MOESP Algorithm for Converting One-dimensional Maxwell Equation into a Linear System

E. F. Yetkin[1], H. Dağ[2], and W. H. A. Schilders[3]

[1] Istanbul Technical University, Informatics Institute, Istanbul, Turkey
fatih@be.itu.edu.tr
[2] Isk University, Information Technologies Department, Istanbul, Turkey
dag@isikun.edu.tr
[3] Technical University of Eindhoven, CASA, Eindhoven, The Netherlands
w.h.a.schilders@tue.nl

Summary. We present a method for converting 1-D Maxwell equation into a linear system using the Multivariable Output Error State Space (MOESP) method, a subspace system identification method. To show the efficiency of the method, we first apply it to a set of ordinary differantial equations. Input and output from the equation set are computed by numerical methods and the obtained data is used for building the required matrices. An appropriate Single Input Single Output (SISO) linear system is estimated by MOESP algorithm for the equation at hand. The goal of the research is to build a low order linear state space system model for the Maxwell equation. On the other hand the order estimation for the system can be used in other way. For example, with this estimation one can determine an appropriate order for the physical system, for which one of the well-known model order reduction techniques can be used to obtain a reduced order model.

1 Introduction

In general, system identification methods are mainly developed in the area of automatic control to determine the best model (in the sense of input-output relationship) from a given observed input-output data set. In this study, a 1-D Maxwell equation is converted into a set of state-space equations using MOESP algorithm, which is a member of subspace system identifacation family of algorithms. The idea can be useful when simulation of the VLSI interconnections are considered. The computation of the effects of VLSI interconnections is mainly based on the solution of the Maxwell equations on chip geometries. The RLC parasitic circuits are realized with the solution of Maxwell equations. Finally, the model order reduction algorithms are employed to reduce the dimension of the linear subsystem of these RLC circuits [ANT05]. In this study, 1-D Maxwell equation is directly converted into a small order SISO system without using any model reduction algorithm. Therefore, it can be also useful for finding an appropriate reduction order of the model order reduction process. Before dealing with the Maxwell equations however, let us use an ordinary differential equation set to show the usage and the details of the method.
The remaining of the paper is organized as follows. In section 2, the methodology and the MOESP algorithm are briefly explained, whereas section 3 contains some numerical results and discussions. We present, in section 4 some concluding remarks and the future work.

2 Definition of the Problem

2.1 Introduction

To explain the basics of and the implementations details of the MOESP algorithm, a general n^{th} order ordinary differential equation (ODE) is considered. We also present numerical results for this case in the paper. Then the method is applied to the partial differential equations (PDE), more specifically to the Maxwell equations.

2.2 n^{th} Ordinary Differential Equation as a Discrete Linear System

A general linear differential equation of order *n* with zero inital values on an interval I is defined as,

$$y^{(n)}(t) + a_{n-1}y^{(n-1)}(t) + \cdots + a_0 y(t) = u(t), \quad t \in I \tag{1}$$

This system can be reduced to an associated first order ordinary differential equation system.

$$\frac{d}{dt}X = AX + Bu \tag{2}$$

where;

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{n-1} \end{bmatrix}, \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{3}$$

Matrix **A** is called *companion matrix*. Components of the **X** vector are called state variables. Furthermore, it is always possible to define the desired output as a linear combination of these state variables [CARL97].

$$Y = CX + Du \tag{4}$$

It is also possible to convert this continuos system into a discrete system with the help of any numerical integration algorithm. For example, if we choose Euler method for integration we obtain below difference equations for x_n at time t_{k+1},

$$\frac{x_n^{k+1} - x_n^k}{h} = a_{n1}x_1^k + a_{n2}x_2^k + \cdots + a_{nn}x_n^k + b_n u^k \tag{5}$$

or in matrix form,

$$\begin{aligned} X_{k+1} &= (I + hA)X_k + Bu_k \\ Y_{k+1} &= CX_{k+1} + Du_{k+1} \end{aligned} \tag{6}$$

Using (6), one can write the input-output formulas for each data point;

$$Y_{k+j} = C(I + hA)^j X_k + \sum_{i=1}^{j} C(I + hA)^{i-1} Bu_{k+j-i} + Du_{k+j} \tag{7}$$

Using (7) we can derive matrix input-output equations which play a fundamental role in subspace identification,

$$\begin{bmatrix} y_k \\ y_{k+1} \\ \vdots \\ y_{k+j} \end{bmatrix} = \begin{bmatrix} C \\ C\widehat{A} \\ \vdots \\ C\widehat{A}^j \end{bmatrix} X_k + \begin{bmatrix} D & & & \\ CB & D & & \\ \vdots & \ddots & \ddots & \\ C\widehat{A}^{j-1}B & \dots & CB & D \end{bmatrix} \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+j} \end{bmatrix} \tag{8}$$

where $\widehat{A} = (I + hA)$. One can define two Hankel matrices in terms of u_k and y_k to generalize the structure. These Hankel matrices are called as $U_{0|k-1}$ and $Y_{0|k-1}$ respectively.

$$\begin{bmatrix} u(0) & u(1) \dots & u(N-1) \\ u(1) & u(2) \dots & u(N) \\ \vdots & \vdots & \vdots \\ u(k-1) & u(k) \dots & u(k+N-2) \end{bmatrix} \in \mathcal{R}^{km\text{x}N} \begin{bmatrix} y(0) & y(1) \dots & y(N-1) \\ y(1) & y(2) \dots & y(N) \\ \vdots & \vdots & \vdots \\ y(k-1) & y(k) \dots & y(k+N-2) \end{bmatrix} \in \mathcal{R}^{kp\text{x}N} \tag{9}$$

where k is strictly greater than the order of the system n, p is the number of the outputs of the system, m is the number of the inputs and finally N is a sufficiently large number for fixing the Hankel matrix. 0 and $k-1$ values in the definitions of Hankel matrices are used for determination of the upper-left and lower-left elements respectively. Using this Hankel matrix definitions one can write below equations for the n^{th} order ordinary differential system.

$$\begin{aligned} Y_{0|k-1} &= \mathcal{O}_k X_0 + \Phi_k U_{0|k-1} \\ Y_{k|2k-1} &= \mathcal{O}_k X_k + \Phi_k U_{k|2k-1} \end{aligned} \tag{10}$$

where

$$\mathcal{O}_k = \begin{bmatrix} C \\ C\widehat{A} \\ \vdots \\ C\widehat{A}^j \end{bmatrix}, \quad \Phi_k = \begin{bmatrix} D & & & \\ CB & D & & \\ \vdots & \ddots & \ddots & \\ C\widehat{A}^{j-1}B & \dots & CB & D \end{bmatrix} \tag{11}$$

Here, X_0 and X_k are the initial states respectively. $U_{0|k-1}$ and $Y_{0|k-1}$ are called past inputs and outputs and $U_{k|2k-1}$ and $Y_{k|2k-1}$ are called future inputs and outputs [KAT05].

Data matrices $U_{0|k-1}$, $Y_{0|k-1}$ can be written in more compact form as:

$$\begin{bmatrix} U_{0|k-1} \\ Y_{0|k-1} \end{bmatrix} = \begin{bmatrix} I_{km} & 0_{kmxn} \\ \Phi_k & \mathcal{O}_k \end{bmatrix} \begin{bmatrix} U_{0|k-1} \\ X_0 \end{bmatrix} \tag{12}$$

Finally, it can be said that it is always possible to rewrite (1) as a matrix equation as given in (12).

2.3 MOESP Algorithm

LQ decomposition, which is the dual of the QR decomposition, is used to make the upper-right block of the data matrix zero. LQ decomposition of a matrix can be given as,

$$\begin{bmatrix} U_{0|k-1} \\ Y_{0|k-1} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} \tag{13}$$

where $L_{11} \in \mathcal{R}^{km\text{x}km}$, $L_{22} \in \mathcal{R}^{kp\text{x}kp}$, $Q_1 \in \mathcal{R}^{N\text{x}km}$ $Q_2 \in \mathcal{R}^{N\text{x}kp}$.
The actual computation of LQ decomposition is performed by taking transpose of the QR decomposition of the matrix.
Using orthogonality conditions on the input output spaces, below equation can be obtained for L_{22},

$$\mathcal{O}_k X_0 Q_2 = L_{22} \tag{14}$$

where $\mathcal{O}_k$ is extended observability matrix, X_0 is the initial states. If we take the SVD of the L_{22} matrix we get,

$$L_{22} = [U_1 U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 \Sigma_1 V_1^T \tag{15}$$

In MOESP algorithm, the system dimension is determined by the singular values of the L_{22} matrix and with this decomposition we have,

$$\mathcal{O}_k X_0 Q_2 = U_1 \Sigma_1 V_1^T. \tag{16}$$

From last identity, we can define the extended observability matrix as

$$\mathcal{O}_k = U_1 \Sigma_1^{1/2}. \tag{17}$$

With (17) we have the C matrix of the estimated system as $C = \mathcal{O}_k(1:p, 1:n)$ and the A matrix as a a solution of below least square equation $\mathcal{O}_k(1:p(k-1), 1:n)A = \mathcal{O}_k(p+1:kp, 1:n)$.
Computation of the B and the D matrices are more complex. We refer the reader to work in [CIG98], [VD92-1], and [VD92-2] for further information. Algorithm of the method is given in Fig. (1).

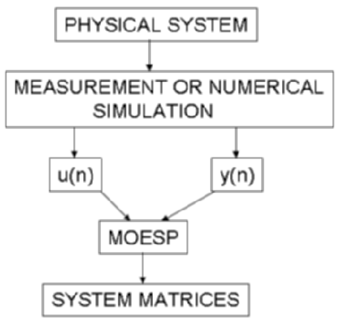


Fig. 1: Main MOESP Algorithm

3 Numerical Examples

3.1 Example ODE System

A second order ODE equation is selected for estimation. The equation is

$$\frac{d^2\varphi(t)}{dt^2} + \frac{d\varphi(t)}{dt} - 10 = 0 \tag{18}$$

Here, the input u is constant and equals to 10 and initial values of equation taken as zero. The output y is computed by a Runge-Kutta algorithm. Data matrices are created after the input and output data collected. Then the SMI Toolbox employed to produce the estimation [SMI]. The singular value distribution of L_{22} matrix is shown in Fig. (2).

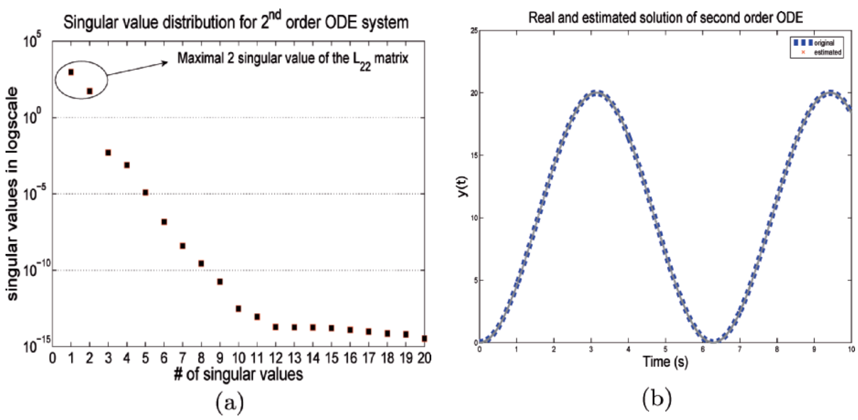


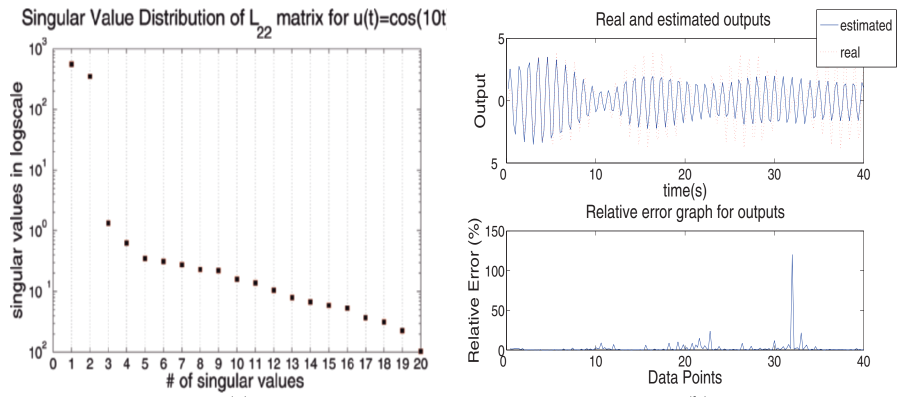
Fig. 2: (a) Singular value distribution of data matrix for u(t)=10 and estimated system order $n = 2$, (b) Original and estimated outputs for a estimated system order $n = 2$

3.2 Maxwell Equation

Consider a one-dimensional space where there are only variations in the x dimension. Assume that the electric field has only a z component. With Faraday and Ampere's laws we can write 1-D Maxwell equations as

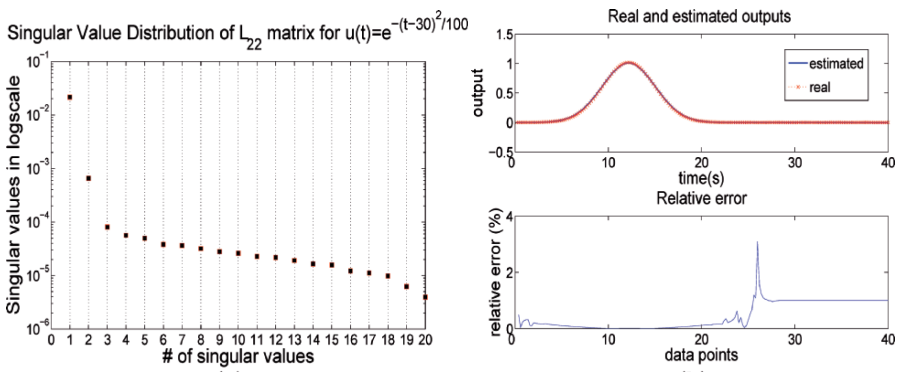
$$\begin{aligned} \mu\frac{\partial H_y}{\partial t} &= \frac{\partial E_z}{\partial x}, \\ \epsilon\frac{\partial E_z}{\partial t} &= \frac{\partial H_y}{\partial x}. \end{aligned} \tag{19}$$

The source function is applied to the 0^{th} node of the computational domain and data is collected as the electrical field of 50^{th} node. After discretization, FDTD (Finite Difference Time Domain) algorithm is employed to obtain the input data u_k and output data y_k. The singular value distribution of the L_{22} matrix and the original and estimated outputs are given in Figs. 3 and 4. Here, two source functions are considered. First one is a sinusoidal and second one is an exponential function . For exponential source function MOESP algorithm works more accurately. Estimated order n, is selected as 2 in both cases.



(a) (b)

Fig. 3: (a) Singular value distribution of L_{22} matrix for $u(t) = cos(10t)$ (b) Original and estimated outputs and relative error for $u(t) = cos(10t)$ where the estimated system order $n = 2$



(a) (b)

Fig. 4: (a) Singular value distribution of L_{22} for $u(t) = exp^{-(t-30)^2/100}$, (b) Original and estimated outputs and relative error for estimated system order $n = 2$

3.3 Comparison of Estimations

For ODE systems, singular value distributions of the L_{22} matrix is reasonable if one considers that a second order differential equation is estimated. Ratios of its maximum two singular values to the other singular values are sufficiently small and the singular values except first two largest ones can be neglected. This situation can be seen from the Fig. 2, the outputs are exactly matched.
This can be verified with one of the possible measures of accuracy which named as VAF (Variance According For) [SMI]. It is defined as,

$$VAF = 1 - \frac{variance(y - y_{est})}{variance(y)} * 100\% \tag{20}$$

where y is the original output and y_{est} is the estimated output. The VAF of two signals that are the same is 100%. If they differ, the VAF will be lower and if the two signals are completly different then VAF gets value of -1000.

For ODE example, VAF is equal to 100%. It means that the estimation works very successfully for this set of equation.
In the Figs. 3 and 4, singular values of the L_{22} matrix are relatively close to each other and we cannot select an exact estimation order like for the ODE system in Fig. 2. Therefore, it can be said that, for Maxwell equations the accuracy of the estimations is more sensitive to the selection of the estimation order. VAF values for these equations are 86% and 96.5% respectively. It can be also said that for exponential source functions, MOESP algorithm produces more accurate result. This fact can be observed from Figs 3 and 4. Its possible reason is the periodicity of the input and output vectors. Linear dependency of the columns of data matrices are determined by the input output vectors. Here we can say that, for non-periodic input sources Maxwell equations also can be modeled as a linear system with high accuracy. But in the case of the periodical input sources some other methodologies have to be used to improve the accuracy of the method.

4 Conclusion and Future Work

We examined the algorithm MOESP to convert a 1-D Maxwell equation into a SISO linear discrete state-space system.
Method is applied to an ordinary differential equation first and it is observed that the method produces a linear system quadruple (A,B,C,D) with high accuracy. However, when applying the same method to 1D-Maxwell equation accuracy of the method varies depending on the input source. With non-periodical input signals results are more accurate than those of the periodical input signal case. There can be a relationship between the periodicity of the input-output data and the behaviour of the algorithm. The future work will be focused on finding this relationship, i.e., the relationship between the order of the estimated system and the properties of data matrices.
We so far have studied the SISO modeling of the equations. In SISO models one has to define only one output point. On the other hand, in realistic systems more than one output point are required for modelling. Therefore, the method has to be extended to MIMO (Multiple Input Multiple Output) cases for extending the implementation area.

References

[ANT05] Antoluas, A. C.: Approximation of Large-Scale Dynamical Systems, SIAM (2005).
[KAT05] Katayama, T.: Subspace Methods for System Identification, Springer-Verlag (2005).
[VD92-1] Verhaegen, M., Dewilde P.: Subspace Model Identification Part 1. The Output-Error State-Space Model Identification Class of Algorithms, Int. J. Control, vol:56, No:5, pp:1187-1210, Taylor&Francis (1992).
[VD92-2] Verhaegen, M., Dewilde P.: Subspace Model Identification Part 2. Analysis of the Elementary Output-Error State-Space Model Identification Algortihm, Int. J. Control, vol:56, No:5, pp:1211-1241, Taylor&Francis (1992).
[CIG98] Ciggaar, E.: The MOESP Algorithm for Time Domain Modelling, Nat. Lab. Tchnical Note, TN 123/98 (1998).

[CARL97] Coddington E. A., Carlson R., Linear Ordinary Differential Equations, SIAM, Philedelphia, 1997.

[SMI] Haverkamp B., Verhaegen M.: State Space Model Identification Software for Multivariable Dynamical Systems, Delft University of Technology, Technical Notes, TUD/ET/SCE96.015 (1996).