turns out that $g_i'(x)$s are polynomial functions in many cases. We enumerate several commonly encountered uni-variable functions (*i.e.*, functions of only one variable) and their derivatives to x in (13).

$$\begin{gathered} y = e^x \Rightarrow (e^x)' = e^x = y \\ y = 1/(k+x) \Rightarrow (1/(k+x))' = -1/(x+k)^2 = -y^2 \\ y = \sqrt{x} \Rightarrow (\sqrt{x})' = 1/(2\sqrt{x}) = 1/(2y) \end{gathered} \tag{13}$$

It is seen that for $y = e^x$ and $y = 1/(k+x)$, the derivatives $y' = g'(x)$ map to a polynomial function of y. For $y = \sqrt{x}$, the derivative maps to a function of the form $y = 1/(k+x)$, and therefore, a further variable change of $z = 1/y$ will make the derivatives be polynomials of x, y, z.

Some other uni-variable functions also need to be handled by two variable changes. For example, if $\dot{x} = \sin(x)$, let $y_1 = \sin(x), y_2 = \cos(x)$, we obtain

$$\dot{x} = y_1, \quad \dot{y_1} = \cos(x)\dot{x} = y_2 y_1, \quad \dot{y_2} = -\sin(x)\dot{x} = -y_1^2. \tag{14}$$

Furthermore, if $g(x)$ is a composition of several uni-variable functions, *e.g.*, $g(x) = (g_2 \circ g_1)(x) = g_1(g_2(x))$, similar procedure can be conducted to make the differential equations of form (9). (As shown in (15), since $\frac{\partial y_1}{\partial y_2}$ is a polynomial function of y_1 and y_2; $\frac{\partial y_2}{\partial x}$ is a polynomial function of y_2 and x; $\dot{x}$ is a polynomial function of x, y_1.)

$$\text{Let } y_1 = g_1(y_2), y_2 = g_2(x) \Rightarrow \dot{y_1} = \frac{\partial y_1}{\partial y_2}\dot{y_2}, \dot{y_2} = \frac{\partial y_2}{\partial x}\dot{x} \tag{15}$$

For example, if $\dot{x} = \frac{1}{1+e^x}$, then by letting $y_1 = \frac{1}{1+y_2}$ and $y_2 = e^x$, we obtain a polynomial system

$$\dot{x} = y_1, \quad \dot{y_1} = -y_1^2 \dot{y_2} = -y_2 y_1^3, \quad \dot{y_2} = y_2 \dot{x} = y_2 y_1. \tag{16}$$

All the above derivations are also valid when x is a vector, in which case $g_i'(x)$ is a row vector instead of a scalar. Therefore, Theorem 4.1 follows:

Theorem 4.1. *By iteratively applying polynomialization by adding polynomial algebraic equations and taking Lie derivatives, a nonlinear system with nonlinear functions being compositions of functions in Table 2 can be polynomialized into a polynomial system in the form of (9).*

Notice that because of the function composition, the nonlinear functions in Table 2 actually covers a rich set of nonlinear functions that are almost enough for engineering problems (*e.g.*, smoothing functions such as the hyperbolic tangent function $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$).

Table 2: Commonly Used Nonlinear Functions

Function	Usage
$x_1^{\beta_1} \cdots x_N^{\beta_N}$	chemical rate equations MOSFET in saturation mode (x^2, x^α[21], α is rational)
$\frac{x}{k+x}$	chemical rate equations, smooth functions
e^x	diodes, BJTs, ion-channel models, smooth functions
$\sin(x), \cos(x)$	control systems (*e.g.*, x is the angle to be steered)

Another remark is that the polynomialization of a nonlinear system is *not unique*. Indeed, there should exist a *minimum* polynomial system that corresponds to the original system. The algorithm to find this *minimum* polynomial system could be devised by symbolic computational tools. Methods in logic synthesis might also be applicable. Though, this topic is out of the scope of this paper.

Despite the non-uniqueness, however, as long as the existence and uniqueness conditions of solutions to differential equations are satisfied, the polynomial system is *equivalent* to the original system since they have the same solution.

After the polynomialization, we obtain (9), a polynomial system of order M. Now we present two approaches to convert a polynomial system (9) to a QLDAE system (8).

4.3 Quadratic-Linearization by Adding Quadratic Algebraic Equations

Similar to the polynomialization procedure, we iteratively define new variables y, such that

$$y = x_1^{\beta_1} \cdots x_N^{\beta_N} / \hat{x} \tag{17}$$

is satisfied, and we add (17) into the original system. In (17), $\hat{x}$ is a variable chosen from the state variables of the expanded system at each iteration – they include all the original state variables, as well as the newly added state variables up to that iteration.

After several iterations, all the equations in the form of (17) can be re-written in a quadratic algebraic equation. In other words, a polynomial system can always be quadratic-linearized, as proved in the following theorem. As a byproduct, the theorem also provides an upper bound on the size of the QLDAE system derived from a polynomial system.

Theorem 4.2. *For a polynomial system of order M, the maximum number of state variables in the resulting QLDAE system does not exceed the number of state variables of* $[x^T, (x^T)^{\textcircled{2}}, (x^T)^{\textcircled{3}}, \cdots, (x^T)^{(\lceil \frac{M}{2} \rceil)}]^T$*, by adding quadratic algebraic equations,*

PROOF. Let $y = [x^T, (x^T)^{\textcircled{2}}, \cdots, (x^T)^{\textcircled{p}}]^T, p \in \mathbb{Z}^+$, then QLDAEs in y can generate any nonlinear functions of the form $G_1 y + G_2 y \otimes y$, which is essentially all the polynomial functions in x of order not greater than $2p$.

So, $M \leq 2p$, *i.e.*, $p = \lceil \frac{M}{2} \rceil$. □

4.4 Quadratic-Linearization by Taking Lie Derivatives

Similarly, taking Lie derivatives of polynomial differential equations can also lead to a QLDAE system, as proved in Theorem 4.3. However, the theoretical upper bound on the size of the resulting QLDAE system is more than that by adding quadratic algebraic equations, as shown in the following theorem.

Theorem 4.3. *For a polynomial system of order M, the maximum number of state variables in the resulting QLDAE system does not exceed the number of state variables of* $[x^T, (x^T)^{\textcircled{2}}, (x^T)^{\textcircled{3}}, \cdots, (x^T)^{\textcircled{M}}]^T$*, by adding differential equations that are derived by taking Lie derivatives of changed variables.*

PROOF. Let $y = x_1^{i_1} \cdots x_N^{i_N}$, where $\sum_{k=1}^{N} i_k \leq M$. So $y \in [x^T, (x^T)^{\textcircled{2}}, (x^T)^{\textcircled{3}}, \cdots, (x^T)^{\textcircled{M}}]^T$. Then the differential equation of y is derived by taking its Lie derivative

$$\frac{d}{dt}(x_1^{i_1}\cdots x_N^{i_N}) = \mathfrak{L}_{\dot{x}}(x_1^{i_1}\cdots x_N^{i_N}) = \sum_{k=1}^{N} x_1^{i_1}\cdots i_k x_i^{i_k-1}\left(\frac{d}{dt}x_i\right)\cdots x_N^{i_k}. \tag{18}$$

Since $\frac{d}{dt}x_i$ is a polynomial function of maximum order M in x and linear in u, therefore from (18), $\mathfrak{L}_{\dot{x}}(x_1^{i_1}\cdots x_N^{i_N})$ is a polynomial function of maximum order $2M-1$ in x and linear in u, *i.e.*, this is a QLDAE system in $[x^T, (x^T)^{\textcircled{2}}, \cdots, (x^T)^{\textcircled{M}}]^T$. □

5. QLMOR

In this section, we focus on the reduction step for QLDAEs (8). We start by performing a variational analysis to show the validity of applying Krylov-subspace methods. We also derive the moments of transfer functions of (8), which provides a guideline of how to choose the proper Krylov subspace for generating a smallest reduced model in the sense of moment-matching. The derivations in this section unifies previous works on polynomial systems and bilinear systems [3, 1], although they look similar.

5.1 Variational Analysis

The variational analysis [15] starts by assuming the QLDAE system is a combination of a series of homogeneous nonlinear subsystems, whose responses to $u(t)$ are $x_1(t)$, $x_2(t)$, *etc.*. That is, when input is $\alpha u(t)$, the response $x(t)$ is (19)

$$x(t) = \alpha x_1(t) + \alpha^2 x_2(t) + \alpha^3 x_3(t) + \alpha^4 x_4(t) + \cdots \tag{19}$$

Therefore, by plugging (19) into (8), and equating the terms corresponding to $\alpha^i, i = 1, 2, \cdots$, we obtain

$$\begin{aligned}
C\dot{x_1} =& G_1 x_1 + Bu,\\
C\dot{x_2} =& G_1 x_2 + G_2 x_1 \otimes x_1 + D_1 x_1 u,\\
C\dot{x_3} =& G_1 x_3 + G_2(x_1 \otimes x_2 + x_2 \otimes x_1) + D_1 x_2 u + D_2(x_1 \otimes x_1)u,\\
C\dot{x_4} =& G_1 x_4 + G_2(x_1 \otimes x_3 + x_2 \otimes x_2 + x_3 \otimes x_1)\\
& + D_1 x_3 u + D_2(x_1 \otimes x_2 + x_2 \otimes x_1)u.
\end{aligned} \tag{20}$$

From (20), if in the i-th set of differential equations (*i.e.*, equations where LHS is $\dot{x_i}$), we lump the terms without x_i to be pseudo inputs, then the original system can be viewed as a series of linear systems. Therefore, the Krylov subspace that guarantees that the projected system matches the moments of these linear systems is $\mathscr{K}(G_1^{-1}C, G_1^{-1}([B, G_2, D_1, D_2]))$, if G_1 is non-singular. However, the size of this subspace can easily exceed the size of the original system. Therefore, care must be taken to choose a useful subspace.

5.2 Matrix Transfer Functions

Following [15], we can show that transfer functions for (8) can be written as follows. (Only first three transfer functions are listed.)

$$\begin{aligned}
&H_1(s) = (sC - G_1)^{-1}B\\
&H_2(s_1, s_2) = ((s_1+s_2)C - G_1)^{-1}\frac{1}{2}\Big[G_2(H_1(s_1)\otimes H_1(s_2)\\
&\quad + H_1(s_2)\otimes H_1(s_1)) + D_1(H_1(s_1) + H_2(s_2))\Big]\\
&H_3(s_1, s_2, s_3) = ((s_1+s_2+s_3)C - G_1)^{-1}\\
&\quad \frac{1}{6}\Big\{G_2\Big[H_1(s_1)\otimes H_2(s_2, s_3) + H_1(s_2)\otimes H_2(s_1, s_3) + H_1(s_3)\otimes H_2(s_1, s_2)\\
&\quad + H_2(s_2, s_3)\otimes H_1(s_1) + H_2(s_1, s_3)\otimes H_1(s_2) + H_2(s_1, s_2)\otimes H_1(s_3)\Big]\\
&\quad + D_1\Big[H_2(s_1, s_2) + H_2(s_2, s_3) + H_2(s_1, s_3)\Big]\\
&\quad + D_2\Big[H_1(s_1)\otimes H_1(s_2) + H_1(s_1)\otimes H_1(s_3) + H_1(s_2)\otimes H_1(s_1)\\
&\quad + H_1(s_2)\otimes H_1(s_3) + H_1(s_3)\otimes H_1(s_1) + H_1(s_3)\otimes H_1(s_2)\Big]\Big\}
\end{aligned} \tag{21}$$

Based on (21), the Taylor expansion of these transfer functions can be derived. Correspondingly, we obtain the moments of the transfer functions, *i.e.*, coefficients of s^k for $H_1(s)$, $s_1^k s_2^l$ for $H_2(s_1, s_2)$, *etc.*. For example,

$$\begin{aligned}
H_1(s) =& \sum_{k=0}^{\infty} M_{1,k}s^k = \sum_{k=0}^{\infty} A^k R_1 s^k\\
H_2(s) =& \sum_{k=0}^{\infty}\sum_{l=0}^{\infty} M_{2,k,l}s_1^k s_2^l = \sum_{k=0}^{\infty} A^k R_1 (s_1+s_2)^k G_1^{-1}\\
& \frac{1}{2}\Big\{G_2\Big[\Big(\sum_{k=0}^{\infty} A^k R_1 s_1^k\Big)\otimes\Big(\sum_{k=0}^{\infty} A^k R_1 s_2^k\Big)\\
& + \Big(\sum_{k=0}^{\infty} A^k R_1 s_2^k\Big)\otimes\Big(\sum_{k=0}^{\infty} A^k R_1 s_1^k\Big)\Big] + D_1\Big[\sum_{k=0}^{\infty} A^k R_1 s_2^k + \sum_{k=0}^{\infty} A^k R_1 s_2^k\Big]\Big\},
\end{aligned} \tag{22}$$

where $A = G_1^{-1}C$, $R_1 = -G_1^{-1}B$, and $M_{1,k}$, $M_{2,k,l}$ are the moments for $H_1(s)$ and $H_2(s_1, s_2)$, respectively. Similar derivations can be performed for higher order moments.

An interesting point to note is that $H_2(s_1, s_2)$ is the same as the summation of the $H_2(s_1, s_2)$ of a bilinear system $C\dot{x} = G_1 x + D_1 x u + Bu$ and the $H_2(s_1, s_2)$ of a quadratic system $C\dot{x} = G_1 x + G_2 x \otimes x + Bu$.

5.3 Subspace Basis Generation

To illustrate how QLMOR generates the projection matrix, we derive the theorem and algorithm for $H_2(s_1, s_2)$ in the following. But they can be easily extended for higher-order moment-matching.

From (22), we can derive the vectors that are the coefficients for an n-th order moment. (*E.g.*, for $H_2(s_1, s_2)$, the n-th order moments refers to the set of coefficients for $s_1^k s_2^l, k + l = n$.) As an example, those vectors for $H_2(s_1, s_2)$ are listed in Table 3. We then have the following theorem:

Theorem 5.1. *Given a QLDAE system (8), suppose*

$$\begin{aligned}
R(V) =& \{A^i R_1, i \le q\}\bigcup\{A^i G_1^{-1} D_1 A^j R_1, i+j \le q\}\bigcup\\
& \{A^i G_1^{-1} G_2 (A^j R_1)\otimes(A^k R_1), i+j+k \le q, k \le j\}
\end{aligned} \tag{23}$$

Then if

$$\begin{aligned}
x = Vz,\quad \hat{C} = V^T C V,\quad & \hat{B} = V^T B,\\
\hat{G_1} = V^T G_1 V,\quad & \hat{G_2} = V^T G_2 V \otimes V,\\
\hat{D_1} = V^T D_1 V,\quad & \hat{D_2} = V^T D_2 V \otimes V,
\end{aligned} \tag{24}$$

the reduced system (25) matches the moments of $H_2(s_1, s_2)$ of the original system up to q-th order.

$$\hat{C}\dot{z} = \hat{G_1}z + \hat{G_2}z \otimes z + \hat{D_1}zu + \hat{D_2}(z\otimes z)u + Bu \tag{25}$$

PROOF. The proof is similar to that in [1]. □

Notice that the second and the third set of basis vectors in Theorem 5.1 count for the moments of the corresponding bilinear system and quadratic system, respectively – this result also follows from the above theorem by setting $G_2 = D_2 = 0$ and $D_1 = D_2 = 0$ in the QLDAE system.

A naive generation of the above subspace might lead to numerical stability problem because of the explicit computation of A^i terms. For quadratic systems, [1] computes the starting vectors directly and suggests to perform an orthogonalization between starting vectors. However, that can still lead to numerical stability problems since the starting vectors are computed explicitly.

Here we present a new algorithm to generate the basis for this subspace, as shown in Algorithm 1. The elementary operation in the algorithm is to generate an orthonormal projection matrix $V = [v_0, \cdots, v_q]$ for a Krylov subspace $\mathscr{K}_{q+1}(A, R)$. We employ Arnoldi algorithm[22] in our implementation.

In this algorithm, we avoid the direct computation of A^i in the starting vector for each Krylov subspace. We use $G_1^{-1}D_1 v_i$ instead of $G_1^{-1}D_1 A^i R_1$, and $G_1^{-1}G_2 v_i \otimes v_j$ instead of $G_1^{-1}G_2(A^i R_1)\otimes(A^j R_1)$, where v_i is the i-th vector in the Krylov subspace $\mathscr{K}(A, R_1)$.

We can prove that $R_1(V) = \{A^i R_1, i \le q\}\bigcup\{A^i G_1^{-1} D_1 A^j R_1, i+j \le q\}\bigcup\{A^i G_1^{-1} G_2 (A^j R_1)\otimes(A^k R_1), i+j+k \le q, k \le j\}$ and

Table 3: Moments of $H_2(s_1, s_2)$

Moments	Bilinear system	Quadratic system
0th order	$G_1^{-1}D_1R_1$	$G_1^{-1}G_2R_1 \otimes R_1$
1st order	$AG_1^{-1}D_1R_1$ $G_1^{-1}D_1AR_1$	$AG_1^{-1}G_2R_1 \otimes R_1$ $G_1^{-1}G_2(AR_1) \otimes R_1$
2nd order	$A^2G_1^{-1}D_1R_1$ $AG_1^{-1}D_1AR_1$ $G_1^{-1}D_1A^2R_1$	$A^2G_1^{-1}G_2R_1 \otimes R_1$ $AG_1^{-1}G_2(AR_1) \otimes R_1$ $G_1^{-1}G_2(A^2R_1) \otimes R_1$ $G_1^{-1}G_2(AR_1) \otimes (AR_1)$
3rd order	$A^3G_1^{-1}D_1R_1$ $A^2G_1^{-1}D_1AR_1$ $AG_1^{-1}D_1A^2R_1$ $G_1^{-1}D_1A^3R_1$	$A^3G_1^{-1}G_2R_1 \otimes R_1$ $A^2G_1^{-1}G_2(AR_1) \otimes R_1$ $AG_1^{-1}G_2(A^2R_1) \otimes R_1$ $G_1^{-1}G_2(A^3R_1) \otimes R_1$ $AG_1^{-1}G_2(AR_1) \otimes (AR_1)$ $G_1^{-1}G_2(A^2R_1) \otimes (AR_1)$

$R_2(V) = \{v_i, i \leq q\} \bigcup \{A^iG_1^{-1}D_1v_j, i+j \leq q\} \bigcup \{A^iG_1^{-1}G_2v_j \otimes v_k, i+j+k \leq q, k \leq j\}$ span the same subspace (proof by induction). Therefore, in the computation of the starting vector, no A^i terms are involved.

We can also prove that $v_{i_1} \otimes v_{j_1}$ is orthogonal to $v_{i_2} \otimes v_{j_2}$ if $i_1 \neq i_2$ or $j_1 \neq j_2$ (by definition of Kronecker product). Hence, the starting vectors have better numerical properties than [1]. However, it is not always the case that $G_1^{-1}G_2v_{i_1} \otimes v_{j_1}$ is orthogonal to $G_1^{-1}G_2v_{i_2} \otimes v_{j_2}$, even if $v_{i_1} \otimes v_{j_1}$ is orthogonal to $v_{i_2} \otimes v_{j_2}$. Therefore, one further improvement of this algorithm is to perform an orthonormalization of the starting vector to the basis already generated.

Proofs are omitted here due to page limits.

Algorithm 1 Generation of basis for subspace in Theorem 5.1

```
Generate matrix $V = [v_0, \cdots, v_q]$ for $\mathscr{K}_{q+1}(A, R_1)$
for $i = 0$ to $q$ do
    Generate matrix $W = [w_0, \cdots, w_{q-i}]$
    for $\mathscr{K}_{q-i+1}(A, G_1^{-1}D_1v_i)$; {subspace for moments of the bilinear system}
    $V = qr([V, W])$;
end for
for $i = 0$ to $q$ do
    for $j = 0$ to $\min(q-i, i)$ do
        Generate matrix $W = [w_0, \cdots, w_{q-i-j}]$ for
        $\mathscr{K}_{q-i-j+1}(A, G_1^{-1}G_2v_i \otimes v_j)$ {subspace for moments of the quadratic system}
        $V = qr([V, W])$;
    end for
end for
$V$ span the subspace that matches up to $q$-th order moments of $H_2(s_1, s_2)$
```

5.4 Local Passivity Preservation

The loss of passivity in [2, 3, 1] are mainly caused by Taylor approximation. QLMOR alleviates this problem. In the following proof, we show that the reduced system linearized at any $z_0 \in \mathbb{R}^q$, is a congruence transformation of the original system linearized at $x_0 = Vz_0 \in \mathbb{R}^n$. Therefore, from [9], the passivity of the linearized system is preserved if C and J_x (in equation (26)) are positive semi-definite.

PROOF. Let $f(x) = G_1x + G_2x \otimes x + D_1xu + D_2(x \otimes x)u + Bu$, and $\hat{f}(z) = \hat{G}_1z + \hat{G}_2z \otimes z + \hat{D}_1zu + \hat{D}_2(z \otimes z)u + \hat{B}u$.

Denote that the original system linearized at x_0 is in the form of

$$C\dot{x} = J_xx + J_uu, \tag{26}$$

and the reduced system linearized at $z_0 = V^Tx_0$ is in the form of

$$\hat{C}\dot{x} = \hat{J}_zx + \hat{J}_uu. \tag{27}$$

In (26) and (27), J_x, J_u, $\hat{J}_z$, $\hat{J}_u$ are Jacobian matrices of the linearized systems.

So the Jacobian matrix J_x of the original system at x_0 is

$$J_x = G_1 + G_2(I \otimes x_0 + x_0 \otimes I) + D_1u + D_2(I \otimes x_0 + x_0 \otimes I)u. \tag{28}$$

The Jacobian matrix $\hat{J}_z$ of reduced QLDAEs at $z_0 = V^Tx_0$ is

$$\begin{aligned}\hat{J}_z =& V^TG_1V + V^TG_2(V \otimes V)(I \otimes z_0 + z_0 \otimes I)\\ &+ V^TD_1Vu + V^TD_2(V \otimes V)(I \otimes z_0 + z_0 \otimes I)u\\ =& V^TG_1V + V^T[G_2(I \otimes x_0 + x_0 \otimes I)]V\\ &+ V^TD_1Vu + V^T[D_2(I \otimes x_0 + x_0 \otimes I)]V\\ =& V^TJ_xV.\end{aligned} \tag{29}$$

Similarly,

$$\begin{aligned}J_u =& D_1x_0 + D_2x_0 \otimes x_0 + B,\\ \hat{J}_u =& V^TD_1Vz_0 + V^TD_2(V \otimes V)(z_0 \otimes z_0) + V^TB\\ =& V^TD_1x_0 + V^TD_2x_0 \otimes x_0 + V^TB\\ =& V^TJ_u.\end{aligned} \tag{30}$$

Therefore, the linearized reduced system (27) is a congruence transformation of the original linearized system (26). □

5.5 Multi-Point Expansion

The multi-point Krylov subspace method [23] can be directly applied in QLMOR, and it potentially leads to a smaller and more accurate model for a specified frequency range of interest.

Also note that the quadratic-linearization procedure in Section 4 might render a QLDAE system where G_1 is *singular*. This can be a potential problem for generating Krylov subspaces at $s = 0$. The current workaround for this problem is to generate Krylov subspaces at $s = s_0, s_0 \simeq 0$.

6. EXPERIMENTAL RESULTS

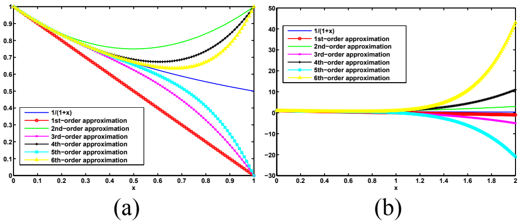
In this section, we illustrate the applicability of the QLMOR method presented in the previous sections, demonstrating its efficiency and accuracy on example circuits and systems.

6.1 A System with a $\frac{x}{1+x}$ Nonlinearity

As noted in Section 4, x, $x \otimes x$ and $\frac{x}{K+x}$ are common nonlinearities in biochemical rate equations. To test how QLMOR works, a random second-order polynomial system with a $\frac{x}{1+x}$ function was generated to mimic a biochemical reaction system:

$$\frac{d}{dt}x + G_1x + G_2x \otimes x + e_1\frac{10x_1}{1+x_1} + Bu = 0. \tag{31}$$

The size of (31) is 10, hence $G_1 \in \mathbb{R}^{10\times10}, G_2 \in \mathbb{R}^{10\times100}, B \in \mathbb{R}^{10\times1}, e_1 \in \mathbb{R}^{10\times1} = [1, 0, \cdots, 0]^T$. Notice that we manually make G_1 dense, non-symmetric, and make its eigenvalues uniformly distributed. Those factors all make MOR for this system more difficult than for many real systems that are normally sparse and symmetric.



(a) (b)

Figure 2: Polynomial approximations of $1/(1+x)$. Fig. 2(a) shows the interval $x \in [0,1]$. Fig. 2(b) shows the interval $x \in [0,2]$.

To compare with the QLMOR method, we first show that Taylor-expansion based methods [3, 1] are not applicable. The Taylor series of the function $\frac{1}{1+x}$ is

$$\frac{1}{1+x} = \sum_{i=0}^{\infty}(-x)^i = 1 - x + x^2 - x^3 + x^4 - x^5 + \cdots, (|x| < 1). \tag{32}$$

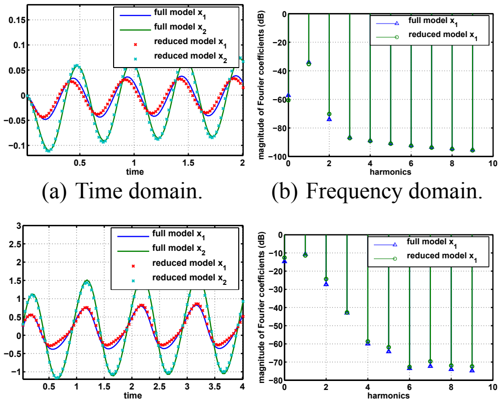
Notice that (32) converges only when $|x| < 1$ (also clearly seen in Fig. 2), making the Taylor approximation irrelevant and highly inaccurate for $|x| \geq 1$; as a result, any reduced model derived from this approximation cannot be accurate for $|x| \geq 1$. Indeed, the Taylor approximation model turns out to be an unstable system for polynomial orders from 2 to 7, *i.e.*, the polynomial model has a finite escape time[24]. As mentioned in [3], such approximations do not preserve properties such as stability and passivity, except by chance.

To apply the QLMOR method, we first perform quadratic-linearization of the original system by making the variable change $y = \frac{x_1}{1+x_1}$. Following the procedure in Section 4, we obtain a QLDAE system equivalent to (31):

$$\begin{aligned} \frac{d}{dt}x + G_1 x + G_2 x \otimes x + e_1 10y + Bu = 0, \\ -x_1 + y + x_1 y = 0. \end{aligned} \tag{33}$$

Notice that the size of (33) is 11, only one more than the original system.

Applying QLMOR on (33), a size 6 reduced model is generated that matches 4 moments of $H_1(s)$ and 2 moments of $H_2(s_1, s_2)$. We then apply a "small" and "fast" input signal $u(t) = \cos(4\pi t)$ so that small higher-order harmonic distortions are generated in the output waveform, as plotted in Fig. 3(a). To assess the accuracy of the reduced order model, we also plot the first few Fourier coefficients of $x_1(t)$ in Fig. 3(b). (Because $x_1(t)$ exhibits the largest nonlinearities compared to the other state variables, we focus on its waveforms to evaluate the accuracy of the reduced model.) We also excite this system by a "large" and "slow" input signal $u(t) = 10\cos(2\pi t)$, so that large higher-order harmonic distortions are generated in the output waveform, as plotted in Fig. 3(c). The speedup of the transient simulation of the reduced model over the original model is about 1.5× – although the model size is nearly halved, the reduced $\hat{G}_2$ matrix becomes dense, and therefore, the speedup is not huge. However, considering that the original model size is very small, a speedup of 1.5× is, in fact, not inconsiderable.



(a) Time domain. (b) Frequency domain.

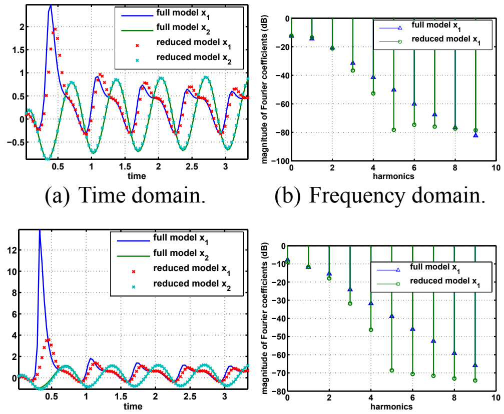
(c) Time domain. (d) Frequency domain.

Figure 3: Fig. 3(a), Fig. 3(b) show the time-domain waveforms of x_1, x_2 and Fourier coefficients of x_1, respectively, when $u(t) = \cos(4\pi t)$. Fig. 3(c), Fig. 3(d) show the time-domain waveforms of x_1, x_2 and Fourier coefficients of x_1, respectively, when $u(t) = 10\cos(2\pi t)$.

We further manipulate the system (31) to make it more nonlinear, and see how QLMOR behaves. In this case, we simply change the nonlinear term to be a function of x_2[4], *i.e.*,

$$\frac{d}{dt}x + G_1 x + G_2 x \otimes x + e_1 \frac{10x_2}{1+x_2} + Bu = 0. \tag{34}$$

The same model reduction procedure is repeated. Results are plotted in Fig. 4 for inputs $u(t) = 8\cos(3\pi t)$ and $u(t) = 10\cos(3\pi t)$. In this case, strong nonlinearities are present in the waveforms, but the QLMOR-reduced model still produces quite accurate results. An interesting point to note is that at time $t \simeq 0.5$, the reduced model does not quite capture a fast transient well. This is only to be expected, however, since MOR based on moment matching tends to reduce size by eliminating fast components in the system, while maintaining fidelity for slower components: observe that once the fast transient dies out, the QLMOR-reduced model is far more accurate. This inaccuracy at higher frequencies is also apparent from the harmonics shown in Fig. 4(d).



(a) Time domain. (b) Frequency domain.

(c) Time domain. (d) Frequency domain.

Figure 4: Fig. 4(a), Fig. 4(b) show the time-domain waveforms of x_1, x_2 and Fourier coefficients of x_1, respectively, when $u(t) = 8\cos(3\pi t)$. Fig. 4(c), Fig. 4(d) show the time-domain waveforms of x_1, x_2 and Fourier coefficients of x_1, respectively, when $u(t) = 10\cos(3\pi t)$.

6.2 Nonlinear Transmission Line Circuit

The nonlinear transmission line circuit shown in Fig. 5 [4] is possibly the most widely used circuit for testing and evaluating any new, experimental nonlinear model reduction technique. All resistors and capacitors are set to 1 and the diode I-V characteristic is $i_D = e^{40v_D} - 1$. The input is set to the current source $i = u(t)$; the output is the voltage at node 1.

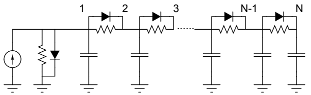


Figure 5: Nonlinear transmission line circuit [4].

The modified nodal equations for this circuit are

$$\begin{aligned} \dot{v_1} &= -2v_1 + v_2 + 2 - e^{40v_1} - e^{40(v_1 - v_2)} + u(t), \\ \dot{v_i} &= -2v_i + v_{i-1} + v_{i+1} + e^{40(v_{i-1} - v_i)} - e^{40(v_i - v_{i+1})}, \quad 2 \leq i \leq N-1, \\ \dot{v_N} &= -v_N + v_{N-1} - 1 + e^{40(v_{N-1} - v_N)}. \end{aligned} \tag{35}$$

To perform quadratic-linearization of (35), if we define new variables to be uni-variable functions, we have to at least make the variable changes $y_{i1} = e^{40v_i}$ and $y_{i2} = e^{-40v_i}$, which results in a third-order polynomial system – the expanded system is at least of size $3N$. However, note that if the state variables are set to be $v_1, v_{i,i+1}, 1 \leq i \leq N-1$ ($v_{i,i+1} = v_i - v_{i+1}$), we obtain

$$\begin{aligned} \dot{v_1} &= -v_1 - v_{12} + 2 - e^{40v_1} - e^{40v_{12}} + u(t), \\ \dot{v_{12}} &= -v_1 - 2v_{12} + v_{23} + 2 - e^{40v_1} - 2e^{40v_{12}} + e^{40v_{23}} + u(t), \\ \dot{v_{i,i+1}} &= -2v_{i,i+1} + v_{i-1,i} + v_{i+1,i+2} + e^{40v_{i-1,i}} - 2e^{40v_{i,i+1}} \\ &\quad + e^{40v_{i+1,i+2}}, \quad (2 \leq i \leq N-2), \\ \dot{v_{N-1,N}} &= -2v_{N-1,N} + v_{N-2,N-1} + 1 + e^{40v_{N-2,N-1}} - 2e^{40v_{N-1,N}}. \end{aligned} \tag{36}$$

Now, making the variable change $y_1 = e^{40v_1} - 1$ and $y_i = e^{40v_{i-1,i}} - 1, 2 \leq i \leq N$, the differential equations for the y_is can

[4] We observe in simulations that the outputs of this system exhibit larger high-order harmonic distortions than (33).

be written as

$$\begin{aligned}
\dot{y_1} &= 40(y_1+1)(-v_1-v_{12}-y_1-y_2+u(t)),\\
\dot{y_2} &= 40(y_2+1)(-v_1-2v_{12}+v_{23}-y_1-2y_2+y_3+u(t))a,\\
\dot{y_i} &= 40(y_i+1)(-2v_{i-1,i}+v_{i-2,i-1}+v_{i,i+1}+y_{i-1}-2y_i+y_{i+1}),\\
\dot{y_N} &= 40(y_N+1)(-2v_{N-1,N}+v_{N-2,N-1}+y_{N-1}-2y_N).
\end{aligned} \tag{37}$$

Therefore, (36) and (37) together are already in QLDAE form with size $2N$. This size is far less than the system size that would result from a second-order Carleman bilinearization [3] (*i.e.*, $N+N^2$). This example illustrates that if we use *branch voltages* as state variables, a much smaller QLDAE system can result. Intuitively speaking, this is reasonable since the device models are usually defined in terms of *branch voltages* and *branch currents*. This suggests that sparse tableau like circuit equation formulations [25], where the state variables are branch voltages and currents, can be useful for quadratic-linearization.

We apply QLMOR on this nonlinear transmission line circuit with $N=10$ and compare against weakly-nonlinear Taylor-expansion-based MOR methods, setting the reduced size to 7.[5] To drive the circuit sufficiently hard to produce large harmonic distortions, we apply $u(t) = 1.8\cos(2\pi \times 0.1t)$ and plot waveforms in Fig. 6. As expected, the Taylor-based weakly nonlinear reduced model matches the Taylor approximation of the original system well; however, the original Taylor approximation itself does of course not match the full model, because of its validity is limited to "small" inputs. QLMOR, on the other hand, produces a reduced model which matches the original system well, the relative error being about 1%.

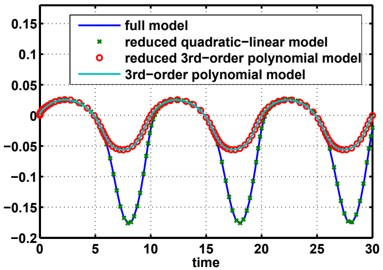


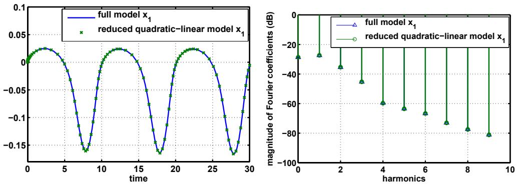
Figure 6: Time-domain waveforms (x_1) of full model, quadratic-linear reduced model, 3rd-order polynomial reduced model and 3rd-order polynomial model.

To illustrate that QLMOR scales to larger system sizes, we apply QLMOR to nonlinear transmission line circuits of 50,100,150,200 stages. To make a fair comparison, we tune QLMOR-reduced model sizes such that in several simulations, the ratio of the maximum absolute error of the time-domain waveform to its amplitude is within 1%. As a result, we produce reduced models of size 11, 15, 18, 20 corresponding to 50,100,150,200 stages, respectively. Again, we excite large harmonic distortions in the simulation, and a typical transient waveform is shown in Fig. 7 (for the circuit with 50 stages).

As comparison, Taylor-based weakly nonlinear MOR did not succeed in producing a reduced model on account of shortage of memory for computing $V^T G_3(V\otimes V\otimes V)$. For example, when $N=50, q=11, V\in\mathbb{R}^{50\times 11}$, we have $V\otimes V\otimes V\in\mathbb{R}^{(50\times 11)^3}$. Hence, suppose the size of a double-precision number is 8 bytes, $V\otimes V\otimes V$ requires a minimum memory of $(50\times 11)^3 \times 8$ Bytes, which is 1.2396 GB.

7. CONCLUSION

Prior polynomial-expansion-based nonlinear model reduction methods, which rely on Taylor approximations, are limited in applicability because they scale poorly with the number of polynomial terms, destroy passivity, and can suffer from numerical stability during subspace basis generation. We have developed a novel polynomial-based reduction technique, QLMOR, that alleviates these problems by (1) rewriting strongly nonlinear systems in quadratic-linear (QLDAE) form without *any* approximation, (2) bounding the computational cost by $O(q^3)$ by using a QLDAE form for the reduced model, (3) employing congruence transformations to preserve local passivity, and (4) using a numerically stable basis generation algorithm. Initial validation demonstrates good matches between QLMOR-reduced models and the originals, including in cases where Taylor approximation based weakly nonlinear methods fail or are very inaccurate.



(a) Time domain. (b) Frequency domain.

Figure 7: Fig. 7(a), Fig. 7(b) show the time-domain waveforms and Fourier coefficients of x_1, respectively, when $u(t) = 1.8\cos(2\pi \times 0.1t)$. The size of the full model is 50; the size of the reduced model is 11.

8. REFERENCES

[1] Peng Li and L.T. Pileggi. Compact reduced-order modeling of weakly nonlinear analog and rf circuits. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 24(2):184–203, Feb. 2005.

[2] J. Roychowdhury. Reduced-order modelling of time-varying systems. *IEEE Trans. Ckts. Syst. – II: Sig. Proc.*, 46(10), November 1999.

[3] J. R. Phillips. Projection-Based Approaches for Model Reduction of WeaklyNonlinear Time-Varying Systems. *IEEE Transactions on Computer-Aided Design*, 22(2):171–187, 2003.

[4] M. Rewienski and 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*, 22(2), February 2003.

[5] C. Gu and J. Roychowdhury. ManiMOR: Model Reduction via Projection onto Nonlinear Manifolds, with Applications to Analog Circuits and Biochemical Systems. In *Proceedings of the IEEE International Conference on Computer Aided Design*, pages 85–92, 10-13 Nov. 2008.

[6] E.J. Grimme. *Krylov Projection Methods for Model Reduction*. PhD thesis, University of Illinois, EE Dept, Urbana-Champaign, 1997.

[7] L.T. Pillage and R.A. Rohrer. Asymptotic waveform evaluation for timing analysis. *IEEE Transactions on Computer-Aided Design*, 9:352–366, April 1990.

[8] P. Feldmann and R.W. Freund. Efficient Linear Circuit Analysis by Padé Approximation via the Lanczos Process. *IEEE Transactions on Computer-Aided Design*, 14(5):639–649, May 1995.

[9] A. Odabasioglu, M. Celik, and L.T. Pileggi. PRIMA: passive reduced-order interconnect macromodellingalgorithm. In *Proceedings of the IEEE International Conference on Computer Aided Design*, pages 58–65, November 1997.

[10] L. Daniel J. R. Phillips and M. Silveira. Guaranteed Passive Balancing Transformations for Model Order Reduction. *Proceedings of the IEEE Design Automation Conference*, 2002.

[11] Z. Bai and D. Skoogh. Krylov Subspace Techniques for Reduced-Order Modeling of Large-Scale Dynamical Systems. *Applied Numerical Mathematics*, 43:9–44, 2002.

[12] L. Daniel and J. R. Phillips. Model Order Reduction for Strictly Passive and CausalDistributed Systems. *Proceedings of the IEEE Design Automation Conference*, 2002.

[13] S. H. Strogatz. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering*. Perseus Books Group, January 2001.

[14] L.O. Chua. The genesis of chua's circuit. Technical Report UCB/ERL M92/1, EECS Department, University of California, Berkeley, 1992.

[15] W. Rugh. *Nonlinear System Theory - The Volterra-Wiener Approach*. Johns Hopkins Univ Press, 1981.

[16] M. Rathinam and L. R. Petzold. A new look at proper orthogonal decomposition. *SIAM J. Numer. Anal.*, 41(5):1893–1925, 2003.

[17] S.K. Tiwary and R.A. Rutenbar. Faster, parametric trajectory-based macromodels via localized linear reductions. In *Computer-Aided Design, 2006. ICCAD '06. IEEE/ACM International Conference on*, pages 876–883, Nov. 2006.

[18] B. N. Bond and L. Daniel. Stabilizing schemes for piecewise-linear reduced order models via projection and weighting functions. In *Proc. IEEE/ACM International Conference on Computer-Aided Design ICCAD 2007*, pages 860–867, 4–8 Nov. 2007.

[19] N. Dong and J. Roychowdhury. Piecewise Polynomial Nonlinear Model Reduction. *Proceedings of the IEEE Design Automation Conference*, 2003.

[20] J.R. Phillips. Projection frameworks for model reduction of weakly nonlinear systems. In *Design Automation Conference, 2000. Proceedings 2000. 37th*, pages 184–189, June 5-9, 2000.

[21] T. Sakurai and A.R. Newton. Alpha-power law mosfet model and its applications to cmos inverter delay and other formulas. *Solid-State Circuits, IEEE Journal of*, 25(2):584–594, Apr 1990.

[22] D.L. Boley. Krylov space methods on state-space control models. *IEEE Trans. Ckts. Syst. – II: Sig. Proc.*, 13(6):733–758, 1994.

[23] J. Phillips. Model Reduction of Time-Varying Linear Systems Using Approximate Multipoint Krylov-Subspace Projectors. In *Proceedings of the IEEE International Conference on Computer Aided Design*, November 1998.

[24] S. Sastry. *Nonlinear Systems - Analysis, Stability and Control*. Springer, 1999.

[25] G. Hachtel, R. Brayton, and F. Gustavson. The sparse tableau approach to network analysis and design. *Circuit Theory, IEEE Transactions on*, 18(1):101–113, Jan 1971.

[5] The size $q=7$ is obtained by trial-and-error. With q larger than 7, QLMOR gives almost the same result as that of $q=7$.