MFTI: Matrix-Format Tangential Interpolation for Modeling Multi-Port Systems

Yuanzhe Wang, Chi-Un Lei, Grantham K. H. Pang and Ngai Wong
Department of Electrical and Electronic Engineering
The University of Hong Kong, Pokfulam Road, Hong Kong
{yzwang, culei, gpang, nwong}@eee.hku.hk

ABSTRACT

Numerous algorithms to macromodel a linear time-invariant (LTI) system from its frequency-domain sampling data have been proposed in recent years [1, 2, 3, 4, 5, 6, 7, 8], among which Loewner matrix-based tangential interpolation proves to be especially suitable for modeling massive-port systems [6, 7, 8]. However, the existing Loewner matrix-based method follows vector-format tangential interpolation (VFTI), which fails to explore all the information contained in the frequency samples. In this paper, a novel matrix-format tangential interpolation (MFTI) is proposed, which requires much fewer samples to recover the system and yields better accuracy when handling under-sampled, noisy and/or ill-conditioned data. A recursive version of MFTI is proposed to further reduce the computational complexity. Numerical examples then confirm the superiority of MFTI over VFTI.

Categories and Subject Descriptors

B.7.2 [**Integrated circuits**]: Design aids-simulation

General Terms

Algorithms

Keywords

matrix-format tangential interpolation (MFTI), Loewner matrix, state space, sampling

1. INTRODUCTION

High-frequency effects, such as signal delay and crosstalk, have become dominant factors limiting system performance in IC design. Accurate simulation is required to capture the high-frequency behavior of systems, so as to ensure consistent design of high-speed electronic systems. To model complicated geometry structures, such as packages, boards and RF objects, data-driven macromodeling is usually applied. Linear macromodeling can be classified as a system identification problem [9] wherein circuits or systems are treated as black boxes, and their responses are measured through experiments or calculated by EM simulators. Given the sampled frequency and/or time responses, such as admittance or scattering matrices, a model is built which fits the samples accurately with a satisfactory computational efficiency. Several methods have been developed, among which the recently proposed Loewner-matrix-based Vector-Format Tangential Interpolation (VFTI) proves to be accurate and efficient for modeling systems with massive ports [6, 7, 8]. VFTI is robust, non-iterative, and does not require pole initialization as in vector fitting. The order of the underlying system is also automatically recognized. However, VFTI interpolates only two vectors of a scattering matrix at a time. Hence, it does not explore all the information contained in the scattering matrices and loses approximation accuracy for noisy responses. It also suffers numerical problems when modeling ill-conditioned samples poorly distributed in the frequency band of interest.

In this paper, the generalization of VFTI to Matrix-Format Tangential Interpolation (MFTI) is proposed, which results in significant improvements over VFTI in practical macromodeling: (i)by extending the tangential data to matrix format, MFTI interpolates the full sampling matrix instead of two vectors. Hence, to achieve the same accuracy, MFTI requires only $1/p$ samples compared to VFTI (p being the number of ports) to recover the underlying system. This is a significant improvement considering that frequency-domain sampling usually involves expensive measurement or computation. Also, the accuracy is improved when modeling noisy responses; (ii)by giving different weightings to different samples, MFTI is suitable for interpolating ill-conditioned data (namely, when the sampling frequencies are poorly distributed in the band of interest). It also provides an option to trade off between computational complexity and fitting accuracy; (iii)a recursive version of MFTI is developed to automatically select the appropriate set of sampled data for approximation; (iv)a minimal sampling theorem is introduced as theorem 3.5 that guides the number of sampling points required for an efficient computation.

This paper is organized as follows. In Section 2, the basics of VFTI are introduced. In Section 3, the formulation of MFTI is presented. In Section 4, two MFTI algorithms are summarized. Numerical examples are given in Section 5 and Section 6 draws the conclusion.

DAC'10, June 13-18, 2010, Anaheim, California, USA

2. BACKGROUND

An LTI system can be expressed as a state-space model:

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \tag{1}$$

where $y(t) \in \mathbb{R}^p$, $u(t) \in \mathbb{R}^m$, $x(t) \in \mathbb{R}^n$, $E, A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, $D \in \mathbb{R}^{p\times m}$. If E is singular, (1) is called a descriptor system (DS) whose transfer function is $H(s) = C(sE - A)^{-1}B + D$.

2.1 Overview of VFTI

The goal of an interpolation problem is to model the underlying system or circuit from its measured/calculated input-output data. Assume that scattering matrices have been sampled at different frequencies f_i, $i = 1, 2, \ldots, k$,

$$S(f_i) = \begin{bmatrix} S_{11}(f_i) & \ldots & S_{1m}(f_i) \\ \vdots & \ddots & \vdots \\ S_{p1}(f_i) & \ldots & S_{pm}(f_i) \end{bmatrix}. \tag{2}$$

Our goal is to find a DS as (1) whose transfer function satisfies

$$H(j2\pi f_i) = C(j2\pi f_i E - A)^{-1}B + D \approx S(f_i). \tag{3}$$

In VFTI, tangential interpolation data are used instead of the whole $S(f_i)$. Specifically, the right and left tangential interpolation data are defined as [6, 7, 8]

$$\begin{aligned} &\{\lambda_i, \mathbf{r}_i, \mathbf{w}_i \mid \lambda_i \in \mathbb{C}, \mathbf{r}_i \in \mathbb{C}^{m\times 1}, \mathbf{w}_i = \mathbf{S}(f_i)\mathbf{r}_i \in \mathbb{C}^{p\times 1}, i = 1..\rho\}, \\ &\{\mu_i, \mathbf{l}_i, \mathbf{v}_i \mid \mu_i \in \mathbb{C}, \mathbf{l}_i \in \mathbb{C}^{1\times p}, \mathbf{v}_i = \mathbf{l}_i\overline{\mathbf{S}(f_i)} \in \mathbb{C}^{1\times m}, i = 1..\nu\}, \end{aligned} \tag{4}$$

whereby $\rho + \nu = k$, $\lambda_i, \mu_i \in \{\pm 2\pi f_i\}$. In (4), the vectors $\mathbf{r}_i$ and $\mathbf{l}_i$ are arbitrarily chosen interpolation directions. The vectors $\mathbf{w}_i$ and $\mathbf{v}_i$ are called interpolation data. The right and left interpolation data are used to generate a state space whose transfer function satisfies

$$\mathbf{H}(\lambda_i)\mathbf{r}_i = \mathbf{w}_i \quad \text{and} \quad \mathbf{l}_i\mathbf{H}(\mu_i) = \mathbf{v}_i. \tag{5}$$

Apparently, VFTI has the intrinsic disadvantage that it can only interpolate two vectors, either row or column, of a sample matrix. Thus (3) is not automatically satisfied by guaranteeing tangential constraints (5).

3. MATRIX-FORMAT TANGENTIAL INTERPOLATION

3.1 Generation of interpolation data

Suppose we have sampled k scattering matrices at k different frequencies (k is assumed even for simplicity). The matrix-format right and left interpolation data are in their general forms

$$\begin{aligned} &\{\lambda_i, R_i, W_i \mid \lambda_i = j2\pi f_i, W_i = S(f_i)R_i \text{ for } i = 1, 3 \ldots k-1; \\ &\quad \lambda_i = -\lambda_{i-1}, R_i = R_{i-1}, W_i = \overline{W_{i-1}} \text{ for } i = 2, 4 \ldots k\}, \end{aligned} \tag{6}$$

$$\begin{aligned} &\{\mu_i, L_i, V_i \mid \mu_i = j2\pi f_{i+1}, V_i = L_i S(f_{i+1}), \text{ for } i = 1, 3 \ldots k-1; \\ &\quad \mu_i = -\mu_{i-1}, L_i = L_{i-1}, V_i = \overline{V_{i-1}} \text{ for } i = 2, 4 \ldots k\}. \end{aligned} \tag{7}$$

They can also be expressed in a more compact format (wherein $t_1 = t_2, t_3 = t_4, \ldots, t_{k-1} = t_k$, and the superscript of a matrix denotes its dimension) as follows

$$\begin{aligned} \Lambda &= \mathrm{diag}[\underbrace{\lambda_1..\lambda_1}_{t_1}, \underbrace{-\lambda_1..-\lambda_1}_{t_2}, \cdots, \underbrace{\lambda_{k-1}..\lambda_{k-1}}_{t_{k-1}}, \underbrace{-\lambda_{k-1}..-\lambda_{k-1}}_{t_k}], \\ R &= [R_1^{m\times t_1}, R_1^{m\times t_2}, \ldots, R_{k-1}^{m\times t_{k-1}}, R_{k-1}^{m\times t_k}], \\ W &= [W_1^{p\times t_1}, \overline{W_1}^{p\times t_2}, \ldots, W_{k-1}^{p\times t_{k-1}}, \overline{W_{k-1}}^{p\times t_k}], \end{aligned} \tag{8}$$

$$M = \mathrm{diag}[\underbrace{\mu_1..\mu_1}_{t_1}, \underbrace{-\mu_1..-\mu_1}_{t_2}, \cdots, \underbrace{\mu_{k-1}..\mu_{k-1}}_{t_{k-1}}, \underbrace{-\mu_{k-1}..-\mu_{k-1}}_{t_k}],$$

$$L = \begin{bmatrix} L_1^{t_1\times p} \\ L_1^{t_2\times p} \\ \vdots \\ L_{k-1}^{t_{k-1}\times p} \\ L_{k-1}^{t_k\times p} \end{bmatrix}, \ V = \begin{bmatrix} V_1^{t_1\times m} \\ \overline{V}_1^{t_2\times m} \\ \vdots \\ V_{k-1}^{t_{k-1}\times m} \\ \overline{V}_{k-1}^{t_k\times m} \end{bmatrix}. \tag{9}$$

Here we use both the scattering matrices and their conjugates as tangential interpolation data to guarantee $H(-j2\pi f_i) = \overline{H(j2\pi f_i)}$. For systems with identical number of inputs and outputs ($m = p$), if $t_i = m$ and $\mathrm{rank}(L_i) = \mathrm{rank}(R_i) = t_i$(for $i = 1, 2 \ldots k$), then all entries in the scattering matrices are exploited for interpolation. We can also set t_i to be different numbers to trade off between speed and accuracy and/or to give different weightings to ill-conditioned samples. The goal of interpolation is to find a state-space realization whose transfer function [cf. (3)] satisfies the left and right constraints

$$H(\lambda_i)R_i = W_i \quad \text{and} \quad L_i H(\mu_i) = V_i. \tag{10}$$

3.2 Block-format (shifted) Loewner matrix

Similar to [6, 7, 8], block-format Loewner matrix $\mathbb{L}$ and shifted Loewner matrix $\sigma\mathbb{L}$ are defined as

$$\mathbb{L} = \begin{bmatrix} \frac{V_1R_1 - L_1W_1}{\mu_1 - \lambda_1} & \ldots & \frac{V_1R_k - L_1W_k}{\mu_1 - \lambda_k} \\ \vdots & \ddots & \vdots \\ \frac{V_kR_1 - L_kW_1}{\mu_k - \lambda_1} & \ldots & \frac{V_kR_k - L_kW_k}{\mu_k - \lambda_k} \end{bmatrix}, \tag{11}$$

$$\sigma\mathbb{L} = \begin{bmatrix} \frac{\mu_1V_1R_1 - \lambda_1L_1W_1}{\mu_1 - \lambda_1} & \ldots & \frac{\mu_1V_1R_k - \lambda_kL_1W_k}{\mu_1 - \lambda_k} \\ \vdots & \ddots & \vdots \\ \frac{\mu_kV_kR_1 - \lambda_1L_kW_1}{\mu_k - \lambda_1} & \ldots & \frac{\mu_kV_kR_k - \lambda_kL_kW_k}{\mu_k - \lambda_k} \end{bmatrix}. \tag{12}$$

Note that every diagonal block entry in (11) and (12) is a $t_i \times t_i$ square matrix, thus $\mathbb{L}$ and $\sigma\mathbb{L}$ are both $(t_1 + t_2 + \ldots + t_k) \times (t_1 + t_2 + \ldots + t_k)$ square matrices. Similar to VFTI, MFTI data fulfill the Sylvester equations,

$$\mathbb{L}\Lambda - M\mathbb{L} = LW - VR; \ \sigma\mathbb{L}\Lambda - M\sigma\mathbb{L} = LW\Lambda - MVR. \tag{13}$$

3.3 State-space realization

The state-space matrices in (1) can be calculated based on R, L, V, W, $\mathbb{L}$ and $\sigma\mathbb{L}$.

LEMMA 3.1. *If* $\forall x \in \{\lambda_1 \ldots \lambda_k, \mu_1 \ldots \mu_k\}$, $\det(x\mathbb{L} - \sigma\mathbb{L}) \neq 0$, *then* $E = -\mathbb{L}$, $A = -\sigma\mathbb{L}$, $B = V$, $C = W$, $D = 0$ *constitute a minimal state-space model whose transfer function satisfies (10). Furthermore, if* $m = p = t_i$ *and* L_i, R_i *are of full rank, the transfer function also satisfies (3).*

Proof: The proof of the first conclusion is similar to Appendix A-A of [8]. Note that the vector-format tangential interpolation direction and data should be replaced by matrix-format ones and that e_i there should be replaced by $E_i = [0^{t_i\times(t_1+..+t_{i-1})}, I^{t_i\times t_i}, 0^{t_i\times(t_{i+1}+..+t_k)}]^T$.

Then consider the second conclusion. If $p = m = t_i$, (10) becomes $H(\lambda_i)R_i^{m\times m} = W_i = S(f_i)R_i^{m\times m}$ for $i = 1, 3 \ldots k-1$. Because the square matrix $R_i^{m\times m}$ is chosen to be of full rank, we get $H(j2\pi f_i) = H(\lambda_i) = S(f_i)$ for $i = 1, 3, \ldots, k-1$. Similarly, $H(j2\pi f_i) = S(f_i)$ for $i = 2, 4, \ldots, k$. Thus (3) is guaranteed. □

This lemma indicates that if the number of inputs is identical to the number of outputs (i.e., $m = p$), which is the case for a large group of (e.g., MNA) circuits, (3) is satisfied exactly and $H(j2\pi f_i) = S(f_i)$ for all i. Note that (3) cannot be guaranteed by VFTI. To guarantee the state-space matrices to be real, we have the following lemma.

Lemma 3.2. *If R, L, V, W, $\mathbb{L}$ and $\sigma\mathbb{L}$ are constructed to satisfy the conditions in Lemma 3.1, then by using * to denote conjugate transpose, $E = -T^*\mathbb{L}T$, $A = -T^*\sigma\mathbb{L}T$, $B = T^*V$, $C = WT$ are guaranteed real and constitute a minimal DS whose transfer function satisfies (10). Here, $T = blkdiag[T_1, T_3, \ldots, T_{k-1}], T_i = \frac{1}{\sqrt{2}}\begin{bmatrix} I^{t_i\times t_i} & -jI^{t_i\times t_i} \\ I^{t_i\times t_i} & jI^{t_i\times t_i}\end{bmatrix}$, $i = 1, 3, \ldots, k-1$.*

3.4 SVD and minimal sampling

Lemma 3.3. *Suppose that the interpolation data are generated by sampling an underlying system with state-space expression $E_0\dot{x} = A_0x + B_0u$ and $y = C_0x + D_0u$. Then for $x \in \{\lambda_i\}\bigcup\{\mu_i\}$, $\mathbf{rank}(x\mathbb{L} - \sigma\mathbb{L}) \le \mathbf{size}(A_0) + \mathbf{rank}(D_0)$.*

Proof: Note that

$$x\mathbb{L} - \sigma\mathbb{L} = \begin{bmatrix} L_1C_0(\mu_1E_0 - A_0)^{-1} \\ \vdots \\ L_kC_0(\mu_kE_0 - A_0)^{-1}\end{bmatrix}(A_0 - xE_0)\times$$
$$\left[(\lambda_1E_0 - A_0)^{-1}BR_1 \ldots (\lambda_kE_0 - A_0)^{-1}BR_k\right] - LD_0R\Box$$

The Lemma indicates that when the size of $\mathbb{L}$ or $\sigma\mathbb{L}$ is larger than $\mathbf{size}(A_0)$+$\mathbf{rank}(D_0)$, the assumption of Lemma 3.1 will not hold. Subsequently, to build appropriate state-space matrices, we need to perform singular value decomposition (SVD) to get rid of the irregular part.

Lemma 3.4. *Suppose that for some $x_0 \in \{\mu_i\}\bigcup\{\lambda_i\}$, $\mathbf{rank}(x_0\mathbb{L} - \sigma\mathbb{L}) = \mathbf{rank}[\mathbb{L}\ \sigma\mathbb{L}] = \mathbf{rank}\begin{bmatrix}\mathbb{L} \\ \sigma\mathbb{L}\end{bmatrix} \triangleq r$. Then we perform an economic SVD: $(x_0\mathbb{L}-\sigma\mathbb{L}) = Y^{K\times r}S^{r\times r}(X^{r\times K})^*$, then $E = -Y^*\mathbb{L}X$, $A = -Y^*\sigma\mathbb{L}X$, $B = Y^*V$, $C = WX$ is a realization of the sampling data.*

Here $K \triangleq t_1 + t_2 + \ldots + t_k$ is the order of $\mathbb{L}$ and $\sigma\mathbb{L}$. The proof is similar to that in [6] except that $E_i = [0^{t_i\times(t_1+..+t_{i-1})}, I^{t_i\times t_i}, 0^{t_i\times(t_{i+1}+..+t_k)}]^T$ should be used instead of e_i

The SVD approach relies on the assumption made in Lemma 3.4. In real implementation, this assumption is generally true provided that the sampling is sufficient and thus the size of $\mathbb{L}$ ($\sigma\mathbb{L}$) is large enough. The test results [see example 1] show that the ranks of $\mathbb{L}$ and $\sigma\mathbb{L}$ satisfy $\mathbf{rank}(\mathbb{L}) \approx \mathbf{order}(\Gamma)$ and $\mathbf{rank}(\sigma\mathbb{L}) \approx \mathbf{rank}(x\mathbb{L} - \sigma\mathbb{L}) \approx \mathbf{order}(\Gamma) + \mathbf{rank}(D_0)$. Here Γ represents the underlying system $(E_0, A_0, B_0, C_0, D_0)$ and $\mathbf{order}(\Gamma) = \mathbf{rank}(E_0)$ represents the number of poles of Γ. This is partially because that the norm of $\sigma\mathbb{L}$ is often much larger than that of $\mathbb{L}$. If x is chosen to be λ_1 or μ_1, $x\mathbb{L} - \sigma\mathbb{L}$ is close to $\sigma\mathbb{L}$, thus they have similar singular value pattern. To recover the system, we should perform SVD and use the singular values to determine the regular part to be kept. Thus, the order of $\mathbb{L}$ and $\sigma\mathbb{L}$ (i.e. K) cannot be smaller than $\mathbf{order}(\Gamma)$+$\mathbf{rank}(D_0)$. The following theorem follows.

Theorem 3.5. *The least number of noise-free samples k_{min} to recover the underlying system satisfies $\mathbf{order}(\Gamma)/\min(m,p) \le k_{min} \le (\mathbf{size}(A_0) + \mathbf{rank}(D_0))/\min(m,p)$. Empirically $k_{min} = (\mathbf{order}(\Gamma) + \mathbf{rank}(D_0))/\min(m,p)$.*

The above theorem implies that the minimum samples required in MFTI are smaller than those in VFTI. The latter needs at least $\mathbf{order}(\Gamma)$ samples.

4. SUMMARY OF ALGORITHMS

Algorithm 1 is proposed by summarizing the results in Section 3.

Algorithm 1 MFTI of noise-free data

1: Set $t_i \in [1, \min(m,p)]$ for $i = 1\ldots k$, construct orthonormal matrix-format interpolation direction L_i, R_i;
2: Construct MFTI data following (6) & (7);
3: Construct $\mathbb{L}$, $\sigma\mathbb{L}$ from (11) & (12) or solve $\mathbb{L}$, $\sigma\mathbb{L}$ from (13);
4: Calculate projected W,V,$\mathbb{L}$,$\sigma\mathbb{L}$ following Lemma 3.2;
5: Select an $x_0 \in \{\lambda_i\}\bigcup\{\mu_i\}$, perform SVD on $(x_0\mathbb{L} - \sigma\mathbb{L})$;
6: Obtain state-space representation (E, A, B, C) of the recovered system following Lemma 3.4.

Real-world data are often noisy, thus the singular values in the above algorithm may be disturbed. To achieve a better accuracy, more samples need to be taken as interpolation data in order to minimize random error. But as the complexity of the algorithm increases quickly with the number of samples and orders of $\mathbb{L}$ ($\sigma\mathbb{L}$) and we cannot estimate how many samplings are required, algorithm 2 (with Matlab-style matrix notations) is proposed to reduce the complexity. When interpolating ill-conditioned samples, t_i can be set as different numbers to give appropriate weightings to these samples. In each loop, k_0 new columns and rows of tangential data are taken into consideration. In Step 4, we only need to update W, V, $\mathbb{L}$ and $\sigma\mathbb{L}$ instead of calculating them all from the beginning, which avoids repetitive computation. "Th" can be manually set to trade off between speed and accuracy.

Algorithm 2 MFTI of noisy data

1: Set $t_i \in [1, \min(m,p)]$ for $i = 1\ldots k$, construct matrix-format interpolation data R, L, W, V, Λ, M as Algorithm 1;
2: Set $\mathbf{II} = \emptyset$, $\mathbf{iI} = \{1, 2, \ldots, K\}$, $index = [1 : k_0 : K,\ 2 : k_0 : K, \ldots, k_0 - 1 : k_0 : K]$;
3: Do {
Set $\mathbf{ii}$ to be equal to the set of the No. $index(1)$, $index(2)$, ..., $index(k_0)$ elements of $\mathbf{iI}$; $\mathbf{II} = \mathbf{II}\bigcup\mathbf{ii}$, $\mathbf{iI} = \mathbf{iI} - \mathbf{ii}$;
4: Use $R(:,\mathbf{II})$, $L(\mathbf{II},:)$, $W(:,\mathbf{II})$, $V(\mathbf{II},:)$, $\Lambda(\mathbf{II},\mathbf{II})$, $M(\mathbf{II},\mathbf{II})$ to update W, V, $\mathbb{L}$ and $\sigma\mathbb{L}$;
5: Construct E, A, B, C following Algorithm 1. Calculate $H(s)$;
6: Calculate $err = \|w_i - H(\lambda_i)r_i\| + \|v_i - l_iH(\lambda_i)\|$ for $i \in \mathbf{iI}$; $[value, index] = sort(err)$;
7: If $\mathbf{iI} = \emptyset$, break;
} while $(mean(err) > Th)$

5. NUMERICAL EXAMPLES

[Example 1] Example 1 is used to illustrate the advantages of using MFTI over VFTI in the under-sampled case. The 8 scattering matrices are sampled from an order-150 system with 30 ports. As illustrated in Fig. 1, no obvious singular value drop can be detected by VFTI, but a sharp drop can be found in MFTI. Fig. 2 shows the Bode diagram (input 1-output 1) of the original system and the recovered system via both VFTI and MFTI. The MFTI-model fits well with the original system while the VFTI-model does not, which demonstrates that the samples are inadequate for VFTI while adequate for MFTI. Further experiments show that VFTI (using 180 matrix samples) requires about 30 times the samples of MFTI (6 matrix samples) to recover

Table 1: Interpolation of noisy data

Algorithms		Test 1 reduced order	Test 1 time(s)	Test 1 relative error	Test 2 reduced order	Test 2 time(s)	Test 2 relative error
VF(10 iterations)	n=140	140	3.1734	3.72e-1	140	2.8014	4.89e-1
	n=280	278	5.2518	7.33e-2	265	3.8492	9.11e-2
VFTI		95	0.4307	1.32e-1	98	0.2563	4.16e-1
MFTI-1	$t_i = 2$ (weight 1)	190	0.8254	9.60e-3	195	0.7178	3.14e-2
	$t_i = 3$ (weight 2)	252	1.4369	1.70e-3	260	1.3026	4.20e-3
MFTI-2(recursive)		130	0.6002	9.91e-3	147	0.4129	2.51e-2

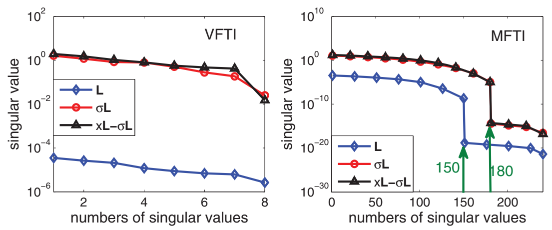

Figure 1: Singular value pattern of VFTI and MFTI

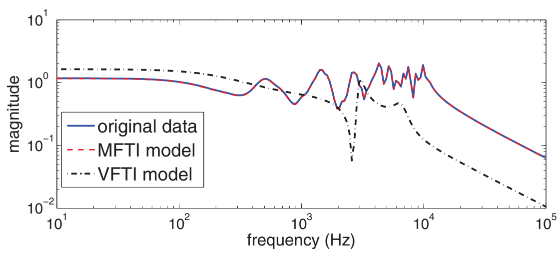

Figure 2: Bode diagrams of original and recovered systems using VFTI and MFTI.

the system. Besides, it is noticed that the singular values of $\mathbb{L}$, $\sigma\mathbb{L}$ and $x\mathbb{L} - \sigma\mathbb{L}$ drop at 150, 180 and 180, respectively, which confirms theorem 3.5.

[Example 2] The data in this example are measured from a 14-port power distribution network for INC board [10], with the order of the underlying system unknown. Table 1 shows the results. The data in Test 1 come from 100 uniformly distributed frequency samples, while those in Test 2 are 100 poorly distributed samples concentrated in the high-frequency band. VF (vector fitting), VFTI, MFTI-1 (Algorithm 1) and MFTI-2 (Algorithm 2) are all tested. The error vector is computed as $err_i = \frac{\|H(j2\pi f_i) - S(f_i)\|_2}{\|S(f_i)\|_2}$ for $i = 1, 2 \ldots, k$, whereas ERR is defined as $\|err\|_2/\sqrt{k}$. The CPU times are also recorded to compare the speeds of these algorithms. For Test 1, $t_i = 2$ and $t_i = 3$ demonstrate the dimension of the interpolation matrices. For Test 2, $t_i = 2$ and $t_i = 3$ represent two weighting choices (weight 1 and weight 2). They both satisfy the condition that $t_i \geq t_j$ for $i < j$ but more data are utilized for the latter. From Table 1, we conclude that MFTI is more accurate than VFTI, especially for ill-conditioned samples. Accuracy improves as t_i increases, which provides option to trade off between speed and accuracy. By employing recursive MFTI-2, the CPU time is only slightly greater than that of VFTI while the accuracy is significantly improved. Note that MFTI-2 automatically selects appropriate samples and thereby can yield good accuracy with moderate model size. Besides, it is shown that MFTI is much faster than the popular VF while at the same time achieves better accuracy.

6. CONCLUSION

Matrix-format tangential interpolation (MFTI), which generalizes vector-format tangential interpolation (VFTI), has been proposed and validated in this paper. The most significant advantage of MFTI over VFTI is that MFTI utilizes more information contained in the sampled matrices, and thus requires fewer samples to recover the system and yields better accuracy when interpolating under-sampled, noisy and/or ill-conditioned data. Besides, a minimal sampling theorem has been proposed. Numerical examples have confirmed the superiority of MFTI over VFTI in practical macromodeling.

Acknowledgment

This work was supported in part by the Research Grants Council of Hong Kong under Project HKU 718509E, and in part by the University Research Committee of The University of Hong Kong.

7. REFERENCES

[1] S.-H. Min and M. Swaminathan, "Construction of broadband passive macromodels from frequency data for simulation of distributed interconnect networks," *IEEE Trans. Electromagn. Compat.*, vol. 46, no. 4, pp. 544–558, Nov 2004.

[2] K. C. Sou, A. Megretski, and L. Daniel, "A quasi-convex optimization approach to parameterized model order reduction," *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, vol. 27, no. 3, pp. 456–469, Mar 2008.

[3] C. P. Coelho, J. R. Phillips, and L. M. Silveira, "Passive constrained rational approximation algorithm using Nevanlinna-Pick interpolation," in *Proc. Design, Automation and Test in Europe*, 2002, pp. 923–930.

[4] B. Gustavsen and A. Semlyen, "Rational approximation of frequency domain responses by vector fitting," *IEEE Trans. Power Del.*, vol. 14, no. 3, pp. 1052–1061, July 1999.

[5] A. Antoulas and B. Anderson, "On the scalar rational interpolation problem," *IMA J. Math. Control Inform.*, vol. 3, no. 2-3, pp. 61–68, 1986.

[6] A. Mayo and A. Antoulas, "A framework for the solution of the generalized realization problem," *Linear Algebra Appl.*, vol. 425, no. 2-3, pp. 634–662, Sep 2007.

[7] S. Lefteriu and A. C. Antoulas, "Modeling multi-port systems from frequency response data via tangential interpolation," in *IEEE Workshop on Signal Propagation on Interconnects*, May 2009.

[8] ——, "A new approach to modeling multiport systems from frequency-domain data," *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, vol. 29, no. 1, pp. 14–27, Jan 2010.

[9] R. Pintelon, P. Guillaume, Y. Rolain, J. Schoukens, and H. V. Hamme, "Parametric identification of transfer functions in the frequency domain - a survey," *IEEE Trans. Autom. Control*, vol. 39, no. 11, pp. 2245–2260, Nov 1994.

[10] S.-H. Min, "Automated construction of macromodels from frequency data for simulation of distributed interconnect networks," Ph.D. dissertation, Georgia Institute of Technology, Atlanta, Apr. 2004.