POLITECNICO DI TORINO
Repository ISTITUZIONALE

Efficient Localization Methods for Passivity Enforcement of Linear Dynamical Models

Original
Efficient Localization Methods for Passivity Enforcement of Linear Dynamical Models / Z., M., GRIVET TALOCIA, S., Chinea, A., Calafiore, G.C., L., D.. - In: IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS. - ISSN 0278-0070. - STAMPA. - 33:9(2014), pp. 1328-1341. [10.1109/TCAD.2014.2329418]

Availability:
This version is available at: 11583/2552541 since:

Publisher:
IEEE

Published
DOI:10.1109/TCAD.2014.2329418

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

08 October 2026

Efficient Localization Methods for Passivity Enforcement of Linear Dynamical Models

Zohaib Mahmood, *Student Member, IEEE,* Stefano Grivet-Talocia, *Senior Member, IEEE,* Alessandro Chinea, *Member, IEEE,* Giuseppe C. Calafiore, *Senior Member, IEEE,* and Luca Daniel, *Member, IEEE*

***Abstract*—This paper describes a novel approach for passivity enforcement of compact dynamical models of electrical interconnects. The proposed approach is based on a parameterization of general state-space scattering models with fixed poles. We formulate the passivity constraints as a unitary boundedness condition on the $\mathcal{H}_\infty$ norm of the system transfer function. When this condition is not verified, we use it as an explicit constraint within an iterative perturbation loop of the system state-space matrices. Since the resulting optimization framework is convex but non-smooth, we solve it via localization based algorithms, such as the ellipsoid and the cutting plane methods. The proposed technique solves two critical bottleneck issues of the existing approaches for passivity enforcement of linear macromodels. Compared to quasi-optimal schemes based on singular value or Hamiltonian eigenvalue perturbation, we are able to guarantee convergence to the optimal solution. Compared to convex formulations based on direct Bounded Real Lemma constraints, we are able to reduce both memory and time requirements by orders of magnitude. We demonstrate the effectiveness of our approach on a number of cases for which existing algorithms either fail or exhibit very slow convergence.**

***Keywords*—Passive macromodeling, convex optimization, localization methods, ellipsoid algorithm, cutting plane method.**

I. Introduction

SIGNAL and Power Integrity analysis of modern electronic systems are based on extensive numerical simulations, aimed at predicting the overall system performance and compliance since early design stages. Such simulations rely on accurate component macromodels that prove efficient when used in standard circuit simulators. The most critical components in terms of both influence on system performance and complexity in model extraction are the electrical interconnects used in signal and power distribution. Hence, over the last few years many techniques [1]–[36] have been developed for the extraction of compact interconnect macromodels.

Copyright (c) 2013 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending an email to pubs-permissions@ieee.org.

Z. Mahmood and L. Daniel are with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 02139 USA e-mail: zohaib@mit.edu, luca@mit.edu.

S. Grivet-Talocia is with the Department of Electronics, Politecnico di Torino, Torino 10129, Italy e-mail: stefano.grivet@polito.it.

A. Chinea is with IdemWorks s.r.l., Torino 10129, Italy e-mail: a.chinea@idemworks.com.

G. C. Calafiore is with the Department of Control and Computer Engineering, Politecnico di Torino, Torino 10129, Italy e-mail: giuseppe.calafiore@polito.it.

A common and very successful approach for the extraction of compact dynamical models of linear interconnects and more general linear time invariant systems, is based on frequency domain fitting techniques applied to measured or simulated frequency response data [1]–[5]. The system response is first approximated by a rational function. Then either an equivalent schematic or a state space form is synthesized to ease the interface with a circuit simulator.

In order to guarantee robust and stable numerical simulations, these models must satisfy physical properties such as causality, stability and passivity [6], [7]. Causality and stability can be easily enforced during the model fitting [4], [5], [8], [9]. Passivity enforcement is instead more challenging and requires special care. Passivity, which is often referred to as *dissipativity* in other scientific communities, intuitively refers to the inability of a dynamical system to generate energy in addition to the one already provided by external connections. As discussed in [7], passive models not only are physically consistent, but also guarantee numerically stable system-level simulations. Non-passive models may instead lead to instability even when their terminations or loads are passive [37].

Passive models can be generated using various approaches. Most of them involve minimizing some cost function, while imposing some form of passivity constraint. The various available techniques differ in the way the cost function and the passivity constraints are formulated, and in the way the resulting optimization problem is solved. All of these approaches represent different points in a trade-off between computational cost and optimality of the solution. We consider a solution as optimal when it represents the most accurate passive model in the considered parameterization class, with respect to the original non-passive model.

For instance, on the more optimal but less efficient side of the trade-off, the technique in [13] (available for download at [38]) presents a quasi-convex relaxation, with the advantage of simultaneous optimization of both model poles and residues including passivity constraints. This approach has also been extended in [13] to the generation of parameterized passive models. Although optimal, this formulations may require very large computational costs both in terms of memory and CPU time for systems with a large number of parameters.

A suboptimal but more efficient class of methods uses a 'two-steps' procedure. In the first step, poles and residues are fitted extremely efficiently without considering passivity constraints [4]. In a second step the poles are kept fixed and the residues are then optimized based on a variety of cost functions, passivity constraints and optimization techniques.

Within these 'two-steps' methods, some approaches guarantee optimality in the second step exploiting convex or quasi-convex formulations ([10]–[12], [14]–[17]). Such algorithms enforce passivity by defining constraints based on the positive real lemma or the bounded real lemma [39], [40]. For example, in [10], [16], [17] passivity constraints are formulated as linear matrix inequalities. As for [13], these formulations may result in a high computational costs both in terms of memory and CPU time, even for moderately complex models.

Also within the 'two-steps' methods, other approaches use a perturbation framework in the second step [18]–[26], where iterations are employed to correct passivity violations. These techniques are computationally very efficient. However their main drawback is that the underlying formulation does not guarantee convergence of the algorithm in all cases. Even when convergence takes place, the obtained solution is generally non-optimal in terms of accuracy. Some of these methods are based on iterative perturbation of the frequency dependent energy gain of the model ([20], [21], [27], [28]) through the solution of approximated local problems. Variants of the above schemes have been presented in [29]–[33]. A comprehensive comparison of such passive linear dynamical modeling techniques is available in [34].

There is thus a need for efficient algorithms that can generate passive linear dynamical models with guaranteed convergence to the global optimal solution, and characterized by moderate computational requirements both in terms of CPU time and memory occupation. An attempt was made in [35] to formulate the perturbation based approaches in a convex framework. This resulted in a convex but non-smooth optimization problem, which was solved in [35] using general descent schemes based on projected or alternate subgradient iterations. Even though the methods in [35] guarantee the optimality of the solution, they are highly sensitive to the given problem specific parameters, and requires tuning of the algorithmic coefficients for individual cases. If proper tuning is not achieved, the number of required iterations may grow very large, with an obvious increase in processing time.

In this paper, we solve the same convex non-smooth passivity enforcement problem of [35] by using efficient localization schemes such as the ellipsoid method [41] and the cutting plane method [42]. The main differentiating factor between our work and [35] is that the algorithms implemented in this paper are reliable and efficient, in the sense that they are less sensitive to the given problem parameters. As a result, they converge to the global optimal solution even for the challenging cases where the subgradient techniques in [35] require excessive iterations. Our framework generates optimal passive models with the same quality of [10], while requiring only a fraction of the memory. In addition, we derive a lower bound for the cost function that can be used to bracket the optimal solution. This bound, which provides a precise quantification of the distance between the original system and the class of passive models with prescribed poles, is here used to stop the iterations whenever the accuracy is satisfactory.

In summary, the main contributions of this paper are as follows:

- the application of efficient localization methods to solve the convex non-smooth passivity enforcement formulation presented in [35];
- the definition of a systematic procedure for the initialization of the iterations, by defining an initial feasible set (a hypersphere or a hypercube with controlled size) that is guaranteed to contain the global optimum;
- the construction of a lower bound on the objective function that gets tighter with every iteration;
- the implementation of efficient modifications and improvements (e.g., deep cuts formulations) to the standard localization methods, see, e.g., [43].

The resulting schemes provide an accuracy-controlled estimate of the passive model that is closest to an initial model in the desired norm, with an acceptable number of iterations and limited memory requirements.

This paper is organized as follows. Section II provides the required background on the problem formulation and on basic localization methods. Section III presents our implementation of localization methods for passivity enforcement. Numerical results are presented and discussed in Section IV.

II. Background

In this section we review some of the concepts used in this paper including linear dynamical modeling and localization methods. Additional definitions are provided in Appendix A.

A. Problem Description

Consider a nominal state-space dynamical model in scattering representation, characterized by its $n_p \times n_p$ transfer matrix

$$H(0,s) = C(sI - A)^{-1}B + D. \tag{1}$$

Here s is the Laplace variable, with state-space matrices $A \in \mathbb{R}^{q,q}, B \in \mathbb{R}^{q,n_p}, C \in \mathbb{R}^{n_p,q}, D \in \mathbb{R}^{n_p,n_p}$. The first argument of H, which is set to 0, shall be used later to parameterize a perturbation of the transfer matrix. We suppose that the dynamical model (1) is available through an approximation process. A common scenario is the availability of frequency samples $\{(\omega_\psi, S_\psi), \psi = 1,...,F\}$ of the scattering matrix for a linear device from either direct measurements or a full-wave electromagnetic field simulation. Common rational approximation schemes such as *Vector Fitting* [4], [5], [8], [9] can be applied to these samples in order to identify a state-space dynamical model (1) with minimal deviation from the raw data. In a least-squares formulation, this amounts to solving

$$\underset{A,B,C,D}{\text{minimize}} \sum_{\psi=1}^{F} ||H(0, j\omega_\psi) - S_\psi||^2. \tag{2}$$

Problem (2) is addressed in the literature [4], [5], [8], [9], hence we consider the nominal dynamical model (1) as our starting point.

System (1) is assumed to be asymptotically stable and the state-space realization is assumed to be minimal. A stable system (1) is passive if and only if its $\mathcal{H}_\infty$ norm is unitary bounded,

$$||H(0)||_{\mathcal{H}_\infty} = \sup_{\omega \in \mathbb{R}} \sigma_1(H(0, j\omega)) \leq 1, \tag{3}$$

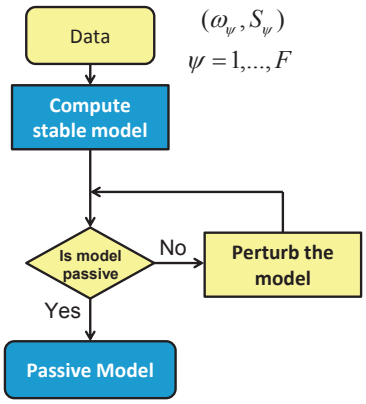

Fig. 1. Algorithmic flow for perturbation based passivity enforcement

where σ_1 denotes the largest singular value [18], [27], [44]. In the cases where (3) does not hold, the standard approach is to introduce a perturbation in the model, to make it passive. The two step process of perturbation based passivity enforcement framework is summarized in Figure 1.

A common choice is to perturb only the state-space C matrix which usually stores the residues for the partial fraction expansion of $H(0,s)$, see [34]. Matrix A is preserved in order to maintain the system poles, and matrix B does not need perturbation since it usually provides a static map between the inputs and the states. Matrix D, which corresponds to the high frequency response $(s \to \infty)$, is assumed to have $||D||_2 = \sigma_1(D) \leq 1$, which is necessary for passivity. The perturbed system is defined as

$$H(C_P, s) = (C + C_P)(sI - A)^{-1}B + D \tag{4}$$

where the perturbation matrix C_P is unknown. Supposing that the original system $H(0,s)$ is *not* passive, the goal here is to find the minimal perturbation such that the perturbed system $H(C_P,s)$ is passive. The problem can be formulated as

$$\underset{C_P}{\text{minimize}} \ ||C_P||_F \quad \text{s.t.} \quad ||H(C_P)||_{\mathcal{H}_\infty} \leq 1, \tag{5}$$

where the minimal perturbation condition is expressed, e.g., in terms of the Frobenius norm. It actually turns out that minimizing the state-space matrix perturbation C_P is not appropriate, since we would like to minimize the perturbation of the model response. This is achieved by the following weighted perturbation

$$\underset{C_P}{\text{minimize}} \ ||C_P G^T||_F \quad \text{s.t.} \quad ||H(C_P)||_{\mathcal{H}_\infty} \leq 1, \tag{6}$$

where G is the Cholesky factor of the controllability Gramian of the system. It is shown in [19] that (6) yields a solution that provides minimal impulse response perturbation in the $\mathcal{L}^2$ (energy) norm. Since (6) is readily reduced to (5) by a change of variable, we will base our derivations on (5) without loss of generality.

Using vectorized variable $x = vec(C_P) \in \mathbb{R}^n$ $(n = qn_p)$ and rewriting (5) we get

$$\underset{x}{\text{minimize}} \ f(x) \quad \text{s.t.} \quad h(x) \leq 1, \tag{7}$$

where

$$f(x) = ||x||_2 = ||C_P||_F, \qquad h(x) = ||H(C_P)||_{\mathcal{H}_\infty}. \tag{8}$$

Any local minimum is also a global minimum for Problem (7), if both f and h are convex. For the problem at hand, both f and h are norms, hence convex by virtue of the triangular inequality. Furthermore, this problem is feasible since one can always find at least one feasible point, namely $x = -x_c = -vec(C)$. This is because

$$x = -x_c \implies ||H(C_P)||_{\mathcal{H}_\infty} = ||H(-C)||_{\mathcal{H}_\infty} = ||D||_2 \leq 1 \tag{9}$$

by assumption. Since minimizing f is equivalent to minimizing f^2 (which is a strongly convex function), and since the feasible set is compact and nonempty, Problem (7) has a unique globally optimal solution. Notice further that, as described in [35], $h(x)$ is convex and continuous but non-smooth.

B. Computation of the $\mathcal{H}_\infty$ Norm

Accurate computation of the $\mathcal{H}_\infty$ norm is a critical step in solving (5). A naive way of computing the $\mathcal{H}_\infty$ norm is by sampling, where a given system $H(C_P, j\omega)$ is densely sampled in the frequency domain, ω, and the largest singular value $\sigma_1(H(C_P, j\omega_k))$ at each frequency sample is computed. The $\mathcal{H}_\infty$ norm is then calculated by finding the maximum, as described in (10).

$$||H(C_P)||_{\mathcal{H}_\infty} \approx \max_k \ \sigma_1(H(C_P, j\omega_k)) \tag{10}$$

The memory and time required for computing (10) are $O(n_p^2)$ and $O(\kappa n_p^3)$ respectively. Here κ is the number of samples. Computing the $\mathcal{H}_\infty$ norm via (10) is simple, however it may require computing a large number of samples consequently increasing the run-time of the algorithm.

An alternative method for computing the $\mathcal{H}_\infty$ norm of a transfer matrix is based on an iterative scheme which computes eigenvalues of a related Hamiltonian matrix. The details for such algorithms can be found in [18], [27], [44]. The memory and time required for these methods are $O(q^2)$ and $O(\tau q^3)$ respectively. Here τ is proportional to the number of iterations and is relatively small. In general, Hamiltonian matrix based methods are more accurate than the sampling methods, however they require more memory. In this paper we use [44] to compute the $\mathcal{H}_\infty$ norm.

C. Localization Methods

Localization methods are the optimization techniques where an initial set containing the global minimum becomes smaller at each iteration, thus bracketing the solution more and more tightly as the iterations progress. These methods are memory efficient and can handle non smooth problems, such as (7). The two localization methods that we employ in this work are the ellipsoid algorithm [41] and the cutting plane method [42]. In the following sections we provide an intuitive description of the two algorithms to solve the generic problem defined in (7).

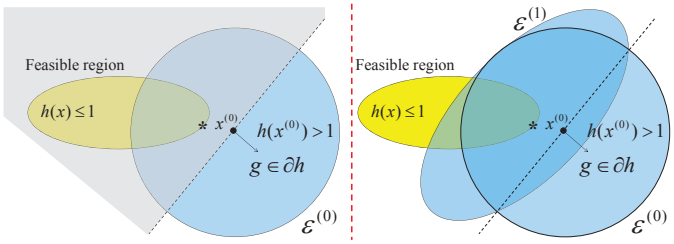

Fig. 2. First (infeasible) iteration of the ellipsoid algorithm.

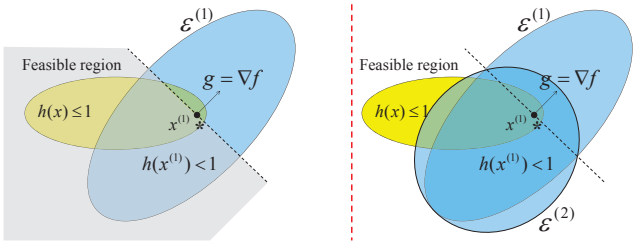

Fig. 3. Second (feasible) iteration of the ellipsoid algorithm.

1) The Ellipsoid Algorithm: The ellipsoid algorithm, proposed in [45], starts with an initial ellipsoid (defined in Appendix A-C) that is guaranteed to contain the global minimum. Figures 2 and 3 illustrate the first two iterations of the algorithm. In Figure 2 the yellow convex region describes the feasible region $h(x) \le 1$. The blue sphere describes the initial ellipsoid $\varepsilon^{(0)}$. The algorithm checks if the point $x^{(0)}$ is feasible. In case $x^{(0)}$ is not feasible, the algorithm defines a hyperplane given by the subgradient $g \in \partial h$ of the *constraint function*. The halfspace given by $\{x | g^T(x - x^{(0)}) \le 0\}$, which is shown by the shaded region in Figure 2, is the one where the constraint function decreases and hence intersects with the feasible region. In the next step, the algorithm updates the ellipsoid such that the updated ellipsoid $\varepsilon^{(1)}$ is the smallest ellipsoid containing the intersection of the original ellipsoid $\varepsilon^{(0)}$ and the half space $\{x | g^T(x - x^{(0)}) \le 0\}$. The updated ellipsoid $\varepsilon^{(1)}$ is shown in the right half of Figure 2.

For the next iteration, without loss of generality, we assume that the center $x^{(1)}$ of the updated ellipsoid lies in the feasible region. The algorithm defines a hyperplane by the gradient $g = \nabla f$ of the *objective function*. The halfspace given by $\{x | g^T(x - x^{(1)}) \le 0\}$, depicted by shaded region in Figure 3, is the one where the objective function decreases and hence contains the global minimum. In the next step, the algorithm updates the ellipsoid such that the updated ellipsoid $\varepsilon^{(2)}$ is the smallest ellipsoid defined by the intersection of current ellipsoid $\varepsilon^{(1)}$ and the half space $\{x | g^T(x - x^{(1)}) \le 0\}$. The updated ellipsoid $\varepsilon^{(2)}$ is shown in Figure 3.

This is continued until the size of the updated ellipsoid is small enough such that all of its interior points fall with in the δ neighborhood of the global minimum. The algorithmic details are described in Algorithm 1.

Algorithm 1 Ellipsoid Algorithm

Input: f, h and $\varepsilon^{(0)}(x^{(0)}, P_0)$ such that $x^* \in \varepsilon^{(0)}$, accuracy $\delta > 0$

1: Set $k = 0$
2: If $h(x^{(k)}) \le 1$, let $g_k \in \partial f(x^{(k)})$, else $g_k \in \partial h(x^{(k)})$
3: Evaluate $\eta_k = \sqrt{g_k^T P_k g_k}$
4: If $h(x^{(k)}) \le 1$ and $\eta_k \le \delta$, then return $x^{(k)}$ and quit
5: Update ellipsoid parameters $x^{(k+1)}$ and P_{k+1} using (11)
6: Let $k \leftarrow k+1$ and goto 1.

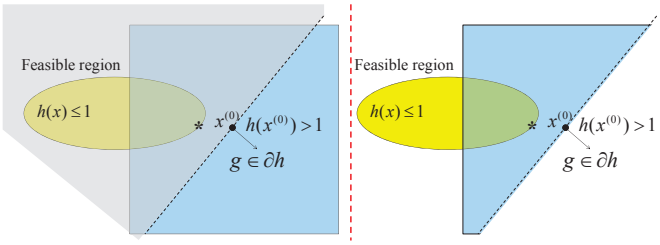

Fig. 4. First (infeasible) iteration of the cutting plane method.

The updates, as described in [41], are given by

$$x^{(k+1)} = x^{(k)} - \frac{1}{n+1} \frac{P_k g_k}{\sqrt{g_k^T P_k g_k}}$$

$$P_{k+1} = \frac{n^2}{n^2-1} \left(P_k - \frac{2}{n+1} \frac{P_k g_k g_k^T P_k}{g_k^T P_k g_k} \right). \qquad (11)$$

The ellipsoid algorithm is extremely efficient in terms of memory but it may take many iterations to converge because at each iteration, the volume reduction factor for the updated ellipsoid depends on n (size of x) as $\mathbf{vol}(\varepsilon_{k+1}) < e^{-\frac{1}{2n}} \mathbf{vol}(\varepsilon_k)$.

2) The Cutting Plane Method: The cutting plane method [42] also belongs to the class of localization methods. This approach uses a *polyhedron* (defined in Appendix A-D) instead of an ellipsoid to define the search region. The algorithm starts with an initial polyhedron $\mathcal{P}_0$ that is guaranteed to contain the global minimum. At each step the algorithm adds a new constraint to the current polyhedron $\mathcal{P}_k$ such that the updated polyhedron $\mathcal{P}_{k+1}$ is smaller and still contains the global minimum.

Figures 4 and 5 illustrate the first two iterations. Consider a scenario similar to the ellipsoid algorithm, where the yellow convex region describes the feasible region while the blue polyhedron defines the initial search space. The algorithm checks if the center of the polyhedron is feasible. When the center $x^{(0)}$ is infeasible the algorithm defines a cutting plane based on the subgradient $g \in \partial h$ of the *constraint function*. The half space $\{x | g^T(x - x^{(0)}) \le 0\}$ overlaps with the feasible region. The algorithm adds this cutting hyperplane to the definition of the polyhedron such that the updated polyhedron is the intersection of the original polyhedron and the shaded halfspace $\{x | g^T(x - x^{(0)}) \le 0\}$, as shown in Figure 4.

For the next iteration, without loss of generality, we assume that the center $x^{(1)}$ of the updated polyhedron falls in the feasible region. The algorithm defines a cutting plane by the

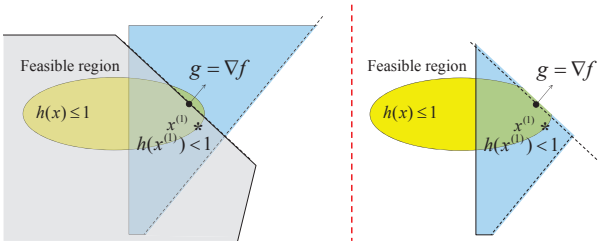

Fig. 5. Second (feasible) iteration of the ellipsoid algorithm.

gradient $g = \nabla f$ of the *objective function*, corresponding to the dotted line in Figure 5. The half space $\{x|g^T(x-x^{(1)}) \le 0\}$ is where the objective function decreases and hence contains the global minimum. The algorithm adds this cutting hyperplane to the definition of the polyhedron such that the updated polyhedron is the intersection of the original polyhedron and the shaded halfspace $\{x|g^T(x-x^{(0)}) \le 0\}$, as shown in Figure 5.

Algorithm 2 describes the cutting plane method conceptually. Since the cutting plane method is not a descent method, the algorithm keeps track of the best feasible solution f_{best} attained throughout all previous iterations.

Algorithm 2 Cutting Plane Method

Input: f, h and $\mathcal{P}_0$ such that $x^* \in \mathcal{P}_0$; initialize the lower bound $L_0 = 0$ and $f_{best} = f(x_{feasible})$; accuracy $\delta > 0$,set $k = 0$

1: If $h(x^{(k)}) \le 1$, let $g_k \in \partial f(x^{(k)})$, else $g_k \in \partial h(x^{(k)})$
2: Query the cutting plane oracle at $x^{(k)}$
3: If $h(x^{(k)}) \le 1$ and $|f_{best} - L_{best}| < \delta$, then return $x^{(k)}$ and quit
4: Update $\mathcal{P}$: add a new cutting plane $a_{k+1}^T z \le b_{k+1}$, $\mathcal{P}_{k+1} := \mathcal{P}_k \bigcap \{z|a_{k+1}^T z \le b_{k+1}\}$ here $a_{k+1} = g_k$ and if feasible $b_{k+1} = g^T x^{(k)}$ else $b_{k+1} = g^T x^{(k)} - h(x^{(k)}) + 1$
5: Update $x^{(k+1)} \in \mathcal{P}_{k+1}$, L_k
6: Update $L_{best} = \max_k L_k$, $f_{best} = \min_k f(x_{feasible}^{(k)})$
7: Let $k \leftarrow k+1$ and goto 1.

The cutting plane method is also efficient in terms of memory. However, compared to the ellipsoid method, the volume reduction factor for the updated polyhedron is less sensitive to n (size of x). An implementation of the cutting plane method based on a maximum volume ellipsoid reduces the volume of the polyhedron as, $\mathbf{vol}(\mathcal{P}_{k+1}) \le (1-1/n)\mathbf{vol}(\mathcal{P}_k)$ [46].

D. Passivity Enforcement Using Semidefinite Programming Based Methods

Passivity of a linear dynamical system described by scattering representation can be directly enforced by the bounded real lemma (or by the positive real lemma for hybrid representation) [10]. Passivity enforcement via bounded real lemma can be formulated as a semidefinite program (SDP) and solved using standard SDP solvers such as [47] to compute the global optimal solution.

$$\underset{B,C,D,W}{\text{minimize}} \sum_{\psi=1}^{F} ||H(0, j\omega_\psi) - S_\psi||^2 \tag{12}$$

$$\text{subject to} \begin{bmatrix} A^TW + WA & WB & C^T \\ B^TW & -I & D^T \\ C & D & -I \end{bmatrix} \preceq 0 \tag{13}$$

$$W = W^T \succeq 0$$

Problem (13) generates a passive model with *optimal* accuracy with in the class of *'two-steps'* based methods. The main drawback of these methods, however, is the excessive computational cost ($O(q^{5.5})$ [10]), due the introduction of a large slack Lyapunov matrix variable W. This limits the scalability of SDP based passive model generation algorithms. Hence the application of such algorithms is restricted to relatively smaller examples.

III. Passivity Enforcement Using Localization Methods

In this paper we employ the localization methods described in the Section II-C to solve the convex continuous but non-smooth problem described in the Section II-A. Specific challenges when using localization methods include how to define an initial set that is guaranteed to contain the global optimum, how to define a cutting plane that effectively reduces the size of the search space, and how to update the search space. In the following sections we provide the algorithmic details including solutions for all such issues.

A. Initialization

One of the main challenges in using localization methods is to define an initial set that is guaranteed to contain the global optimum. This initial set needs to be as small as possible, because for larger initial sets the algorithms may take more iterations to converge. In this section, we define an initial set in the form of a hypersphere, with radius R, that is guaranteed to contain the global optimum for our problem. We also compute an upper and a lower bound on R, which help us to pick a value of R that is appropriate.

1) Upper Bound on R (R_{UB}): Computation of an upper bound on R is pictorially described in Figure 6. To understand this, consider the following Lemma.

Lemma 3.1: Let $f(x) = ||x||_2$ be the cost function, and let the feasible set be defined by $h(x) \le 1$. Suppose that we are given *any* feasible point x_f such that $h(x_f) \le 1$. Then, the hypersphere centered at origin with radius equal to the Euclidean distance of x_f from the origin is guaranteed to contain the global optimum.

Proof: Let x_f be the given feasible point. We define the initial hypersphere with radius $R = ||x_f||_2$. Now suppose that the global optimal solution x^* lies outside the hypersphere, which by definition requires $||x^*||_2 > ||x_f||_2$. This means $f(x^*) > f(x_f)$ which leads to a contradiction because the condition of optimality requires $f(x^*) \le f(x_f)$. ■

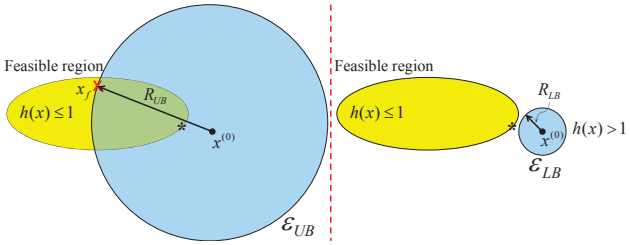

Fig. 6. [Left] R_{UB}, [Right] R_{LB}

As described in Section II, one of such feasible points is $x=-x_c$. We can compute an upper bound on the radius of the initial hypersphere by $R_{UB}=||x_c||_2$. Furthermore, we can use this feasible point to find an even smaller upper bound on the radius by using simple line search as

$$\begin{aligned}\beta^* &= \underset{\beta\in[-1,0]}{\operatorname{argmin}}\ f(\beta x_c) \quad \text{s.t.}\ \ h(\beta x_c)\le 1\\ R_{UB} &= ||\beta^* x_c||_2 \end{aligned} \tag{14}$$

If instead a weighted objective function, such as (6), is used the cost function becomes

$$f(x)=||Wx||_2, \tag{15}$$

where $W=G\otimes I$ and I is the identity matrix. The derivation of (15) is described in Appendix B. For this case, we can compute the upper bound on R by $R_{UB}=||Wx_c||_2$, which can be further shortened by the line search (14).

2) Lower Bound on R (R_{LB}): We define R_{LB} to be the radius of an infeasible hypershere ε_{LB} centered at the initial point $x^{(0)}$, as described in Figure 6. We assume that the initial unperturbed system is non-passive, hence the initial point $x^{(0)}$ is infeasible. The hypersphere ε_{LB} is a special ellipsoid for which the corresponding matrix $P_{LB}=R_{LB}^2 I$, where I is the identity matrix of size $n\times n$. Since all the points $x\in\varepsilon_{LB}$ are infeasible we get

$$h(x)>1 \quad \forall\ x\in\varepsilon_{LB}. \tag{16}$$

Since the constraint function $h(x)$ is convex, we have for all $x\in\varepsilon_{LB}$

$$\begin{aligned} h(x) &\ge h(x^{(0)})+\partial h(x^{(0)})^T(x-x^{(0)})\\ &\ge h(x^{(0)})+\inf_{z\in\varepsilon_{LB}}\partial h(x^{(0)})^T(z-x^{(0)})\\ &= h(x^{(0)})-\sqrt{\partial h(x^{(0)})^T P_{LB}\partial h(x^{(0)})}, \end{aligned} \tag{17}$$

where ∂h denotes a subgradient of the constraint function h. In the above derivation, we have used the fact that

$$\inf_{z\in\varepsilon_{LB}}\partial h(x^{(0)})^T(z-x^{(0)})=-\sqrt{\partial h(x^{(0)})^T P_{LB}\partial h(x^{(0)})}, \tag{18}$$

see Appendix C for a derivation. From (17), we note that $h(x^{(0)})-\sqrt{\partial h(x^{(0)})^T P_{LB}\partial h(x^{(0)})}>1$ implies $h(x)>1$, which means that all points in ε_{LB} are infeasible,

$$\begin{aligned} h(x) &\ge h(x^{(0)})-\sqrt{\partial h(x^{(0)})^T P_{LB}\partial h(x^{(0)})}>1,\\ (h(x^{(0)})-1)^2 &> \partial h(x^{(0)})^T R_{LB}^2 I\partial h(x^{(0)}). \end{aligned} \tag{19}$$

Solving (19) gives us R_{LB} as

$$R_{LB}=\frac{h(x^{(0)})-1}{\sqrt{\partial h(x^{(0)})^T\partial h(x^{(0)})}}=\frac{h(x^{(0)})-1}{||\partial h(x^{(0)})||_{\mathcal{L}2}}. \tag{20}$$

This means that for any initial hypershpere with $R<R_{LB}$, all the points in the hypersphere are infeasible. Hence, we must have $R>R_{LB}$ in order to guarantee that the hypersphere with radius R includes the global optimum.

3) Practical selection of R: In this section we provide guidelines on selecting the ratius R of the initial hypershpere. From earlier discussions in Section III-A2 and III-A1, the value of R must satisfy the following inequality

$$R_{LB}<R\le R_{UB} \tag{21}$$

From our experience with a diverse range of examples, the lower bound R_{LB} is tight. For most practical purposes $R=\gamma R_{LB}$ with $\gamma=2$, defines a large enough hypersphere. If however the resulting hypershpere is infeasible, this is detected and reported within few iterations. If this happens, the user may choose to increase the multiplying factor γ or simply pick the upper bound $R=R_{UB}$. In general, R is selected as

$$R=\min(\gamma R_{LB},R_{UB}) \tag{22}$$

B. Passivity Enforcement Using The Ellipsoid Algorithm

1) Initialization: The functions $f(x)$ and $h(x)$ for passivity enforcement are defined in (8). The gradient $\nabla f(x)$ and subgradients $\partial h(x)$ are derived in [35] and are not repeated here. We initialize Algorithm 1 by defining an initial ellipsoid, $\varepsilon^{(0)}$, that is guaranteed to contain the global optimum x^*. We define $\varepsilon^{(0)}(x^{(0)},P_0)$ to be the hypershpere, $P_0=R^2 I_n$, with the radius R and centered around the origin. The radius R is computed by (21).

2) Complexity: The main attractive feature of the ellipsoid algorithm is that it is extremely efficient in terms of the memory usage. Computationally, Algorithm 1 has two major components a) Computation of the $\mathcal{H}_\infty$ norm (or $h(x)$), described in Section II-B b) Updating and storing ellipsoid parameters. Their corresponding time and memory complexities are given in Table I.

TABLE I. COST PER ITERATION

Component	memory	time
(a) $\mathcal{H}_\infty$ norm	$O(q^2)$	$O(\tau q^3)$
(b) Ellipsoid Parameters	$O(n^2)\equiv O(q^2 n_p^2)$	$O(n^2)\equiv O(q^2 n_p^2)$

The algorithm requires only a modest storage of $O(n^2)$ to store the $P_k\in\mathbb{R}^{n\times n}$ matrix. As described in Section II-B, the memory required to compute the $\mathcal{H}_\infty$ norm is only $O(q^2)$ which is negligible. Similarly, the update cost for the algorithm in terms of time involves matrix vector products and is also of the order of $O(n^2)$. However, the cost per iteration is dominated by the computation of the $\mathcal{H}_\infty$ norm which is $O(\tau q^3)$.

3) Deep Cut Ellipsoid Algorithm: We use a modified version of the original ellipsoid algorithm known as the deep cut ellipsoid algorithm. These modifications, proposed in [43], do not come at any additional computational cost. Intuitively, in the deep cut ellipsoid algorithm at each step the hyperplane cuts off more region from the ellipsoid than the regular ellipsoid algorithm. A deep cut is defined by

$$g^T(x-x^{(k)})+g_{DC}^{(k)} \leq 0 \tag{23}$$

In the deep cut ellipsoid algorithm, the updated ellipsoid at each iteration is given by:

$$\varepsilon^{(k+1)} = \varepsilon^{(k)} \cap \{x | g^T(x-x^{(k)})+g_{DC}^{(k)} \leq 0\} \tag{24}$$

The newly introduced variable $g_{DC} \in \mathbb{R}_+$ is given by

$$g_{DC}^{(k)} = \begin{cases} f(x^{(k)}) - f_{best}^{(k)} & \text{if} \quad h(x^{(k)}) \leq 1 \\ h(x^{(k)}) - 1 & \text{if} \quad h(x^{(k)}) > 1, \end{cases} \tag{25}$$

here $f_{best}^{(k)}$ is the best objective value of the feasible iterates so far. The updates are given by ([43])

$$x^{(k+1)} = x^{(k)} - \frac{1+\alpha n}{n+1}\frac{P_k g_k}{\sqrt{g_k^T P_k g_k}}, \quad \text{where} \quad \alpha = \frac{g_{DC}^{(k)}}{\sqrt{g_k^T P_k g_k}}$$

$$P_{k+1} = \frac{n^2(1-\alpha^2)}{n^2-1}\left(P_k - \frac{2(1+\alpha n)}{(n+1)(1+\alpha)}\frac{P_k g_k g_k^T P_k}{g_k^T P_k g_k}\right). \tag{26}$$

We will show in the results section that the deep cut ellipsoid algorithm solves our problem much faster, when compared with the original ellipsoid algorithm used in [36]. Proof of convergence of the deep cut ellipsoid algorithm is provided in [43].

C. Passivity Enforcement Using The Cutting Plane Method

1) Initialization: We define the initial set (a polyhedron) required for the cutting plane method, described in Algorithm 2, to be the smallest hypercube $\mathcal{P}_0$ enclosing the hypershpere with radius R. The radius R is computed by (21). This hypercube $\mathcal{P}_0$ is guaranteed to contain the global minimum. $\mathcal{P}_0$ is centered around the origin with side length $2R$. We shall refer to the size of this hypercube as R.

2) Computing the center of the polyhedron: At each step of the cutting plane method we need to find a point $x^{(k+1)}$ in the interior of the polyhedron $\mathcal{P}_{k+1}$. In principle we can define $x^{(k+1)}$ to be anywhere in the polyhedron however, the benefit of having $x^{(k+1)}$ close to the center of $\mathcal{P}_{k+1}$ is that the updated polyhedron will be nearly half of the size of the original polyhedron. There are various ways to define and compute the center of a polyhedron, each having pros and cons. We choose x_{k+1} to be the analytic center of the inequalities defining $\mathcal{P}_{k+1} = \{z | a_i^T z \leq b_i,\ i = 1...q\}$. This gives us a well rounded performance both in terms of computational cost and convergence. We compute the analytic center by solving (27) using infeasible start Newton method

$$x_{k+1} = \underset{x}{\text{argmin}} - \sum_i \log(b_i - a_i^T x) \tag{27}$$

3) Complexity: The cutting plane method requires a modest storage of $O(mn^2)$ where n denotes the degrees of freedom ($x \in \mathbb{R}^n$) while m is the multiplying factor that increases with increasing iteration. The update cost for the algorithm is also of the order of $O(mn^2)$. As with the ellipsoid algorithm, the cost per iteration for the cutting plane method, as described in Table II, is dominated by the evaluation of the $\mathcal{H}_\infty$ norm. In general, the cutting plane method requires less iterations to converge compared to the ellipsoid algorithm. However, since we have to compute the center of the polyhedron at every step, the computational cost per iteration is slightly larger.

TABLE II. COST PER ITERATION

Component	memory	time
(a) $\mathcal{H}_\infty$ norm	$O(q^2)$	$O(\tau q^3)$
(b) Polyhedron Parameters	$O(mn^2) \equiv O(mq^2 n_p^2)$	$O(mn^2) \equiv O(mq^2 n_p^2)$

4) Epigraph Cutting Plane Method with Deep Cuts: The convergence rate of the cutting plane method improves significantly by solving the epigraph form of the problem and defining deep cuts. For the examples presented in this paper, this helps in reducing the number of iterations by more than 20% when compared with the original cutting plane method used in [36]. The epigraph form [46] of our problem (7) is given as follows

$$\begin{aligned} \underset{x,t}{\text{minimize}} \quad & t \\ \text{subject to} \quad & f(x) \leq t \\ & h(x) \leq 1 \end{aligned} \tag{28}$$

If at th k-th iteration $x^{(k)}$ is infeasible, we add the following cutting plane

$$h(x^{(k)}) + g^T(x-x^{(k)}) \leq 1, \quad \text{where} \quad g \in \partial h(x^{(k)})$$

If on the other hand, $x^{(k)}$ is feasible, we add the following two cutting planes

$$f(x^{(k)}) + g^T(x-x^{(k)}) \leq t \quad \text{and} \quad t \leq f(x^{(k)}) \tag{29}$$

where $g = \nabla f(x^{(k)})$.

D. Piecewise Linear Lower Bound on the Global Optimum

Suppose the function f and its gradient $g_i = \nabla f(x_i)$ have been evaluated at k distinct points x_i with $i = 1,\ldots,k$. Since f is convex, one can compute the local linear under estimators of f as follows

$$f(z) \geq f(x_i) + g_i^T(z-x_i), \quad \forall z, i = 1,...,k. \tag{30}$$

This implies that the function $\hat{f}$ defined as

$$f(z) \geq \hat{f}(z) = \max_{i=1,\ldots,k}(f(x_i) + g_i^T(z-x_i)) \tag{31}$$

is a convex piecewise linear global under estimator of f. Similarly, one can also find piecewise linear approximations

$\hat{h}(x) \leq 1$ of the constraint $h(x) \leq 1$. This helps formulating a piecewise linear relaxed problem

$$x^*_{LB} = \text{argmin } \hat{f}(x) \quad s.t.\ \hat{h}(x) \leq 1, \tag{32}$$

which can be solved using standard linear programming solvers (in order to solve (32), per iteration a standard primal-dual interior point method [48] would solve a linear system with approximately $4k+2n$ equations with $4k+2n$ unknowns). It can be shown that $L_k = ||x^*_{LB}||_2$ defines a lower bound on the global optimum of (7). This lower bound gets tighter with increasing iteration, so we can use it to define stopping criteria for the localization iterations. Furthermore, we can guarantee δ–optimality of the solution with the help of this lower bound (by definition, the distance of a δ–optimal solution from the true optimal solution is less than δ). For this purpose, note that the linear program (32) is rarely solved to compute the lower bound. The lower bound can be easily computed as a by-product when solving for the analytic center of the inequalities for the cutting plane method. For the ellipsoid method, the lower bound can be computed using the following expression

$$l_k = \max_{\{\forall k | h(x^{(k)}) \leq 1\}} \left(f(x^{(k)}) - \sqrt{g_k^T P_k g_k} \right). \tag{33}$$

IV. Results

In this section we provide small to medium scale challenging examples that arise in passive macromodeling. The computations were performed on a desktop computer having an Intel Xeon processor with 2.4 GHz clock. We have implemented and tested these algorithms in matlab without exploiting the speed-up provided by the parallel programming. For all cases an initial model was obtained using a fast vector fitting implementation of [4], [5], [49], whose complexity scales only as $O(Fq_p^2)$, where $q = q_p n_p$. This cost is negligible with respect to to the passivity check and enforcement. In fact, for the examples presented in this section, the first step took a fraction of a second to generate a nominal model with $< 0.1\%$ error, requiring less than one MB of memory.

A. Example 1: A 4-Port PCB Interconnect Structure

The first example we shall present is a 4-port coupled printed circuit board structure. The scattering matrix was measured from DC up to 2 GHz with a resolution of 10 MHz. The frequency response samples were processed with the vector fitting algorithm [4] to obtain an initial stable linear dynamical model with 272 states and 4 inputs/outputs. We used the ellipsoid algorithm with deep cuts to solve this problem. The algorithm was initialized with $R = 1.5R_{LB}$. The algorithm converged in about 25 iterations, as shown by the solid blue line in Figure 7. We report convergence by plotting the objective function. Since the localization-methods are not descent methods, the objective function may increase or decrease at each iteration. The algorithm converges when the current solution is in the δ neighborhood of the global minimum. We also plot the progress for the ellipsoid algorithm without deep cuts, shown by red diamonds in Figure 7. Note that by using deep cuts

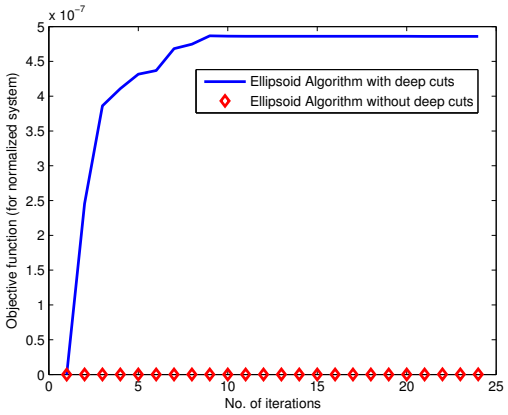

Fig. 7. Example 1: the cost function is plotted against the iteration number to show convergence. Solid blue line is for the ellipsoid algorithm with deep cuts, red diamonds are for the ellipsoid algorithm without deep cuts. (We initialize the algorithm with zero perturbation, i.e. zero cost, which refers to an infeasible point. The cost increases as the algorithm searches for a feasible point. Since the localization-methods are not descent methods, the objective function may increase or decrease at each iteration.)

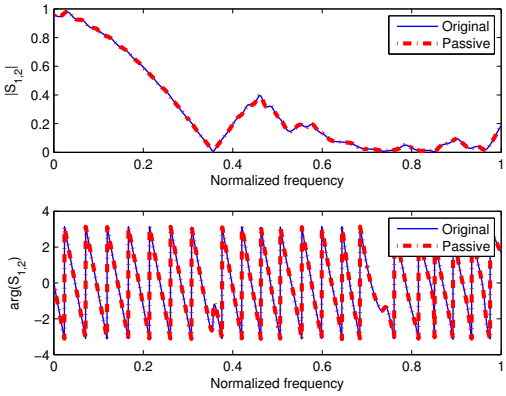

Fig. 8. Example 1: comparison between passive model and original non-passive model

we achieve a significant speed-up of $100\times$ over the original ellipsoid algorithm used in [36].

The total runtime required for this example was about 24 seconds. However, note that over 92% of the time was spent in computing the $\mathcal{H}_\infty$ norm while only 1.8 seconds were spent in the algorithmic computations. We have used a standard algorithm to compute the $\mathcal{H}_\infty$ norm [44].

Figure 8 demonstrates the accuracy of our passive model. We compare the $S_{1,2}$ response of our passive model with the one from the original non-passive model. Figure 9 compares the frequency dependent singular values of the original and the passive model. As reported in [35], standard techniques such as [19] generated only a suboptimal and less accurate passive model. The alternate gradient method presented in [35] took 900 iterations (833 seconds) to converge. This gives us a speed-up of $35\times$ over [35].

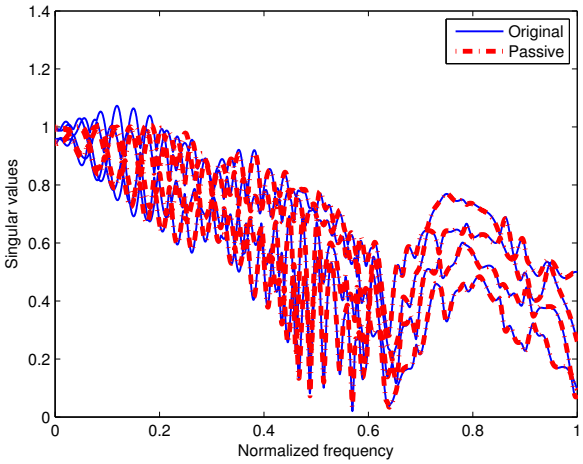

Fig. 9. Example 1: singular values of the original non-passive (solid blue) and the perturbed passive (dashed red) models are plotted versus frequency.

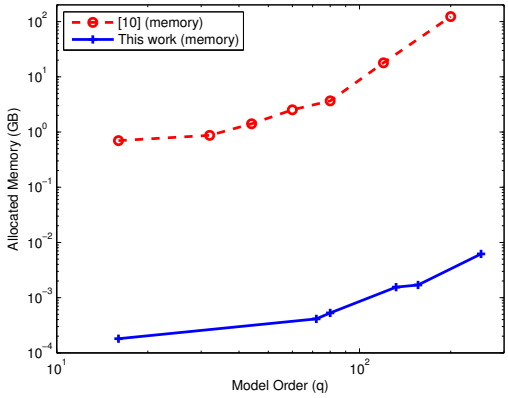

Fig. 10. Example 2: comparison of the memory requirements for our framework and [10].

B. Example 2: A 4-Port Transformer

We recall that the technique presented in [10] generates passive models with optimal accuracy, at the cost of large computing times and memory requirements, the latter being the most critical limitation, as described in Section II-D. With the proposed techniques, we are able to obtain the same optimal solution with a comparable accuracy, but requiring only a fraction of memory and time. We consider a 4-port transformer structure, and we compute a passive model using both [10] and the ellipsoid algorithm described in this work. We monitor the memory and time requirements for increasing model orders. Figures 10 and 11 plot the results. For models with the same dynamical order, our proposed ellipsoid algorithm based passivity enforcement framework achieved *four* orders of magnitude improvement in terms of memory with a speed-up of upto 1000× in terms of total run time. Figures 10 and 11 clearly demonstrate the scalability of our algorithm.

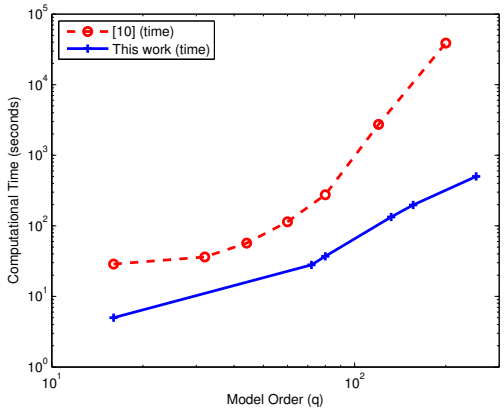

Fig. 11. Example 2: comparison of the time required for our framework and [10].

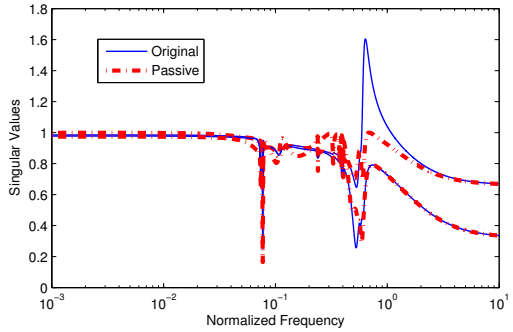

Fig. 12. Example 3: singular values of the original non-passive (solid blue) and perturbed passive (dashed red) models plotted against normalized frequency.

C. Example 3: Handling Large Passivity Violations

We consider a system for which the initial stable but non passive model has large passivity violations, as shown by the blue solid curves in Figure 12. This testcase corresponds to a SAW filter with 2 ports, whose initial model is characterized by 36 poles, with $C \in R^{2\times 72}$ and the corresponding unknown vector $x = vec(C_P) \in R^{144}$. To improve accuracy in the frequency response, we use the Cholesky factor of the controllability Gramian to define weights on the cost function as described in (6). We ancitipate the passivity enforcement results in Figure 13, which compares the scattering response $S_{1,1}$ of the original non-passive model with the perturbed passive model.

We solve first the problem using the cutting plane method. Even though we would like the initial hypercube to be as small as possible, in our experiments we vary the size of initial hypercube to demonstrate that the convergence of the cutting plane method is not too sensitive to the initial size. We define the initial hypercubes with sizes $R \in \{10.0, 1.0, 0.06\}$. Since for this testcase we have $R_{UB} = 1.0$ and $R_{LB} = 0.036$, all of these hypercubes are guaranteed to contain the global optimum. Figure 14 shows the convergence of the cost function

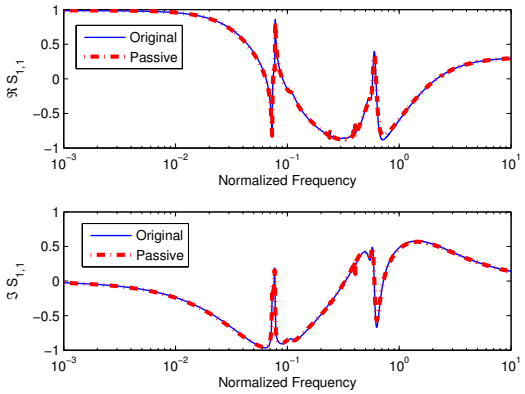

Fig. 13. Example 3: comparison between original non-passive and perturbed passive scattering response $S_{1,1}$

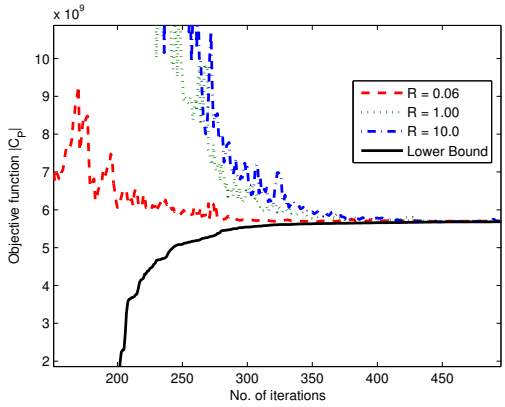

Fig. 14. Example 3: cutting plane method using analytic centering. The cost function is plotted against the iteration number for different initial hypercubes.

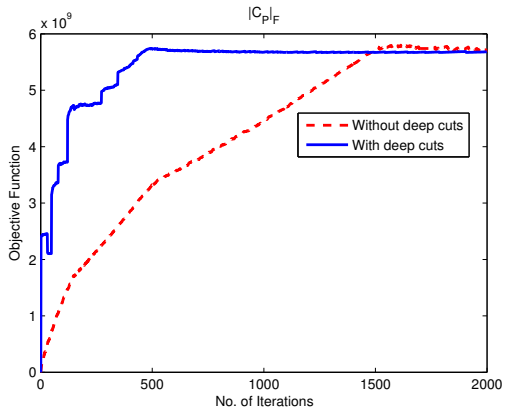

Fig. 15. Example 3: convergence of the ellipsoid algorithm with and without deep cuts.

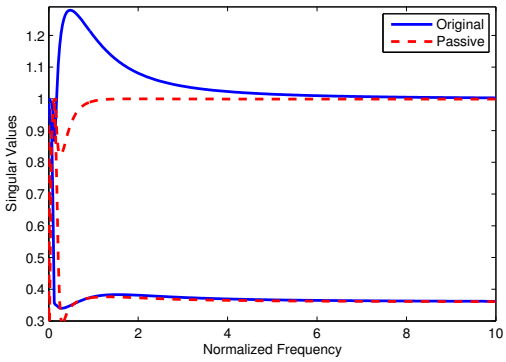

Fig. 16. Example 4: singular values of the original non-passive (solid blue) and perturbed passive (dashed red) models plotted against normalized frequency.

for different initial hypercubes. We note that increasing the size from 1 to 10 did not have a significant impact, since the algorithm converged in about 400 iterations. The figure also shows with a solid black line the lower bound on the cost function, obtained by solving (32). Using this lower bound, we can guarantee that the computed solution is δ–optimal, so that we control how far we are from the actual optimum. For this case, the algorithm converged in 30 seconds.

Next, we solve the problem using the ellipsoid method with deep cuts. The algorithm converged in less than 600 iterations, as shown by the solid blue line in Figure 15. The algorithm found a feasible point at the 500-th iteration. In total, it took only 20 seconds for the algorithm to converge. We also plot the progress for the ellipsoid algorithm without deep cuts, shown by the dashed red line in Figure 15. Note that by using deep cuts we achieve a speed-up of 3× over the original ellipsoid algorithm used in [36].

Note that both the ellipsoid algorithm and the cutting plane method converge to the same global optimum. The cutting plane method took more wall clock time than the ellipsoid algorithm even though it converged in less iterations. The reason is that the update cost for the cutting plane method increases with the number of iterations, where it remains constant for the ellipsoid algorithm.

For this example, standard techniques such as [19], [22] failed to generate a passive model. The alternate gradient algorithm presented in [35] took about 15,000 iterations (631 seconds) to converge. Hence we get an average speed up of 25× over [35].

D. Example 4: Wide Bandwidth Passivity Violation

In this example, we consider a system for which the initial stable but non passive model has passivity violations spread over a wide bandwidth, as shown by the solid blue curves in Figure 16. Such a behavior is known to pose serious challenges to existing perturbation based approaches. This testcase is characterized by 2 ports and 34 poles, with $C \in R^{2\times 68}$ and the unknown vector $x = vec(C_P) \in R^{136}$. The bounds on the initial feasible set are $R_{UB} = 2.4$ and $R_{LB} = 0.04$, respectively.

Figures 17 shows the convergence of the cutting plane method to the global optimum for three initial hypercubes of different sizes, together with the corresponding lower bound. A

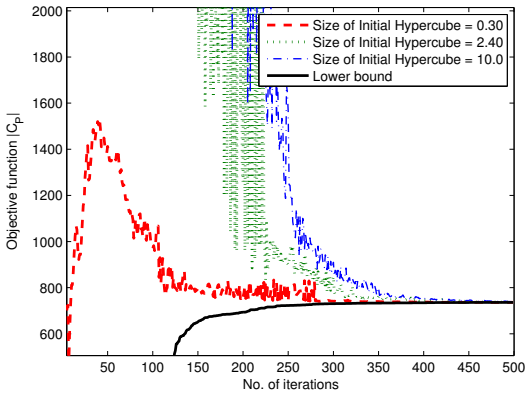

Fig. 17. Example 4: cutting plane method using analytic centering. The cost function is plotted against the iteration number for different initial hypercubes.

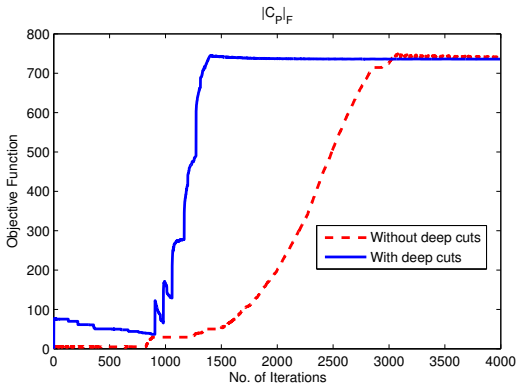

Fig. 18. Example 4: convergence of the ellipsoid algorithm with and without deep cuts.

total of 21 seconds were sufficient to reach convergence. Next, we solve the problem using the ellipsoid method with deep cuts. The algorithm converged in less than 2000 iterations, as shown by the blue line in Figure 18. The algorithm found a feasible point in 40 seconds at the 1400-th iteration. We also plot the progress for the ellipsoid algorithm without deep cuts, shown by the dashed red line in Figure 18. Note that by using deep cuts we achieve a speed-up of more than $2\times$ over the original ellipsoid algorithm used in [36]. For this example, we get an average speed up of $250\times$ over [35].

E. General Discussion

For the two localization-methods presented in this paper, there is no clear winner between the ellipsoid algorithm and the cutting plane method. As shown in the examples, wall clock time for both of the localization algorithms is similar. The ellipsoid algorithm is very attractive because its update cost is very small, however it may take more iterations if the initial set is larger. One of the key feature of the cutting plane method is that it is relatively less sensitive to the size of the initial search space. However the update cost for the cutting plane method increases with the iteration number. Hence, if the initial search space is tight with some confidence, we recommend using the ellipsoid method with deep cut, on the other hand if the initial search space is larger then we recommend using the cutting plane method.

It is important to note that the total runtime can be traded with accuracy in localization based methods. Since both an upper and a lower bound are available at each step, the algorithm can be stopped at any time as soon as the gap between the bounds is within the desired tolerance, even though successive iterations would improve the solution. This possibility is ruled out for common non-convex passivity enforcement schemes.

Note that the methods described in this paper can also be used to improve the accuracy of an existing perturbed passive system. For such systems, a feasible but inaccurate solution can be used to define the initial hypersphere and the algorithms described in this paper can be used to improve accuracy by providing the global optimal solution. Additionally, the lower bounds on the global optimum can be used to assess the quality of any possible available solution.

V. Conclusion

In this paper we employ non-smooth localization-based optimization methods to solve the formulation of passivity enforcement presented in [35]. We guarantee that the solution can be found up to any prescribed accuracy within a finite number of iterations. The techniques presented in this paper are memory efficient and demonstrate significant improvements both in terms of time and memory when compared with similar passive modeling algorithms.

Appendix A
Definitions

In this section we provide some basic definitions for subgradients, hyperplanes, ellipsoids and polyhedra. This material is available in the respective textbooks and online class materials such as [46], and is summarized here as a convenience for the readers.

A. Gradients and Subgradients

A vector $g \in \mathbb{R}^n$ is a subgradient of a convex function f at x, if for all $y \in \mathbf{dom} f$, it holds that

$$f(y) \geq f(x) + g^T(y-x). \tag{34}$$

If f is convex and differentiable at x, then $g = \nabla f$ is the unique subgradient. However, subgradients also exist at points where f is non differentiable. The set

$$\partial f(x) = \{g : g \text{ is a subgradient of } f \text{ at } x\} \tag{35}$$

is called the subdifferential of f at x.

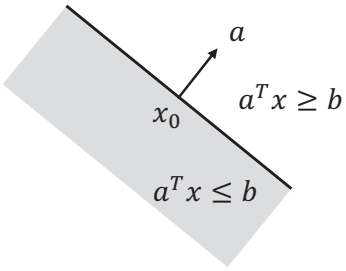

Fig. 19. A hyperplane defined by $a^T x = b$

B. Hyperplanes

A hyperplane is a set of the form

$$\{x \mid a^T x = b\} \tag{36}$$

where $a \in \mathbb{R}^n, a \neq 0$ and $b \in \mathbb{R}$. A hyperplane divides $\mathbb{R}^n$ in two halfspaces. It is illustrated in Figure 19. The halfspace determined by $a^T x \geq b$ (unshaded region) is the halfspace extending in the direction of a. The halfspace determined by $a^T x \leq b$ (shaded region) extends in the direction of $-a$. The vector a is the outward normal of this halfspace.

C. Ellipsoids

An ellipsoid $\mathfrak{E}$ is described by its center $x_c \in \mathbb{R}^n$ and the matrix P.

$$\mathfrak{E}(x_c, P) = \{x \mid (x - x_c)^T P^{-1}(x - x_c) \leq 1\}, \tag{37}$$

where $P = P^T \succ 0$ is a symmetric and positive definite matrix which defines the squared length of the semi-axes (its eigenvalues) and their orientation (its eigenvectors) for the ellipsoid $\mathfrak{E}$.

D. Polyhedra

A polyhedron is defined as the solution set of a finite number of linear equalities and inequalities. For our purpose, a polyhedron is defined only by the inequalities

$$\mathcal{P} = \{x \mid a_j^T x \leq b_j, j = 1, ..., m\}. \tag{38}$$

A polyhedron is thus the intersection of a finite number of halfspaces as shown in Figure 20.

APPENDIX B
DERIVATION OF THE WEIGHTED OBJECTIVE FUNCTION

In this section we derive (15) starting from (6)

$$\begin{aligned}
\underset{C_P}{\text{minimize}}\ ||C_P G^T||_F &\equiv \underset{C_P}{\text{minimize}}\ ||vec(C_P G^T)||_2 \\
&\equiv \underset{C_P}{\text{minimize}}\ ||vec(IC_P G^T)||_2 \\
&\equiv \underset{C_P}{\text{minimize}}\ ||(G \otimes I)vec(C_P)||_2 \\
&\equiv \underset{x}{\text{minimize}}\ ||(G \otimes I)x||_2 \\
&\equiv \underset{x}{\text{minimize}}\ ||Wx||_2 \\
&\equiv \underset{x}{\text{minimize}}\ f(x)
\end{aligned}$$

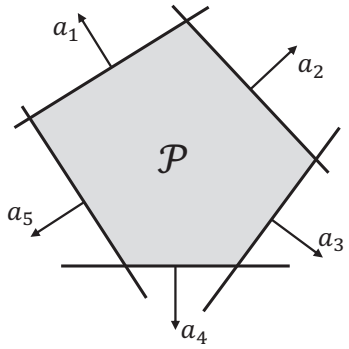

Fig. 20. A polyhedron defined by $\mathcal{P} = \{x | a_j^T x \leq b_j, j = 1, ..., 5\}$

APPENDIX C
DERIVATION OF (18)

In this section we derive (18). The left hand side of (18) is equivalent to the following convex optimization problem

$$\underset{x}{\text{minimize}}\ g^T x \quad \text{s.t.} \quad x^T P^{-1} x \leq 1 \tag{39}$$

The Lagrangian, $L(x, \lambda)$, and the dual function, $f_D(\lambda)$, of (39) are given by

$$\begin{aligned}
L(x, \lambda) &= g^T x + \lambda x^T P^{-1} x - \lambda \\
f_D(\lambda) &= \inf_x \quad L(x, \lambda) \\
f_D(\lambda) &= -\frac{1}{4\lambda} g^T P g - \lambda
\end{aligned} \tag{40}$$

Since the problem (39) is convex and $P \succ 0$, it satisfies Slater's conditions [50]. This implies that strong duality holds. Hence we compute the global optimal solution to (39) by solving the its dual problem which is given by

$$\underset{\lambda}{\text{maximize}}\ -\frac{1}{4\lambda} g^T P g - \lambda \quad \text{s.t.} \quad \lambda \leq 0 \tag{41}$$

The global optimal solution (λ^*) for (41) is given by

$$\lambda^* = \frac{1}{2}\sqrt{g^T P g} \tag{42}$$

Substituting (42) in (40) gives us the right hand side of (18), which is

$$f_D(\lambda^*) = -\sqrt{g^T P g} \tag{43}$$

REFERENCES

[1] D. G. Luenberger, *Optimization by vector space methods*. John Wiley & Sons, 1968.

[2] T. Kailath, *Linear systems*. Prentice-Hall Englewood Cliffs, NJ, 1980, vol. 1.

[3] J. Schoukens and R. Pintelon, *Identification of linear systems: a practical guideline to accurate modeling*. Pergamon press New York, 1991.

[4] 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.

[5] D. Deschrijver, M. Mrozowski, T. Dhaene, and D. De Zutter, "Macromodeling of multiport systems using a fast implementation of the vector fitting method," *Microwave and Wireless Components Letters, IEEE*, vol. 18, no. 6, pp. 383–385, 2008.

[6] M. R. Wohlers, *Lumped and distributed passive networks: a generalized and advanced viewpoint*. Academic press New York, 1969.

[7] P. Triverio, S. Grivet-Talocia, M. S. Nakhla, F. G. Canavero, and R. Achar, "Stability, causality, and passivity in electrical interconnect models," *Advanced Packaging, IEEE Transactions on*, vol. 30, no. 4, pp. 795–808, 2007.

[8] B. Gustavsen and A. Semlyen, "A robust approach for system identification in the frequency domain," *Power Delivery, IEEE Transactions on*, vol. 19, no. 3, pp. 1167–1173, 2004.

[9] D. Deschrijver, B. Haegeman, and T. Dhaene, "Orthonormal vector fitting: A robust macromodeling tool for rational approximation of frequency domain responses," *Advanced Packaging, IEEE Transactions on*, vol. 30, no. 2, pp. 216–225, 2007.

[10] 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.

[11] B. Dumitrescu, "Parameterization of positive-real transfer functions with fixed poles," *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol. 49, no. 4, pp. 523–526, 2002.

[12] H. Chen and J. Fang, "Enforcing bounded realness of s parameter through trace parameterization," in *Electrical Performance of Electronic Packaging, 2003*. IEEE, 2003, pp. 291–294.

[13] 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.

[14] 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.

[15] L. De Tommasi, M. de Magistris, D. Deschrijver, and T. Dhaene, "An algorithm for direct identification of passive transfer matrices with positive real fractions via convex programming," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 24, no. 4, pp. 375–386, 2011.

[16] Z. Mahmood and L. Daniel, "Circuit synthesizable guaranteed passive modeling for multiport structures," in *Behavioral Modeling and Simulation Conference (BMAS), 2010 IEEE International*. IEEE, 2010, pp. 19–24.

[17] Z. Mahmood, R. Suaya, and L. Daniel, "An efficient framework for passive compact dynamical modeling of multiport linear systems," in *DATE*, March 2012, pp. 1203–1208.

[18] S. Boyd, V. Balakrishnan, and P. Kabamba, "A bisection method for computing the h norm of a transfer matrix and related problems," *Mathematics of Control, Signals and Systems*, vol. 2, no. 3, pp. 207–219, 1989.

[19] 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.

[20] D. Saraswat, R. Achar, and M. S. Nakhla, "A fast algorithm and practical considerations for passive macromodeling of measured/simulated data," *Advanced Packaging, IEEE Transactions on*, vol. 27, no. 1, pp. 57–70, 2004.

[21] ——, "Global passivity enforcement algorithm for macromodels of interconnect subnetworks characterized by tabulated data," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 13, no. 7, pp. 819–832, 2005.

[22] S. Grivet-Talocia and A. Ubolli, "On the generation of large passive macromodels for complex interconnect structures," *Advanced Packaging, IEEE Transactions on*, vol. 29, no. 1, pp. 39–54, 2006.

[23] S. Grivet-Talocia, "An adaptive sampling technique for passivity characterization and enforcement of large interconnect macromodels," *Advanced Packaging, IEEE Transactions on*, vol. 30, no. 2, pp. 226–237, 2007.

[24] 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.

[25] B. Gustavsen and A. Semlyen, "Fast passivity assessment for s-parameter rational models via," *IEEE Transactions on Microwave Theory and Techniques*, vol. 56, no. 12, p. 2701, 2008.

[26] A. Semlyen and B. Gustavsen, "A half-size singularity test matrix for fast and reliable passivity assessment of rational models," *Power Delivery, IEEE Transactions on*, vol. 24, no. 1, pp. 345–351, 2009.

[27] B. Gustavsen and A. Semlyen, "Enforcing passivity for admittance matrices approximated by rational functions," *Power Systems, IEEE Transactions on*, vol. 16, no. 1, pp. 97–104, 2001.

[28] B. Gustavsen, "Computer code for passivity enforcement of rational macromodels by residue perturbation," *Advanced Packaging, IEEE Transactions on*, vol. 30, no. 2, pp. 209–215, 2007.

[29] ——, "Fast passivity enforcement of rational macromodels by perturbation of residue matrix eigenvalues," in *Signal Propagation on Interconnects, 2007. SPI 2007. IEEE Workshop on*. IEEE, 2007, pp. 71–74.

[30] A. Lamecki and M. Mrozowski, "Equivalent spice circuits with guaranteed passivity from nonpassive models," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 55, no. 3, pp. 526–532, 2007.

[31] S. Grivet-Talocia and A. Ubolli, "Passivity enforcement with relative error control," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 55, no. 11, pp. 2374–2383, 2007.

[32] A. Ubolli and S. Grivet-Talocia, "Weighting strategies for passivity enforcement schemes," in *Electrical Performance of Electronic Packaging, 2007 IEEE*. IEEE, 2007, pp. 55–58.

[33] C. S. Saunders, J. Hu, C. E. Christoffersen, and M. B. Steer, "Inverse singular value method for enforcing passivity in reduced-order models of distributed structures for transient and steady-state simulation," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 59, no. 4, pp. 837–847, 2011.

[34] S. Grivet-Talocia and A. Ubolli, "A comparative study of passivity enforcement schemes for linear lumped macromodels," *IEEE Trans. on Advanced Packaging*, vol. 31, no. 4, Nov. 2008.

[35] G. Calafiore, A. Chinea, and S. Grivet-Talocia, "Subgradient techniques for passivity enforcement of linear device and interconnect macromodels," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 60, no. 10, pp. 2990 –3003, oct. 2012.

[36] Z. Mahmood, A. Chinea, G. C. Calafiore, S. Grivet-Talocia, and L. Daniel, "Robust localization methods for passivity enforcement of linear macromodels," in *Proc. of the IEEE workshop on Signal and Power Integrity*, Paris, France, May 2013.

[37] S. Grivet-Talocia, "On driving non-passive macromodels to instability," *International Journal of Circuit Theory and Applications*, vol. 37, no. 8, pp. 863–886, 2009.

[38] "Squid: Stable quasiconvex identification." [Online]. Available: http://www.mit.edu/~dluca/squid/

[39] S. P. Boyd, *Linear matrix inequalities in system and control theory*. Siam, 1994, vol. 15.

[40] J. R. Phillips, L. Daniel, and L. M. Silveira, "Guaranteed passive balancing transformations for model order reduction," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 22, no. 8, pp. 1027–1041, 2003.

[41] R. G. Bland, D. Goldfarb, and M. J. Todd, "The ellipsoid method: A survey," *Operations Research*, pp. 1039–1091, 1981.

[42] J. E. Kelley Jr, "The cutting-plane method for solving convex programs," *Journal of the Society for Industrial & Applied Mathematics*, vol. 8, no. 4, pp. 703–712, 1960.

[43] J. Frenk, J. Gromicho, and S. Zhang, "A deep cut ellipsoid algorithm for convex programming: Theory and applications," *Mathematical Programming*, vol. 63, no. 1-3, pp. 83–108, 1994. [Online]. Available: http://dx.doi.org/10.1007/BF01582060

[44] N. Bruinsma and M. Steinbuch, "A fast algorithm to compute the h-norm of a transfer function matrix," *Systems & Control Letters*, vol. 14, no. 4, pp. 287–293, 1990.

[45] L. G. Khachiyan, "Polynomial algorithms in linear programming," *USSR Computational Mathematics and Mathematical Physics*, vol. 20, no. 1, pp. 53–72, 1980.

[46] S. P. Boyd, "Course materials." [Online]. Available: http://www.stanford.edu/~boyd/teaching.html

[47] J. F. Sturm, "Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones," *Optimization methods and software*, vol. 11, no. 1-4, pp. 625–653, 1999.

[48] S. Mehrotra, "On the implementation of a primal-dual interior point method," *SIAM Journal on optimization*, vol. 2, no. 4, pp. 575–601, 1992.

[49] A. Chinea and S. Grivet-Talocia, "On the parallelization of vector fitting algorithms," *Components, Packaging and Manufacturing Technology, IEEE Transactions on*, vol. 1, no. 11, pp. 1761–1773, 2011.

[50] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

Zohaib Mahmood (M'10) received the bachelors of science (B.Sc.) degree with honors in electrical engineering from the University of Engineering and Technology Lahore, Pakistan in 2007, and the masters of science (S.M.) degree in electrical engineering and computer science from Massachusetts Institute of Technology (MIT), in 2010; where he is currently a candidate for doctorate.

Stefano Grivet-Talocia (M'98–SM'07) received the Laurea and the Ph.D. degrees in electronic engineering from Politecnico di Torino, Italy. From 1994 to 1996, he was with the NASA/Goddard Space Flight Center, Greenbelt, MD, USA. Currently, he is an Associate Professor of Circuit Theory with Politecnico di Torino. His research interests are in passive macromodeling of lumped and distributed interconnect structures, modeling and simulation of fields, circuits, and their interaction, wavelets, time-frequency transforms, and their applications. He is author of more than 120 journal and conference papers. He is co-recipient of the 2007 Best Paper Award of the IEEE Trans. Advanced Packaging. He received the IBM Shared University Research (SUR) Award in 2007, 2008 and 2009. Dr. Grivet-Talocia served as Associate Editor for the IEEE TRANSACTIONS ON ELECTROMAGNETIC COMPATIBILITY from 1999 to 2001. He is co-founder and President of IdemWorks.

Alessandro Chinea received the Laurea Specialistica (M.Sc.) and Ph.D. degrees in electronic engineering from Politecnico di Torino, Italy in 2006 and 2010, respectively. In 2009 he spent a period at the Department of Information Technology (INTEC) of the Ghent University, Belgium, working under the supervision of the professors T. Dhaene and L. Knockaert. Since 2012 he works at the IdemWorks s.r.l. as a senior engineer. His research interests concern passive macromodeling of electrical interconnects for electromagnetic compatibility and signal/power integrity problems. Dr. Chinea received the Optime Award from the Unione Industriale di Torino and he was selected for the IBM EMEA Best Student Recognition Event 2006.

G.C. Calafiore received the "Laurea" degree in Electrical Engineering from Politecnico di Torino in 1993, and the Ph.D. degree in Information and System Theory from Politecnico di Torino, in 1997. Since 1998 he was with the faculty of Dipartimento di Automatica e Informatica, Politecnico di Torino, where he currently serves as a tenured Associate Professor. Dr. Calafiore held several visiting positions at international institutions: at the Information Systems Laboratory (ISL), Stanford University, California, in 1995; at the Ecole Nationale Suprieure de Techniques Avances (ENSTA), Paris, in 1998; and at the University of California at Berkeley, in 1999, 2003 and 2007. He had an appointment as a Senior Fellow at the Institute of Pure and Applied Mathematics (IPAM), University of California at Los Angeles, in 2010. He was an Associate Editor for the IEEE Transactions on Systems, Man, and Cybernetics (T-SMC), for the IEEE Transactions on Automation Science and Engineering (T-ASE), and for Journal Europen des Systmes Automatiss (JESA). He was the coordinator of the Specializing Master in Automatica and Control Technologies (MACT) at Politecnico di Torino. Dr. Calafiore is the author of more than 150 journal and conference proceedings papers, and of eight books. He is a senior member of the IEEE. He received the IEEE Control System Society "George S. Axelby" Outstanding Paper Award in 2008. His research interests are in the fields of convex optimization, randomized algorithms, identification, and control of uncertain systems, with applications ranging from finance and economic systems to robust control, pattern recognition and robotics.

Luca Daniel (S'98-M'03) received the Laurea degree (summa cum laude) in electronic engineering from the Universita di Padova, Italy, in 1996, and the Ph.D. degree in electrical engineering from the University of California, Berkeley, in 2003. He is an Associate Professor in the Electrical Engineering and Computer Science Department of the Massachusetts Institute of Technology (MIT), Cambridge. In 1998, he was with HP Research Labs, Palo Alto, and in 2001 with Cadence Berkeley Labs. His research interests include development of integral equation solvers for very large complex systems, stochastic field solvers for large number of uncertainties, and automatic generation of parameterized stable compact models for linear and nonlinear dynamical systems. Dr. Daniel received the 1999 IEEE TRANSACTIONS ON POWER ELECTRONICS best paper award, the 2003 ACM Outstanding Ph.D. Dissertation Award in Electronic Design Automation, four best paper awards in international conferences, the 2009 IBM Corporation Faculty Award, 2010 Early Career Award from the IEEE Council on Electronic Design Automation, and the 2014 IEEE TRANSACTION ON COMPUTER AIDED DESIGN best paper award.