An Efficient Framework for Passive Compact Dynamical Modeling of Multiport Linear Systems

Zohaib Mahmood
MIT, Cambridge, MA
zohaib@mit.edu

Roberto Suaya
Mentor Graphics, Grenoble, France
roberto_suaya@mentor.com

Luca Daniel
MIT, Cambridge, MA
luca@mit.edu

***Abstract*—We present an efficient and scalable framework for the generation of guaranteed passive compact dynamical models for multiport structures. The proposed algorithm enforces passivity using frequency independent linear matrix inequalities, as opposed to the existing optimization based algorithms which enforce passivity using computationally expensive frequency dependent constraints. We have tested our algorithm for various multiport structures. An excellent match between the given samples and our passive model was achieved.**

I. Introduction

Generating a *passive* multiport model from available frequency domain data is an extremely challenging task. Considerable effort has been put into finding a convex relaxation to such highly nonlinear and non convex problem [1]–[3]. These techniques are computationally expensive. Some of these algorithms ([2], [3]) rely on enforcing the positive real lemma by constraining the real part of the impedance matrix to be positive definite over *all frequencies*. Although such a constraint can be certifiably enforced by using a Sum-Of-Squares (*SOS*) relaxation, it is normally a costly operation, specially when the constraints are defined on frequency dependent matrices such as in [2], [3]. In [1], passivity constraints are formulated as linear matrix inequalities, but the problem is still computationally expensive because of the large matrices used in defining the constraints. Furthermore, these optimization based algorithms exhibit poor scalability, and quite often exhaust computational resources, such as memory.

Researchers have been working on an iterative perturbation framework such as [4], [5]. In these techniques a stable but non-passive model is first identified. This non-passive model is then checked for passivity violations by examining if there exist pure imaginary eigenvalues of the corresponding Hamiltonian matrix. Finally, some parameters of the initially identified non-passive model are perturbed to correct for passivity violations. These techniques are computationally efficient, however since perturbing the system is an ill-posed problem, there is no guarantee that the final passivated model is optimal for accuracy. For instance we will show in Section VI that the passive models generated by using [5] may lose accuracy when the initial passivity violations are significant.

In this paper we present a framework for identifying passive dynamical models from frequency response data. We cast the problem as a standard semidefinte program (SDP) which can be solved by SDP solvers such as [6], [7].We solve the problem in two steps. First a set of common poles is identified using already established techniques [2], [8], [9]. Next, we identify residue matrices while simultaneously enforcing passivity.

Theoretically our identified models are restrictive in the sense that we are enforcing passivity through a sufficient but not necessary condition, however in practice these models can model most practical systems as is demonstrated in Section VI. By paying a small price in terms of accuracy we manage to solve larger problems within limited computational resources, such as memory, and gain orders of magnitude improvement in terms of speed compared to [1], [2]. We remark that although [1], [2] can generate more accurate models, their usability is limited to very small examples.

We use a framework similar to the one proposed in [10], however we overcome the underlying challenges which arise from the decoupling of the two steps (i.e. identification of stable poles and passive residues) by proposing an adaptive or assisted pole placement algorithm which improves the stable pole placement. The optimization problem in [10] suffers from poor conditioning and may run into problems for systems with a large number of poles. The algorithm proposed in this paper addresses the conditioning problem, and proposes an efficient way of preconditioning the matrices for the optimization problem, hence allowing us to model multiport structures with larger number of poles. Furthermore the algorithm proposed in this paper is tested on a set of challenging examples for which existing approaches do not perform well.

Although, passivity conditions similar to the ones used in this paper were derived in [11], [12], in these papers such conditions were used *only* to 'check' for passivity violations. In our proposed algorithm, these conditions are instead built into the model generation procedure to 'enforce' passivity. Also, no efficient algorithm was proposed in [11], [12] to rectify for passivity violations. For example in [11] it was proposed that the pole-residue pairs violating passivity conditions should be discarded, this is highly restrictive and can significantly deteriorate the accuracy. We instead propose that the identified residue matrices should conform to passivity conditions during the model generation, such that there are no passivity violation in the final model. The formulation presented in this paper, being convex, is guaranteed to converge to the global minimum and can be easily implemented using publicly available convex optimization solvers such as SeDuMi [6].

The remainder of the paper is organized as follows: Section II describes background on rational fitting of transfer matrices and the notion of passivity. Section III formulates the problem of passive fitting for multiport LTI systems. Section V describes our algorithm and implementation. Finally, Section VI demonstrates the effectiveness of the proposed approach in modeling various multiport structures.

978-3-9810801-8-6/DATE12/©2012 EDAA

II. Background

A. Rational Transfer Matrix Fitting

The problem of constructing a rational approximation of multiport systems can be formulated as finding residue matrices A_n, poles p_n and the matrices A_0 and A_∞ such that the generated model, defined by the transfer function $\hat{H}(s)$ in (1), approximates the given data.

$$\hat{H}(s) = \sum_{n=1}^{N} \frac{A_n}{s - p_n} + A_0 + sA_\infty \tag{1}$$

where A_n,A_0 and A_∞ are $M \times M$ residue matrices (assuming the system has M ports) and p_n are poles. Since most of the passive structures are reciprocal systems, A_n,A_0 and A_∞ are assumed to be symmetric matrices.

B. Passivity of a Transfer Matrix

Passivity is the inability of a system (or model) to generate energy. Since arbitrary connections of passive systems are guaranteed to be passive, passivity becomes an essential requirement if the model is to be used for time domain simulations while being interconnected with other subsystems. While it may be possible for a non-passive model to provide high accuracy in the frequency domain, the same model, when used in time domain simulation could produce extremely inaccurate results, resulting from passivity violations.

Passivity for an impedance or admittance system corresponds to *'positive realness'* of the transfer matrix. The transfer matrix $\hat{H}(s)$ is positive real if and only if it satisfies the following constraints

$$\overline{\hat{H}(\bar{s})} = \hat{H}(s) \tag{2a}$$

$$\hat{H}(s) \text{ is analytic in } Re\{s\} > 0 \tag{2b}$$

$$\hat{H}(j\omega) + \hat{H}(j\omega)^\dagger \succeq 0 \quad \forall\omega \tag{2c}$$

Where $Re\{\ \}$ denotes the real part, and $\dagger$ indicates the hermitian transpose.

The first condition (2a), commonly known as *conjugate symmetry*, ensures that the impulse response corresponding to $\hat{H}(s)$ is real. The second condition (2b) implies stability of the transfer function. A causal linear system in the transfer matrix form is stable if all of its poles are in the left half of the complex plane, i.e. all the poles have negative real part. The third and final condition (2c), i.e. the *positivity condition*, implies that the real symmetric part of the transfer matrix on the $j\omega$ axis is positive semidefinite.

III. Passive Model Formulation

Separating purely real from complex poles in (1) and evaluating it on the imaginary axis we obtain

$$\hat{H}(j\omega) = \sum_{n=1}^{N_r} \hat{H}_n^r(j\omega) + \sum_{n=1}^{N_c/2} \hat{H}_n^c(j\omega) + A_0 + j\omega A_\infty \tag{3}$$

$$\text{where: } \hat{H}_n^r(j\omega) = \frac{A_n^r}{j\omega - p_n^r} \tag{4}$$

$$\hat{H}_n^c(j\omega) = \frac{Re\{A_n^c\} + j\ Im\{A_n^c\}}{j\omega - Re\{p_n^c\} - j\ Im\{p_n^c\}} + \frac{Re\{A_n^c\} - j\ Im\{A_n^c\}}{j\omega - Re\{p_n^c\} + j\ Im\{p_n^c\}}$$

Here N_r and N_c denote the number of purely real and the number of complex poles, respectively and $N = N_r + N_c$. Also, $A_n^r \in \mathbb{R}^{M\times M}$, $A_n^c \in \mathbb{C}^{M\times M}$, $p_n^r \in \mathbb{R}$, $p_n^c \in \mathbb{C}$ $\forall n$, and $A_0, A_\infty \in \mathbb{R}^{M\times M}$, where M is the number of ports. In the following subsections, we consider one by one the implications of each passivity condition in (2) on the structure of (3).

A. Implications of Passivity on $\hat{H}(j\omega)$

Conjugate Symmetry requires that the complex-poles p_n^c and complex residue matrices A_n^c always appear in *complex-conjugate-pairs*. *Stability* requires $Re\{p_n\} < 0$.

The *positivity* condition for passivity (2c) is the most difficult condition to enforce analytically. To enforce (2c) we consider a sufficient condition which is described by the following lemma

Lemma 3.1: (Positive Summation Lemma) Let $\hat{H}(j\omega)$ be a stable and conjugate symmetric transfer matrix given by (3), then $\hat{H}(j\omega)$ satisfies positivity if $\hat{H}_n^r(j\omega)$, $\hat{H}_n^c(j\omega)$ and A_0 satisfy positivity $\forall n$. (using $H + H^\dagger = Re\{H\}$, for $H = H^T$) i.e.

$$Re\{\hat{H}_n^r(j\omega)\} \succeq 0, Re\{\hat{H}_n^c(j\omega)\} \succeq 0 \forall n \ \&\ A_0 \succeq 0 \implies Re\{\hat{H}(j\omega)\} \succeq 0 \tag{5}$$

Lemma 3.1 requires $\hat{H}_n^r(j\omega)$, $\hat{H}_n^c(j\omega)$ and A_0 to satisfy positivity. After rationalizing $\hat{H}_n^r(j\omega)$, $\hat{H}_n^c(j\omega)$ and using the fact that all poles are stable i.e. ($Re\{p_n\} < 0$) we get:

$$\begin{aligned} Re\{\hat{H}_n^r(j\omega)\} \succeq 0 &\implies \mathbf{A_n^r} \succeq 0\ \forall n = 1,...,N_r \\ Re\{\hat{H}_n^c(j\omega)\} \succeq 0 &\implies -Re\{p_n^c\}Re\{A_n^c\} \pm Im\{p_n^c\}Im\{A_n^c\} \succeq 0\ \forall n = 1,...,N_c/2 \\ Re\{A_0\} \succeq 0 &\implies A_0 \succeq 0 \end{aligned} \tag{6}$$

Hence in order to enforce passivity during the model identification, we can enforce the conditions described in (6) as constraints.

IV. Passive Modeling Algorithm

In this section we describe our algorithm for generation of passive compact dynamical models for multiport systems. In order to relax the original non-convex problem into a convex problem, we solve the optimization problem in two steps. The first step consists of finding a set of stable poles p_n for the system. The second step is finding a passive multiport dynamical model for the system, given stable poles from step 1. We also propose an adaptive algorithm to add new poles as described in Section V-E.

A. Step 1: Identification of stable poles

Several efficient algorithms already exist for the identification of stable poles for multiport systems. Some of the stable pole identification approaches use optimization based techniques such as in [2]. Some schemes such as [8], [9] find the location of stable poles iteratively. Any one of these algorithms can be used as the first step of our algorithm, where we identify a common set of stable poles for all the transfer functions in the transfer matrix. As mentioned before, to enforce conjugate symmetry, the stable poles can either be real or be in the form of complex-conjugate pairs. We have used [8] in our tests.

B. Step 2: Identification of Residue Matrices

In this section we formulate a convex optimization program for the identification of the residue matrices which correspond to passive $H(j\omega)$, using stable poles p_n from step 1. Combining the passivity conditions derived earlier (6), we get the following convex optimization problem.

$$\begin{aligned} \underset{A_n^r, A_n^c, A_0, A_\infty}{\text{minimize}} \quad & \sum_i \left| Re\{H_i\} - Re\{\hat{H}(j\omega_i)\} \right|^2 + \sum_i \left| Im\{H_i\} - Im\{\hat{H}(j\omega_i)\} \right|^2 \\ \text{subject to} \quad & A_0 \succeq 0, \quad A_n^r \succeq 0\ \forall n = 1,...,N_r \\ & -Re\{p_n^c\}Re\{A_n^c\} \pm Im\{p_n^c\}Im\{A_n^c\} \succeq 0\ \forall n = 1,...,N_c/2 \\ \text{where} \quad & \hat{H}(j\omega) = \sum_{n=1}^{N_r} \hat{H}_n^r(j\omega) + \sum_{n=1}^{N_c/2} \hat{H}_n^c(j\omega) + A_0 + j\omega A_\infty \end{aligned} \tag{7}$$

Here H_i refers to the given frequency samples. This final problem (7) is convex, since the objective function is a summation of L_2 norms. All the constraints in (7) are linear matrix inequalities. This convex optimization problem is a special case of semidefinite programming, enforcing linear matrix inequalities. This problem formulation can be solved much faster compared to other convex formulations [2], [3] where the unknown matrices are frequency dependent. The implementation details on how to solve this optimization problem using a standard semidefinite programming framework is described in Section V.

C. Proposed Algorithm

We summarize the identification procedure in Algorithm 1.

Algorithm 1 Passive Multiport Model Identification

Input: The set of frequency response samples $\{H_i, \omega_i\}$, and the number of poles N

Output: Passive model $\hat{H}(j\omega)$

1: Find a stable accurate system with N poles p_n

2: Solve the optimization problem (7) for A_n

3: Construct the model in pole/residue form as in (3)

This algorithm minimizes a cost function based on L_2 norm subject to linear matrix inequalities. Such a formulation is guaranteed to converge to the global minimum. Furthermore, the fact that this algorithm provides analytical expressions to enforce passivity in an efficient manner has potential, such as in future extensions to parameterized passive multiport models; or to include designers specific constraints such as ensuring a good match for qualify factors in RF inductor dynamical models.

D. Complexity

In the problem formulation (7), all the matrices are symmetric, allowing us to search only for the upper triangular part. Also, complex-valued residues are enforced by construction to appear in conjugate pairs, hence we solve directly only for half of the terms in the complex conjugate pair. This implies that the unknowns in our problem are $\eta = \frac{M(M+1)}{2}(N+2)$, where N is the number of poles and M is the number of ports. If we are given κ frequency samples then the complexity of solving our problem is roughly $O(\kappa^{\nu}\eta^{\gamma})$, where $\gamma = 2$ and $\nu = 2.5$ typically.

V. Implementation

In this section we will show how to cast the optimization problem (7) into a standard semidefinite program (SPD) format: SPDA [13] which can then be solved using any SPD solver [6], [7], [14].

A. Semidefinite Programs

Semidefinte Programs, or simply SDPs, belong to a special class of convex optimization problems where a linear cost function is minimized, subject to linear matrix inequalities. SDPs in standard form can be written as

$$\begin{aligned} &\text{minimize} \quad c^T x \\ &\text{subject to} \quad F_1 x_1 + F_2 x_2 + \cdots + F_n x_n - F_0 \succeq 0 \end{aligned} \tag{8}$$

where $F_0, F_1, \ldots F_n \in S^k$, where S^k indicates set of symmetric matrices of order $k \times k$.

B. Casting the Objective Function into Standard SDP

The objective function in our optimization problem (7) is a quadratic function, which needs to be reformulated as an SDP. For illustration we consider a generic quadratic function of the form $||Ax-b||^2$. We can cast this minimization problem into an equivalent semidefinite program as

$$\begin{aligned} &\underset{x}{\text{minimize}} ||Ax-b||^2 \\ &\quad \equiv \underset{t,x}{\text{minimize}} \quad t \\ &\quad \text{subject to} \quad \begin{bmatrix} tI & (Ax-b) \\ (Ax-b)^T & t \end{bmatrix} \succeq 0 \end{aligned} \tag{9}$$

In (9) we used the Schur Complement. The constraint in (9) can be transformed into standard SDP constraint resulting into the final program as:

$$\begin{aligned} &\underset{x}{\text{minimize}} ||Ax-b||^2 \implies \\ &\underset{t,x}{\text{minimize}} \quad t \\ &\text{subject to} \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix} t + \sum_{i=1}^{n} \begin{bmatrix} 0 & A_i \\ A_i^T & 0 \end{bmatrix} x_i - \begin{bmatrix} 0 & b \\ b^T & 0 \end{bmatrix} \succeq 0 \end{aligned} \tag{10}$$

here A_i indicates the i-th column of matrix A.

C. Casting the Positive Real Constraints into SDP

The constraints for our problem (7) are comprised of Linear Matrix Inequalities (LMIs). In this section we discuss how we can cast LMIs as SDP in standard format. For the purpose of illustration, we consider a generic LMI enforced on 2×2 matrices as in (11).

$$c_1 \begin{bmatrix} x_1 & x_2 \\ x_2 & x_3 \end{bmatrix} + c_2 \begin{bmatrix} x_4 & x_5 \\ x_5 & x_6 \end{bmatrix} \succeq 0 \tag{11}$$

such a constraint can be enforced as a standard SDP constraint as follows

$$\begin{aligned} &\begin{bmatrix} c_1 & 0 \\ 0 & 0 \end{bmatrix} x_1 + \begin{bmatrix} 0 & c_1 \\ c_1 & 0 \end{bmatrix} x_2 + \begin{bmatrix} 0 & 0 \\ 0 & c_1 \end{bmatrix} x_3 \quad + \ldots \\ &\begin{bmatrix} c_2 & 0 \\ 0 & 0 \end{bmatrix} x_4 + \begin{bmatrix} 0 & c_2 \\ c_2 & 0 \end{bmatrix} x_5 + \begin{bmatrix} 0 & 0 \\ 0 & c_2 \end{bmatrix} x_6 \quad \succeq 0 \end{aligned} \tag{12}$$

D. Pre-Conditioning of the SDP Matrices

When we formulate our problem as an SDP, the coefficients for the constraints and the objective function appear in the same matrix. These numbers may be several orders of magnitude apart from each other, causing ill-conditioning of the SDP matrices. To solve this problem, we normalize our constraints with the magnitude of the associated pole. If, however, stable residue matrices are available from step 1 the norms of the corresponding residue matrices should instead be used for normalization.

E. Adaptive Pole Placement

In the algorithm [10], placement of poles is independent from the identification of passive residue matrices. Such an isolation between the two steps simplifies the problem, the price is paid in terms of accuracy of the final passive model. Specially, when the fitting for stable poles reaches saturation, adding new poles leads to over fitting and does not help in the first step, while the fitting for residue matrices still has some room for improvement.

In order to address this point we propose the following semi-coupled procedure which takes into account the error in the passive model while adding new poles as described in Algorithm 2. We weight frequency samples (used as input to the stable pole identification procedure) with the normalized absolute value of the error obtained from passive identification using the previous set of poles. This way we ensure that the frequency band where mismatch was larger gets more weight.

Algorithm 2 Adaptive Pole Placement Algorithm

Input: $N_{initial}, N_{increment}, W_{initial} (W := weights)$

1: $N \leftarrow N_{initial}, W \leftarrow W_{initial}$
2: **while** $N \leq N_{desired}$ **do**
3: Stable Pole Identification $\leftarrow \{H_i, \omega_i, N, W\}$
4: $\hat{H} \leftarrow$ Passive Model Identification
5: $W \leftarrow |H_i - \hat{H}(j\omega_i)| \forall i$
6: $N = N + N_{increment}$
7: **end while**

VI. Results

In this section we shall present the efficacy of our algorithm in modeling various industry provided examples of passive multiport structures. All the computations, except for the 60 port example, are performed on a laptop with 2.1 *GHz* Core2Duo processor, 3 *GB* of main memory and running Windows 7. For demonstration purposes, the 60 port example was run on a server. Our modeling algorithm generates passive models from frequency response data formated in either Y or Z parameters. If the samples are given in $S-parameter$ format, they are first converted into equivalent $Y\ or\ Z-parameters$ and then fed to the passive modeling algorithm to generate passive models. Table I summarizes the performance of our algorithm. We have compared our algorithm with [1], [2], [5]. The implementation for [4] available at the authors' website is restricted to two port structures only, and hence could not be used for our comparison.

A. 1-Port Structure

The data set corresponding to the single port structure in Table I is contained in the code distribution package available on the website of the authors of [2]. We have tested [2] using the original implementation provided by the authors. For this structure we compute a model with 4 poles. The data is noisy and hence the peak and total errors for all the techniques are high, even with very good fits. The total error defined by (13) for our work and for [1], [5] are very similar. However, despite the fact that this was the only example for which we were able to test [2], the total error for [2] was very large i.e. 161%.

$$e_{total} = \sqrt{\sum_i |\Re H_i - \Re H(\omega_i)|^2 + \sum_i |\Im H_i - \Im H(\omega_i)|^2} \quad (13)$$

Here H_i and $H(\omega_i)$ are normalized values.

B. 4-Port Structure

In this section we discuss in detail the $4-port$ test case from Table I. To demonstrate the accuracy of our generated model, we've plotted Y_{11} in the form of a 3D plot in Figure 1. Here frequency is plotted on the x-axis, while the real and imaginary parts of Y_{11} are plotted on the y and z-axes respectively. We observe an excellent match between the output from our passive dynamical model (solid red line) and the data from the touchstone file (blue dots). We've also plotted the error defined by $e_{i,k}(\omega) = \frac{|H_{i,k}(j\omega) - \hat{H}_{i,k}(j\omega)|}{\max_{i,k,\omega} |H_{i,k}(j\omega)|}$ in Figure 2 which shows that the peak error is less than 2.5%.

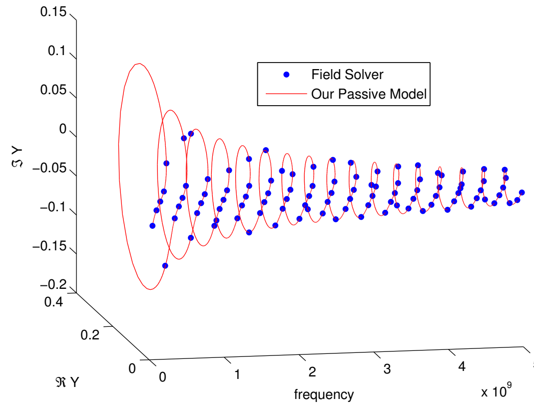

Fig. 1. **4-port structure:** Blue dots indicate the given Y_{11} frequency response samples. The Red solid line is the Y_{11} from our identified model.

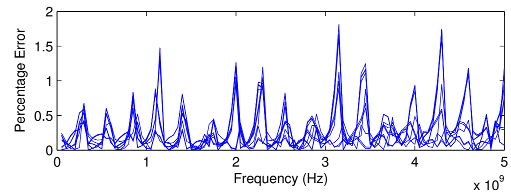

Fig. 2. **4-port structure:** Error between given data and the identified transfer matrix. Each curve corresponds to the mismatch for individual transfer function.

We provide a comparison with the algorithm presented in [1]. To make a fair comparison, both algorithms are implemented in matlab and the convex optimization problems are solved using the same solver SeDuMi. To compare scalability of the two algorithms, we perform comparison both in terms of CPU time and the memory requirement. Figure 3 plots the amount of allocated memory (left y-axis, blue solid lines) and the CPU time (right y-axis, red dotted lines) required to generate the models for both algorithms. It is clear from Figure 3 that for same model order, we get orders of magnitude improvement both in terms of speed and memory compared to [1]. On average we observed a significant 80× speedup. We note that the even for very small model orders, very large amount of memory is required by [1], and increasing model parameters quickly exhausts the resources. Our algorithm, on the other hand, utilizes only 1% of the memory required by the algorithm presented in [1].

Although our proposed technique is theoretically more restrictive then the one in [1] and [2]. We can see from the accuracy plots that our technique can successfully model practical passive networks, in a very reasonable time, and using only a fraction of the memory required for other convex optimization based algorithms. For this example none of the other convex optimization based algorithms were able to generate models with 45 common poles which were required to get sufficient accuracy.

TABLE I
RESULTS SUMMARY

No. of Ports	No. of Common Poles	Model Order	Generation Time (seconds)	Peak Percentage Error	Total error [This work]	Total error for [5]	Total error for [1]	Total error for [2]
1	4	4	0.38	5.7% (noisy data)	23.7% (noisy data)	21.8% (noisy data)	21.8% (0.53 sec) (noisy data)	161% (noisy data)
2	11	22	0.58	1.9%	7%	4%	4% (13 sec)	Out of Memory
4	45	180	144.4	1.8%	12%	8%	Out of Memory	Out of Memory
8	5	40	0.498	0.3%	2.7%	0.014%	0.33% (58 sec)	Out of Memory
12	9	108	98.30	0.007%	0.043%	1.71%	Out of Memory	Out of Memory
16	9	144	115.4	0.1%	2.391%	16.717%	Out of Memory	Out of Memory
60	5	300	9720	1.5%	-	-	-	-

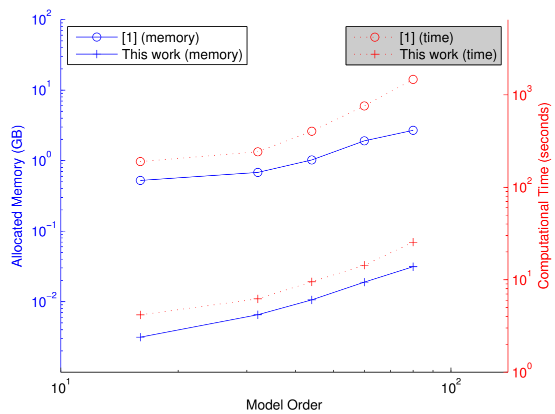

Fig. 3. **4-port structure:** Allocated memory (left y-axis, blue solid lines) and CPU time (right y-axis, red dotted lines) required to generate the models for both algorithms.

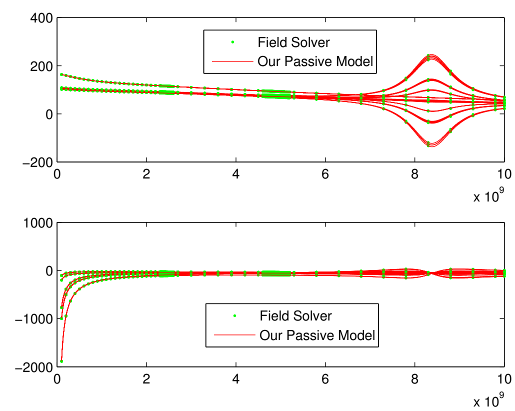

Fig. 4. **16-port structure:** Given frequency response samples (green dots) vs. the output from our identified models (red solid lines). Real and imaginary parts of Z-parameters are plotted in the top and bottom plots respectively.

C. 16-Port Structure

In this section we present a 16-port example. Important model parameters are summarized in Table I. The accuracy of the generated model is demonstrated in Figures 4 and 6. We see that our algorithm scales efficiently with the number of ports, compared to other optimization based algorithms such as [1], [2] which scale very badly and hence failed to generate model for our example. We also compared our passive model with the passive model generated by [5]. It is clear form Figure 5 that [5] lost significant accuracy during the passivity enforcement step, while our passive model achieved an excellent match with the given samples while simultaneously enforcing passivity. From Table I we can see that the total error, defined by (13) is 16.171% for [5] while it is only 2.391% for our algorithm. Passivity of our identified model was verified by Hamiltonian matrix based test and by checking the feasibility of the final solution.

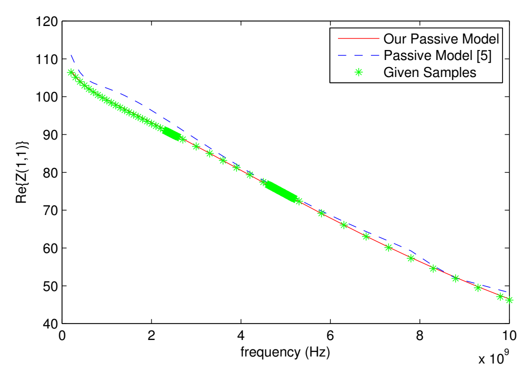

Fig. 5. **16-port structure:** Comparing $Re\{Z(1,1)\}$ from our passive model (solid red line) with [5] (dashed blue line) and given samples (asterisks). Our passive model matches the given samples perfectly while the passive model from [5] loses accuracy after passivity enforcement.

D. 60-Port Structure

To demonstrate that our algorithm can handle much larger structures, we have also tested it on a 60 port structure. This example was run on a server where it took approximately 3 hours to generate the model of order $N = 300$. In Figure 7 we plot magnitude of few Y parameters selected arbitrarily, both from the model and the given data. The generated model exhibits good accuracy where the peak error was less than 1.4%.We trust that a dedicated solver will significantly reduce the computational resources required.

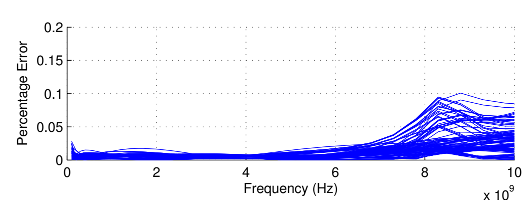

Fig. 6. **16-port structure:** Error between given data and the identified transfer matrix.

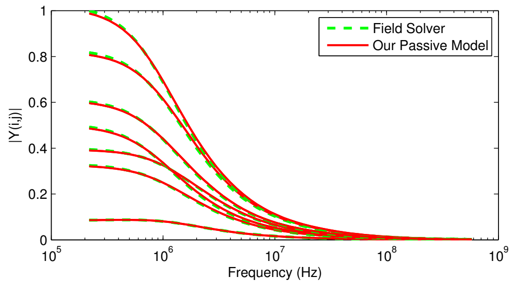

Fig. 7. **60-port structure:** Magnitude of frequency response samples from given data (green broken lines) and from our models (red solid line).

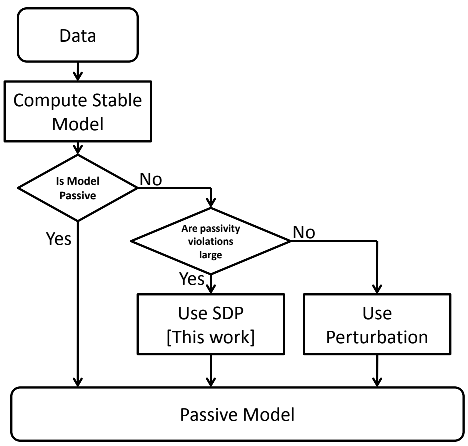

Fig. 8. Proposed framework for passive model identification

E. General Discussion

For all of the test cases, passivity of our identified model was verified by the absence of purely imaginary eigenvalues of the associated Hamiltonian matrix. Since we enforce passivity as constraints in the optimization problem, passivity of our models was further certified by checking feasibility of the solution i.e. by the presence of only *positive* eigenvalues for the semidefinite constraints defined in (7).

From Table I, we note that the algorithms presented in [1] and [2], ran out of memory for most of the test cases and did not generate the model. For the smaller test cases, where [1] did generate a model, we gained significant speed-up ($22\times$ and $116\times$) at the price of small degradation in accuracy. Compared to [5] we gain accuracy for some test cases, while for others, the algorithm in [5] performs better. This is because for some of the testcases the initially identified stable model in [5] is already passive or have minor passivity violations, in such a scenario, the model generated by [5] retains accuracy and performs better than our algorithm. For the test cases where the initial non-passive models have significant passivity violations, the perturbation step could severely degrade the accuracy for [5] (such as the 12 and 16 port examples in Table I). For such examples our algorithm gives better accuracy.

To summarize, we encourage the user to first try the relatively inexpensive algorithm [5]. If the initially generated stable model is already passive then the model can be used as is. On the other hand, if significant passivity violations are observed before the perturbation step in [5], then the user may switch to our algorithm to get better results. The proposed flow is demonstrated in the Figure 8.

VII. Conclusion

In this paper we have presented an efficient optimization based framework for the identification of passive multiport models. We have also presented a standard semidefinite programming based implementation. Furthermore, the underlying challenges related to over fitting and poor conditioning are addressed. The algorithm is supported by various multiport examples where accurate and certified passive models are identified in a reasonable time demonstrating the scalability of the algorithm. A comprehensive comparison with existing techniques is provided and a framework is proposed which uses our algorithm in conjunction with the existing approaches.

Acknowledgment

The authors would like to thank the following company and agencies for providing partial funding support for this project: Mentor Graphics, the FCRP Interconnect Focus Center, the Semiconductor Research Corporation.

References

[1] C. P. Coelho, J. R. Phillips, and L. M. Silveira, "A convex programming approach to positive real rational approximation," in *Proc. of the IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, November 2001, pp. 245–251.

[2] K. C. Sou, A. Megretski, and L. Daniel, "A quasi-convex optimization approach to parameterized model order reduction," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 3, Mach 2008.

[3] Z. Mahmood, B. Bond, T. Moselhy, A. Megretski, and L. Daniel, "Passive reduced order modeling of multiport interconnects via semidefinite programming," in *Proc. of the Design, Automation and Test in Europe (DATE)*, Dresden, Germany, March 2010.

[4] S. Grivet-Talocia, "Passivity enforcement via perturbation of hamiltonian matrices," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 51, no. 9, pp. 1755–1769, Sept. 2004.

[5] B. Gustavsen, "Fast passivity enforcement for pole-residue models by perturbation of residue matrix eigenvalues," *IEEE Trans. on Power Delivery*, vol. 23, no. 4, Oct. 2008.

[6] http://sedumi.ie.lehigh.edu/.

[7] K. C. T. R. H. Tutuncu and M. J. Todd, "Solving semidefinite-quadratic-linear programs using sdpt3," *Mathematical Programming*, vol. 95, no. 2, pp. 189–217, 2003.

[8] B. Gustavsen and A. Semlyen, "Rational approximation of frequency domain responses by vector fitting," *IEEE Trans. on Power Delivery*, vol. 14, no. 3, Jul 1999.

[9] R. Achar and M. Nakhla, "Simulation of high-speed interconnects," *Proceedings of the IEEE*, vol. 89, no. 5, pp. 693 –728, may 2001.

[10] Z. Mahmood and L. Daniel, "Circuit synthesizable guaranteed passive modeling for multiport structures," in *IEEE Intern. Workshop on Behavioral Modeling and Simulation*, San Jose, California, USA, September 2010.

[11] J. Morsey and A. Cangellaris, "Prime: passive realization of interconnect models from measured data," in *Electrical Performance of Electronic Packaging, 2001*, 2001, pp. 47 –50.

[12] S.-H. Min and M. Swaminathan, "Construction of broadband passive macromodels from frequency data for simulation of distributed interconnect networks," *Electromagnetic Compatibility, IEEE Transactions on*, vol. 46, no. 4, pp. 544 – 558, nov. 2004.

[13] K. Fujisawa and et al, "Sdpa (semidefinite programming algorithm) – user's manual," Tech. Rep., 1995.

[14] S. J. Benson and Y. Ye, "DSDP5: Software for semidefinite programming," Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, Tech. Rep. ANL/MCS-P1289-0905, Sep. 2005, submitted to ACM Transactions on Mathematical Software. [Online]. Available: http://www.mcs.anl.gov/~benson/dsdp