Operator-based Model-Order Reduction of Linear Periodically Time-Varying Systems

Yayun Wan, Jaijeet Roychowdhury
ECE Department
University of Minnesota
Minneapolis, MN, USA
yayun,jr@ece.umn.edu

ABSTRACT

Linear periodically time-varying (LPTV) abstractions are useful for a variety of communication and computer subsystems. In this paper, we present a novel operator-based model-order reduction (MOR) algorithm for reducing large LPTV systems to smaller ones, a capability useful for system-level performance analysis. Our procedure is based on generalizing existing matrix-based Krylov-subspace algorithms to arbitrary function-space operators. Practical benefits of our approach include significantly enhanced algorithm and code modularity, compared to previous LPTV-MOR approaches based on a-priori discretization. We demonstrate the use of the proposed technique on several circuit examples.

Categories and Subject Descriptors: B.7.2 [Integrated Circuits]: Design Aids-simulation

General Terms: Algorithms

Keywords: LPTV systems, model-order reduction, operator, modularity

1. INTRODUCTION

In communication systems, many subsystems can be usefully modeled as Linear Periodically Time-Varying (LPTV) systems. Examples include up/down-conversion circuits in modulators and demodulators [15], certain phase detectors in PLLs [1], sampled-data filters [3], free-running oscillators [5], *etc.*. Recently, LPTV abstractions have also been shown to be useful for the digital noise estimation problem [16].

Macromodels of LPTV subsystems have, so far, been typically constructed manually in practice. However, manual abstraction needs to be aided by extensive nonlinear simulation; parameters of interest, such as poles, may not be easy to obtain; and manual abstraction may miss important nonidealities, especially those relating to interactions between sub-systems. In short, manual abstraction can be computationally expensive, time-consuming and of varying fidelity.

Therefore, it is very desirable to be able to abstract macromodels directly and automatically from full SPICE-level circuits without manual intervention. Automation of the macromodeling process has many advantages: it does not require additional modeling expertise from designers, it can provide parameters of interest directly, tradeoffs between accuracy and speed can be made easily, *etc.*.

Techniques for automatically generating macromodels have already met with considerable success for linear time-invariant (LTI) systems. These are often called model-order reduction (MOR) algorithms, and a variety of methods, such as PRIMA [12] and PVL [8], are well established. MOR methods are based on projecting the LTI system into subspaces of lower dimension, often using Krylov-subspace-based methods. Such reduced models match the input-output transfer functions of the original system to a prescribed, acceptable accuracy.

For LTI systems, the input-output transfer function can be expressed entirely in terms of matrix operations. As we will outline in Section 3, most existing Krylov-subspace-based algorithms rely on operations on matrices (we will call these *matrix-based MOR algorithms*). In contrast to LTI systems, the input-output transfer functions of LPTV systems cannot be expressed in terms of simple matrix operations alone. Instead, they require manipulations with *differential operators* involving time-varying matrices, as we will show in Section 2.

It has been shown, however, that LTI MOR techniques can be leveraged for the problem of reducing LPTV systems to smaller ones. By discretizing the periodic time-variation of the system using a finite basis (either time or frequency domain), [14, 11] showed that LPTV systems could be recast as artificial LTI systems in which extra "outputs", corresponding to coefficients of the time-varying basis, appear. The artificial LTI systems, after reduction via LTI MOR techniques, were recast as smaller LPTV systems, thus completing the LPTV reduction procedure. These prior techniques use matrix-based MOR methods for the actual reduction; they rely on a fixed, a-priori discretization of the time-varying differential operators to convert the LPTV system into an LTI one.

This approach, compared to the *operator-based MOR* alternatives we explore in this paper, imposes several restrictions on the model-reduction process. The separation of the LPTV system's parametric time scale and its input time scale is not preserved through the reduction process. Operator discretization is intimately enmeshed with the model-reduction process, which affects code structuring and modularity, and limits the flexibility using different discretization bases during the reduction process.

The concept of operator-based MOR is not new; for example, it has been applied in the context of transmission lines [13, 7]. The key step is to modify the internals of Krylov-subspace methods to use general function-space operators, rather than simply matrices.

DAC 2005, June 13–17, 2005, Anaheim, California, USA.

Thus, discretization of the operators before Krylov-subspace reduction can be avoided.

In this work, we present the use and application of operator-based MOR techniques for LPTV system reduction. One of the main advantages of using operator-based MOR, rather than a-priori discretization of the LPTV problem, is that the separation of the LPTV system's parametric time variation and its input (signal) time scale is preserved through the reduction process. Retaining this separation through the internals of Krylov-subspace model-order reduction confers a key advantage: the discretization basis can be changed dynamically during the model-order reduction process. Not only does this have advantages for MOR accuracy, but it enables significantly greater code modularity, as well as algorithm flexibility, than possible with a-priori discretization. For example, by using operator-based MOR, the number of time-points or harmonics used for periodic time-varying discretization can be dynamically changed at each Krylov reduction step; indeed, one can switch between time- and frequency-domain discretizations, if so desired, without any code changes necessary at the "upper level" (*i.e.*, the operator-LPTV-MOR routine) of the procedure. Such flexibility is difficult or impossible to achieve in practical implementations of a-priori discretized LPTV MOR algorithms [14, 11]. This flexibility enables unified LPTV macromodeling of system blocks with different discretization bases, with implications for hierarchical LPTV macromodeling, especially important for digital interference macromodeling [16].

The remainder of the paper is structured as follows. We review the derivation of transfer functions of LPTV systems using multiple time variables and multirate partial differential equations (MPDEs) in Section 2. In Section 3, a brief outline of Krylov-subspace methods for matrices is provided. In Section 4, we extend the algorithm for matrix-based MOR to construct an operator-based MOR procedure. We apply the operator-based MOR approach to reduce LPTV systems. Finally, in Section 5, three examples, including a simple up-converter and a double-balanced mixer, are reduced using operator-LPTV-MOR and results presented.

2. LPTV SYSTEMS

Consider a LPTV system driven by a large periodic signal $\vec{l}(t)$ and a small input signal $u(t)$. The output of the system is $z_y(t)$. For simplicity, we consider only one input and one output here, *i.e.*, $u(t)$ and $z_y(t)$ are scalars. The system is described by the following differential-algebraic equations (DAEs):

$$\begin{aligned} \frac{\mathrm{d}\vec{q}(\vec{y}(t))}{\mathrm{d}t} + \vec{f}(\vec{y}(t)) &= \vec{l}(t) + \vec{b}u(t), \\ z_y(t) &= \vec{d}^T \vec{y}(t), \end{aligned} \tag{1}$$

where $\vec{y}(t)$ is the state vector containing node voltages and branch currents; $\vec{q}(\vec{y})$ and $\vec{f}(\vec{y})$ contain charge/flux and resistive terms, respectively; $\vec{b}$ and $\vec{d}$ are vectors linking the input and output to the internal nodes of the circuit. By using the MPDE formalism and linearization methods introduced in [14], and assigning $\vec{l}(t)$ to time scale t_1 and $u(t)$ to time scale t_2, we obtain:

$$\begin{aligned} \frac{\partial[C(t_1)\vec{x}(t_1,t_2)]}{\partial t_1} + \frac{\partial[C(t_1)\vec{x}(t_1,t_2)]}{\partial t_2} + G(t_1)\vec{x}(t_1,t_2) &= \vec{b}u(t_2), \\ z(t_1,t_2) &= \vec{d}^T \vec{x}(t_1,t_2), \end{aligned} \tag{2}$$

where $\vec{x}(t_1,t_2)$, $z(t_1,t_2)$ are bivariate forms of the small-signal versions of $\vec{y}(t)$ and $z_y(t)$, respectively, linearized around $\vec{y} = \vec{y}^*(t_1)$, where $u(t_2) = 0$. $C(t_1)$ and $G(t_1)$ are defined as:

$$C(t_1) = \frac{\mathrm{d}\vec{q}(\vec{y}(t_1,t_2))}{\mathrm{d}\vec{y}(t_1,t_2)}|_{\vec{y}=\vec{y}^*(t_1)}, \tag{3}$$

$$G(t_1) = \frac{\mathrm{d}\vec{f}(\vec{y}(t_1,t_2))}{\mathrm{d}\vec{y}(t_1,t_2)}|_{\vec{y}=\vec{y}^*(t_1)}. \tag{4}$$

Applying Laplace transforms to (2) with respect to t_2, we obtain

$$\begin{aligned} \frac{\partial[C(t_1)\vec{X}(t_1,s_2)]}{\partial t_1} + s_2C(t_1)\vec{X}(t_1,s_2) + G(t_1)\vec{X}(t_1,s_2) &= \vec{b}U(s_2), \\ Z(t_1,s_2) &= \vec{d}^T \vec{X}(t_1,s_2). \end{aligned} \tag{5}$$

Using (5), we can obtain an expression for the input-output relationship between $U(s_2)$ and $Z(t_1,s_2)$:

$$H(t_1,s_2) = \vec{d}^T [\frac{D}{dt_1}[\cdot] + s_2C(t_1) + G(t_1)]^{-1}[\vec{b}]. \tag{6}$$

Observe that (6) expresses the transfer function of a LPTV system in *operator form*, where $\frac{D}{dt_1}[\cdot]$ is a differential operator, defined as

$$\frac{D}{dt_1}[\cdot] = \frac{d[C(t_1)\cdot]}{dt_1}. \tag{7}$$

(6) can be re-expressed in a form formally similar to the following form for LTI systems, useful for model reduction using Krylov methods , as explored further in Section 3:

$$H(s) = \vec{d}^T (I + sL)^{-1}\vec{r}. \tag{8}$$

For LTI systems, I and L are matrices, *i.e.*, linear operators on vector spaces of dimension n. To re-express (6) in similar form, we define the operators

$$\begin{aligned} \mathfrak{I}_{t_1}[\cdot] &= (\frac{D}{dt_1}[\cdot] + G(t_1)), \\ \mathfrak{L}_{t_1}[\cdot] &= \mathfrak{I}_{t_1}^{-1}[C(t_1)\cdot], \end{aligned} \tag{9}$$

and the vector

$$\vec{r}(t_1) = \mathfrak{I}_{t_1}^{-1}(t_1)[\vec{b}]. \tag{10}$$

Subscripts "$_{t_1}$" indicate that the operators subsume operations with respect to time scale t_1. Using these, the transfer function (6) can be written as

$$\begin{aligned} H(t_1,s_2) &= \vec{d}^T (\mathfrak{I}_{t_1}[*] + s_2C(t_1))^{-1}\vec{b} \\ &= \vec{d}^T (I + s_2\mathfrak{I}_{t_1}^{-1}[C(t_1)*])^{-1}\mathfrak{I}_{t_1}^{-1}[\vec{b}] \\ &= \vec{d}^T (I + s_2\mathfrak{L}_{t_1}[*])^{-1}\vec{r}(t_1). \end{aligned} \tag{11}$$

This is the operator form of the LPTV system transfer function, using which the proposed operator-LPTV-MOR algorithm will be developed in Section 4.

3. KRYLOV-SUBSPACE-BASED MOR

Compared to alternative model-order reduction techniques such as truncated/balanced realizations, Krylov-subspace methods require much less computation and memory. For this reason, they are practical for large systems, *i.e.*, of size exceeding 10^4. The main reason why Krylov-subspace methods are more efficient is that they avoid matrix-matrix multiplications and singular-value decompositions (SVDs) (which have quadratic and cubic computational complexity), relying instead solely on matrix-vector multiplications. In this section, we will review matrix-based Krylov-subspace algorithms using the Arnoldi process for LTI systems, preparatory to extending them to the operator case in Section 4.

The transfer function of a LTI system can be written as

$$H(s) = \vec{d}^T (I + sL)^{-1} \vec{r}. \tag{12}$$

Using the matrix L and the "starting vector" $\vec{r}$, a Krylov subspace K is defined as the space spanned by the vectors $L^i\vec{r}$, *i.e.*,

$$K = \text{span}\{\vec{r}, L\vec{r}, L^2\vec{r}, L^3\vec{r}, \cdots, L^n\vec{r}, \cdots\}. \tag{13}$$

It is well known that the space K can be expressed in terms of an alternative basis, *i.e.*,

$$K = \text{span}\{\vec{q}_0, \vec{q}_1, \vec{q}_2, \vec{q}_3, \cdots, \vec{q}_{n-1}, \cdots\} \tag{14}$$

where $\vec{q}_k$ $(k = 0, 1, \cdots)$ is obtained via a modified Gram-Schmidt procedure known as the *Arnoldi process*, summarized below in (15):

$$\begin{aligned}
&\vec{q}'_0 = \vec{r} \qquad \vec{q}_0 = \frac{\vec{q}'_0}{||\vec{q}'_0||} \\
&\vec{q}'_1 = L\vec{q}_0 - < L\vec{q}_0, \vec{q}_0 > \vec{q}_0 \qquad \vec{q}_1 = \frac{\vec{q}'_1}{||\vec{q}'_1||} \\
&\vdots \\
&\vec{q}'_m = L\vec{q}_{m-1} - < L\vec{q}_{m-1}, \vec{q}_0 > \vec{q}_0 - ... \\
&\quad - < L\vec{q}_{m-1}, \vec{q}_{m-1} > \vec{q}_{m-1}
\end{aligned} \tag{15}$$

Using the $\vec{q}_k$'s, $(k = 1, \cdots, m)$ obtained in (15), the projection matrix V_m can be defined as

$$V_m = \begin{pmatrix} \vdots & & \vdots \\ q_0 & \cdots & q_{m-1} \\ \vdots & & \vdots \end{pmatrix}. \tag{16}$$

In the Arnoldi process (15), the correlation coefficients are defined as

$$r_{kk} = ||\vec{q}'_k|| \qquad r_{jk} = < L\vec{q}_{k-1}, \vec{q}_j >, \tag{17}$$

and using the correlation coefficients r_{ij}'s $(i = 0...m-1, j = 1...m)$ in (17), the upper Hessenberg matrix H_m is defined as

$$H_m = \begin{pmatrix} r_{01} & \cdots & r_{0m} \\ r_{11} & \cdots & r_{1m} \\ & \ddots & \vdots \\ & r_{m-1,m-1} & r_{m-1,m} \end{pmatrix}. \tag{18}$$

The Arnoldi process described in (15) can be re-expressed by using the matrices in (16) and (18) as:

$$LV_m = V_m H_m + r_{mm} \vec{q}_m \vec{e}_m^T, \tag{19}$$

where $\vec{e}_m$ is a $m \times 1$ column vector with only the last term as one and all the others as zero. By multiplying V_m^T from left to both sides of (19), we obtain

$$V_m^T L V_m = H_m, \tag{20}$$

because the second term in (19) multiplied by V_m^T from left is zero. With $m < n$, by putting the above transformation into the transfer function of the original LTI system (12), we obtain

$$\begin{aligned}
\hat{H}(s) &= \vec{d}^T V_m (I + sH_m)^{-1} V_m^T \vec{r} \\
&= r_{00} \vec{d}^T V_m (I + sH_m)^{-1} \vec{e}_1
\end{aligned} \tag{21}$$

which is the transfer function of the reduced model and $\vec{e}_1$ is a $m \times 1$ column vector with only the first term as one and all the others as zero.

Implicitly in the Arnoldi process, the first m moments of the original system (12) which is defined as

$$m_k = \vec{d}^T L^k r, \tag{22}$$

where, $k = 0...m-1$, match the first m moments of the reduced model (21) which is given as

$$(r_{00} V_m^T \vec{d})^T H_m^k \vec{e}_1 \tag{23}$$

where, $k = 0...m-1$.

The reduced system can be constructed by using the projection matrix V_m and H_m as

$$\begin{aligned}
H_m \dot{\vec{z}}(t) + \vec{z}(t) &= \vec{e}_1 u(t) \\
y(t) &= r_{00} \vec{d}^T V_m \vec{z}(t)
\end{aligned} \tag{24}$$

where $\vec{z}(t)$ is a $m \times 1$ vector.

In addition, there are many interpolation techniques [2] following the projection step.

Based on the techniques introduced in [14], LPTV system transfer function (6) can be expressed as an artificial LTI system in the following form

$$\vec{H}_{FD}(s_2) = D_{FD}^T (I + s_2 L_{FD})^{-1} \vec{R}_{FD}, \tag{25}$$

where the size of the system is expanded N times by using N harmonics. The artificial LTI system can be reduced by using the above introduced Arnoldi process. This approach has been successful in obtaining the input-output relationship with acceptable accuracy. However, the discretization is enmeshed in the model-order reduction procedure, which adds limitations in the modularity and coding flexibility of the model-order reduction procedure.

4. OPERATOR-MOR FOR LPTV SYSTEMS

In this section, we generalize the MOR procedure of the previous section to apply to operators. Let $\mathfrak{L}$ be a linear operator (*e.g.*, as show in (9), $\mathfrak{L}_{t_1}[\cdot]$) with domain and range consisting of n-dimensional vectors of *T-periodic functions*. Note that the domain and range are infinite-dimensional. Define an operator-Arnoldi process to be:

$$\begin{aligned}
&\vec{q}'_0(t_1) = \vec{r}(t_1) \qquad \vec{q}_0(t_1) = \frac{\vec{q}'_0(t_1)}{||\vec{q}'_0(t_1)||} \\
&\vec{q}'_1(t_1) = \mathfrak{L}_{t_1}[\vec{q}_0(t_1)] - < \mathfrak{L}_{t_1}[\vec{q}_0(t_1)], \vec{q}_0(t_1) > \vec{q}_0(t_1) \\
&\vec{q}_1(t_1) = \frac{\vec{q}'_1(t_1)}{||\vec{q}'_1(t_1)||} \\
&\qquad \vdots \\
&\vec{q}'_m(t_1) = \mathfrak{L}_{t_1}[\vec{q}_{m-1}(t_1)] - < \mathfrak{L}_{t_1}[\vec{q}_{m-1}(t_1)], \vec{q}_0(t_1) > \vec{q}_0(t_1) \\
&\qquad - .... - < \mathfrak{L}_{t_1}[\vec{q}_{m-1}(t_1)], \vec{q}_{m-1}(t_1) > \vec{q}_{m-1}(t_1).
\end{aligned} \tag{26}$$

It remains to define the inner-products and norms used above. Noting that we are operating on function spaces, we use standard L^2 inner-products and norms, similar to [13] and [7].

$$||\vec{q}'_k(t_1)|| = \sqrt{\frac{1}{T} \int_0^T \vec{q}'^H_k(t_1) \vec{q}'_k(t_1) dt_1} \tag{27}$$

$$< \mathfrak{L}_{t_1}[\vec{q}_k(t_1)], \vec{q}_k(t_1) > = \frac{1}{T} \int_0^T \vec{q}^H_k(t_1) \mathfrak{L}_{t_1}[\vec{q}_k(t_1)] dt_1. \tag{28}$$

From equation (9)-(11), we obtain $\mathfrak{L}_{t_1}[\vec{q}_k(t_1)]$ and $\vec{r}(t_1)$ as follows:

$$\begin{aligned}\vec{q}'_{k+1} &= \mathfrak{L}_{t_1}[\vec{q}_k(t_1)] \\ &= \mathfrak{I}_{t_1}^{-1}[C(t_1)\vec{q}_k(t_1)], \\ (\frac{D}{dt_1}+G(t_1))\vec{q}'_{k+1} &= C(t_1)\vec{q}_k(t_1),\end{aligned} \quad (29)$$

and

$$(\frac{D}{dt_1}+G(t_1))\vec{r}(t_1) = \vec{b}. \quad (30)$$

Since here we are dealing with periodic cases in which $C(t_1)$, $G(t_1)$ are all periodic, we can use harmonic balance (HB) [14], finite difference time domain (FDTD) [14] or shooting methods, to solve (29) and (30). This leads to the following structure for the operator LPTV MOR procedure:

Algorithm: Operator-LPTV-MOR

1. Let $\vec{q}'(t_1) = \vec{r}(t_1)$, and $k = 0$.
2. $\vec{q}_k(t_1) = \vec{q}'(t_1)/||\vec{q}'(t_1))||$. The norm is defined in (27).
3. Solve $\mathfrak{L}_{t_1}[\vec{q}_k(t_1)] \Rightarrow \vec{q}'(t_1)$. **Note that details of the solution technique can change between iterations of this step.**
4. Subtract from $\vec{q}'(t_1)$ with $< \vec{q}'(t_1), \vec{q}_i(t_1) > \vec{q}_i(t_1)$ from $i = 0\cdots k$ with inner-product calculated as in (28).
5. k++. Repeat[2]-[4], loop till k = m (a reduced size that meets accuracy criteria).

The reduced system obtained from the above algorithm is

$$\begin{aligned}\hat{W}(t_1,s_2) &= [\frac{\hat{D}}{dt_1}[*]+s_2\hat{C}(t_1)+\hat{G}(t_1)]^{-1}[\tilde{b}(t_1)]U(s_2) \\ \tilde{b}(t_1) &= V_m^T(t_1)\vec{b} \\ Z(t_1,s_2) &= d^T V_m(t_1)\hat{W}(t_1,s_2) = \tilde{d}^T(t_1)\hat{W}(t_1,s_2)\end{aligned} \quad (31)$$

where

$$\hat{C}(t_1) = V_m^T(t_1)C(t_1)V_m(t_1), \quad (32)$$

$$\hat{G}(t_1) = V_m^T(t_1)G(t_1)V_m(t_1), \quad (33)$$

and

$$\frac{\hat{D}}{dt_1}[\cdot] = V_m^T(t_1)\frac{d[C(t_1)V_m(t_1)\cdot]}{dt_1}. \quad (34)$$

Therefore, the transfer function of the reduced system becomes

$$\hat{H}(t_1,s_2) = d^T V_m(t_1)[\frac{\hat{D}}{dt_1}[*]+s_2\hat{C}(t_1)+\hat{G}(t_1)]^{-1}[V_m^T(t_1)\vec{b}]. \quad (35)$$

In terms of the operators and vectors

$$\begin{aligned}\hat{\mathfrak{I}}_{t_1}[*] &= (\frac{\hat{D}}{dt_1}[*]+\hat{G}(t_1)), \\ \hat{\mathfrak{L}}_{t_1}[\cdot] &= \hat{\mathfrak{I}}_{t_1}^{-1}[\hat{C}(t_1)\cdot], \\ \hat{r}(t_1) &= \hat{\mathfrak{I}}_{t_1}^{-1}[\tilde{b}],\end{aligned} \quad (36)$$

the transfer function of the reduced system is

$$\begin{aligned}\hat{H}(t_1,s_2) &= \tilde{d}^T(t_1)(\hat{\mathfrak{I}}_{t_1}[*]+s_2\hat{C}(t_1))^{-1}\tilde{b}(t_1) \\ &= \tilde{d}^T(t_1)(I+s_2\hat{\mathfrak{I}}_{t_1}^{-1}[\hat{C}(t_1)*])^{-1}\hat{\mathfrak{I}}_{t_1}^{-1}[\tilde{b}(t_1)] \\ &= \tilde{d}^T(t_1)(\hat{I}+s_2\hat{\mathfrak{L}}_{t_1}[*])^{-1}\hat{r}(t_1).\end{aligned} \quad (37)$$

(37) is in the same form as the transfer function of the original system as shown in (11).

We emphasize that the main utility of this operator-based approach is that the choice of solution method of the linear system at each step of the model-order reduction procedure can be made completely dynamic. In other words, the original system does not need to be discretized a-priori using a given frequency- or time-domain technique. Indeed, different discretization and solution methods, can be used to gain efficiency or accuracy advantage as MOR proceeds. An equally important advantage is that the operator view of LPTV reduction significantly enhances code re-use and modularity. For example, we use a single implementation of Arnoldi for reducing both LTI and LPTV systems, simply by passing it appropriate function handles for norms, inner-products and linear system solution.

5. EXAMPLES

Three examples are presented in this section. The first example is a hand-calculable example to illustrate and verify the operator-based MOR approach. The second example is a simple upconverter (of size 5) and the third example is a double-balanced mixer of size 52. For the last two examples, the proposed algorithm is used to reduce the size of the original system (*e.g.* $52 \rightarrow 14$) and the frequency responses of the generated macromodel are shown to match the frequency responses of the original system in different ranges of interest.

5.1 A hand-calculable analytical example

A hand-calculable analytical example is first provided to illustrate the operator-based Arnoldi process. Define the vector $r(t)$ and the linear operator $A(t)[\cdot]$ to be, for example:

$$r(t) = \begin{bmatrix} sin(t) \\ cos(t) \end{bmatrix} \quad A(t)[\cdot] = \begin{bmatrix} 0 & 2 \\ 1 & 0 \end{bmatrix}$$

To make this example simple and hand-calculable, $A(t)[\cdot]$ is selected to be an LTI operator (*i.e.*, a matrix) and each term in $r(t)$ is a *sin* or *cos* function. The interpretation of the model-order reduction procedure will be different compared to the case when $A(t)[\cdot]$ is in an operator form.

Using the norm and inner-product defined in (27) and (28), and following the operator-based Arnoldi process defined in (26), we obtain

$$V_m(t) = \begin{bmatrix} sin(t) & 2\sqrt{\frac{2}{5}}cos(t) \\ cos(t) & \sqrt{\frac{2}{5}}sin(t) \end{bmatrix} H_m = \begin{bmatrix} 0 & 2\sqrt{\frac{2}{5}} \\ \sqrt{\frac{5}{2}} & 0 \end{bmatrix}.$$

with structure similar to (16) and (18) . H_m and $V_m(t)$ are needed for constructing the reduced system $\hat{H}(s)$. In this case, m=2, so the system size is the same as the original system. We compare the original and "reduced" systems to see if they match each other.

$$\begin{aligned}H(s) &= d^T(I+sA(t))^{-1}r(t) \\ &= d^T(1-2s^2)^{-1}\begin{bmatrix} sin(t)-2scos(t) \\ cos(t)-ssin(t) \end{bmatrix}\end{aligned}$$

$$\hat{H}(s) = r_{00} d^T V_m(t)(I + sH_m)^{-1} e_1$$
$$= d^T (1-2s^2)^{-1} \begin{bmatrix} sin(t) - 2scos(t) \\ cos(t) - ssin(t) \end{bmatrix}$$

From the above, the analytical solution from operator-Arnoldi can be seen to be identical to the original system.

5.2 A Simple Upconverter Example

A simple upconverter (from [14]) is shown in Figure 1. It has four stages: a low-pass filter (LPF), an ideal mixer and two band-pass filters (BPF). In the LPF, $R1 = 160\Omega$, $C1 = 10nF$; in the first BPF, $R2 = 1.6k\Omega$, $C2 = 10nF$, $L2 = 25.33nH$; in the second BPF, $R3 = 500\Omega$, $C3 = 10nF$, $L3 = 25.33nF$. The LPF's pole is at 100kHz and the two BPFs have center frequency at 10MHz. The first BPF has a bandwidth (BW) of about 10kHz, while the second BPF has a BW of about 30kHz. The LO frequency of the ideal mixer is 10MHz. The size of the system is $n = 5$.

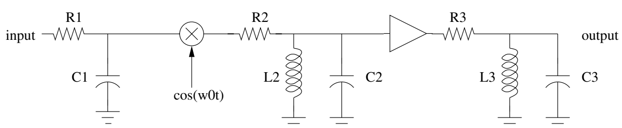

Figure 1: Upconverter

Using HB for discretizing the LPTV system (11), we consider the harmonic transfer functions $H_1(s)$ and $H_{-1}(s)$ [14] as outputs of the system. The analytical expression for $H_1(s)$ is known to be [14]

$$H_1(s) = \frac{0.5}{1+sC_1R_1} \frac{\frac{(s+jw_0)L_2}{1+(s+jw_0)^2L_2C_2}}{R_2 + \frac{(s+jw_0)L_2}{1+(s+jw_0)^2L_2C_2}} \frac{\frac{(s+jw_0)L_3}{1+(s+jw_0)^2L_3C_3}}{R_3 + \frac{(s+jw_0)L_3}{1+(s+jw_0)^2L_3C_3}}.$$

Figure 2 shows a comparison of the analytical expression vs frequency sweeps generated from the proposed operator-based LPTV macromodel. Similarly the comparison of the $H_{-1}(s)$ is given in Figure 3. The CPU runtime for the operator-based Arnoldi is about 94ms.

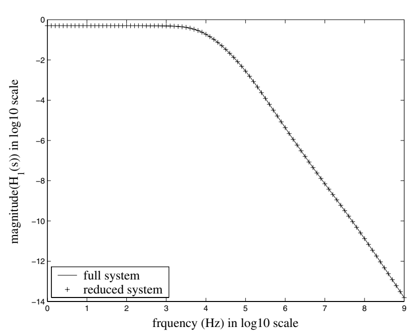

Figure 2: Upconverter $H_1(s)$: full versus reduced system

5.3 Double balanced mixer

Our second example is a double balanced mixer of size 52. The mixer topology, shown in Figure 4, is adapted from [10]. The mixer is followed by an operational amplifier (op-amp) block taken from

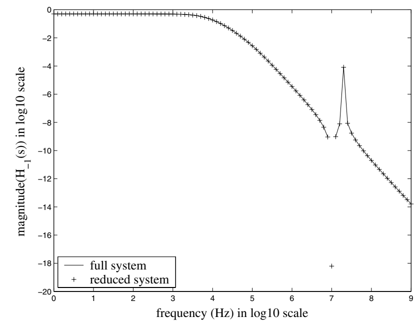

Figure 3: Upconverter $H_{-1}(s)$: full versus reduced system

[6] and depicted in Figure 5. The mixer, used for down-conversion, features a local oscillator (LO) frequency of 100MHz, followed by a low-pass filter with a pole at 1MHz.

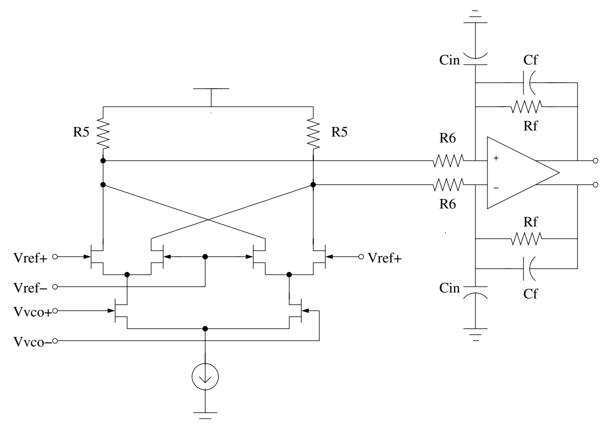

Figure 4: DBmixer Topology

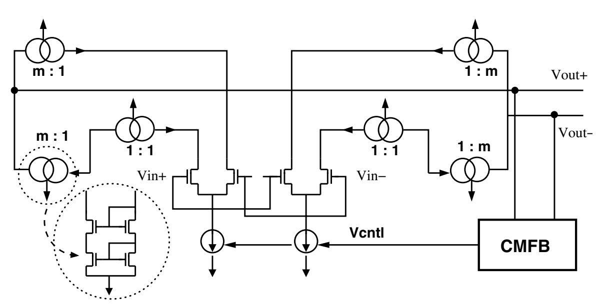

Figure 5: Op-amp Topology

The LPTV system to be reduced is obtained by linearization around the periodic steady state of the circuit at the LO frequency, computed using harmonic balance with a total of harmonics $N = 25$. Using operator-Arnoldi, a size 14 macromodel was generated in runtime 859 seconds. The $H_{-1}(s)$ component of the output is of greatest interest as it captures the downconversion component. Figure 6 shows a plot of $H_{-1}(s)$ vs frequency (the input RF frequency is greater than the LO frequency). The horizontal coordinate is $f_{RF} - f_{LO}$ (using a log scale), with f_{RF} ranging from the

LO frequency up to 200MHz. The full system is depicted as a solid line, while the reduced system of size 14 is denoted with "+" markers. Figure 7 shows frequency plots of $H_{-1}(s)$ when the range of the input RF frequency is smaller than the LO frequency. The horizontal coordinate in this case is $f_{LO} - f_{RF}$, with f_{RF} ranging from 1Hz to the LO frequency. From the two plots, we observe that the frequency response of this harmonic of the macromodel matches the original system's well; the relative error is 5.32×10^{-8}.

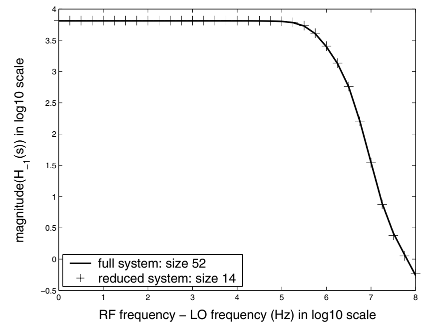

Figure 6: DBmixer $H_{-1}(s)$: reduced versus full system, RF frequency is from LO frequency (100MHz) to 200MHz

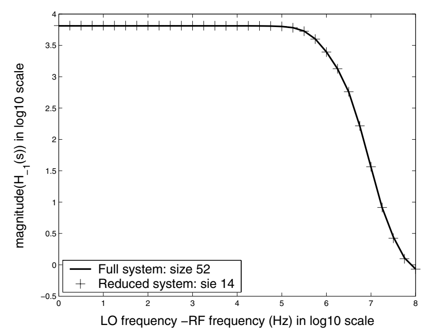

Figure 7: DBmixer $H_{-1}(s)$: reduced versus full system, RF frequency is from LO frequency (100MHz) to 1Hz

Figure 8 shows frequency plots of $H_{-1}(s)$ over the range f_{RF} from 1Hz to 10GHz. Observe that the responses are fully matched over the entire frequency range. If smaller bandwidths are of interest, macromodels of smaller size often suffice.

6. CONCLUSION

In this paper, we have presented an operator-based MOR algorithm useful for reducing LPTV systems. The key advantage of the method is that it enables superior code modularity and implementation flexibility, compared to prior LPTV MOR approaches such as [14]. Examples presented show that the proposed algorithm can capture frequency-response characteristics with good accuracy while reducing system size significantly.

7. REFERENCES

[1] W. Chauvet, B. Lacaze, D. Roviras, and A. Duverdier. Characterization of a set of invertible LPTV filters using circulant matrices. In Proc. International Conference Acoust. Speach Signal Process., pages VI–45–VI–48, 2003.

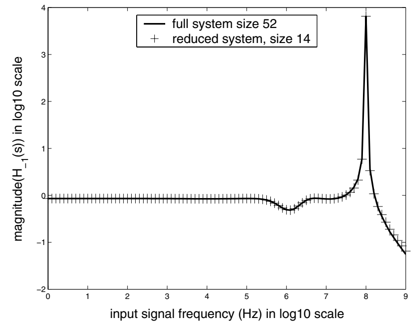

Figure 8: DBmixer H_{-1}(s): reduced versus full system, RF frequency is from 1Hz to 10GHz

[2] L. Daniel and J. Phillips. Model Order Reduction for Strictly Passive and Causal Distributed Systems. In Proc. IEEE DAC, pages 46–51, June 2002.

[3] D.C. McLernon. Relationship between an LPTV system and the equivalent LTI MIMO structure. IEE Proc.-Vis. Image Signal Process., 150(3):133–137, June 2003.

[4] A. Demir. Phase noise in oscillators: Daes and colored noise sources. In Proc. ICCAD, pages 170–177, 1998.

[5] A. Demir, A. Mehrotra, and J. Roychowdhury. Phase Noise in Oscillators: A Unifying Theory and Numerical Methods for Characterization. In Proc. IEEE DAC, pages 26–31, June 1998.

[6] N. Dong and J. Roychowdhury. Automated Extraction of Broadly Applicable Nonlinear Analog Macromodels from SPICE-level Descriptions. In Proc. IEEE CICC, 2004.

[7] E. Gad and M. Nakhla. Model Order Reduction of Nonuniform Transmission Lines Using Integrated Congruence Transform. In Proc. IEEE DAC, pages 238–243, June 2003.

[8] P. Feldmann and R.W. Freund. Efficient linear circuit analysis by Padé approximation via the Lanczos process. IEEE Trans. CAD, 14(5):639–649, May 1995.

[9] R. Grimshaw. Nonlinear Ordinary Differential Equations. Blackwell Scientific, 1990.

[10] J. Crols and M. S. J. Steyaert. A 1.5 GHz Highly Linear CMOS Downconversion Mixer. IEEE J. Solid-State Ckts., 30(7):736–742, July 1995.

[11] J. R. Phillips. Projection-Based Approaches for Model Reduction of Weakly Nonlinear, Time-Varying Systems. IEEE Trans. CAD, 22(2):171–187, February 2003.

[12] A. Odabasioglu, M. Celik, and L.T. Pileggi. PRIMA: passive reduced-order interconnect macromodelling algorithm. In Proc. ICCAD, pages 58–65, November 1997.

[13] Q. Yu, J. M. L. Wang, and E. S. Kuh. Passive Multipoint Moment Matching Model Order Reduction Algorithm on Multiport Distributed Interconnect Networks. IEEE Trans. Ckts. Syst. – I: Fund. Th. Appl., 46(1):140–160, January 1999.

[14] J. Roychowdhury. Reduced-order modelling of time-varying systems. IEEE Trans. Ckts. Syst. – II: Sig. Proc., 46(10):1273–1288, November 1999.

[15] P. Vanassche, G. Gielen, and W. Sansen. Time-Varying, Frequency-Domain Modeling and Analysis of Phase-Locked Loops with Sampling Phase-Frequency Detectors. In Design, Automation and Test in Europe Conference and Exhibition, 2003.

[16] Z. Wang, R. Murgai, and J. Roychowdhury. Automated, Accurate Macromodelling of Digital Aggressors for Power and Ground Supply Noise Prediction. In Design, Automation and Test in Europe Conference and Exhibition, volume 2, pages 20824–20829, February 2004.