Optimal passivity enforcement of state-space models via localization methods

Original

Availability:
This version is available at: 11583/2606207 since:

Published
IEEE

Published
DOI:10.1109/EPEPS.2014.7103602

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)
Optimal Passivity Enforcement of State-Space Models via Localization Methods

Zohaib Mahmood*, Stefano Grivet-Talocia†, Giuseppe C. Calafiore‡, and Luca Daniel*

*Dept. Electrical Engineering and Computer Science, MIT, Cambridge, MA, 02139 USA e-mail: {zohaib,luca}@mit.edu
†Dept. Electronics and Telecommunications, Politecnico di Torino, Torino 10129, Italy e-mail: stefano.grivet@polito.it
‡Dept. Control and Computer Engineering, Politecnico di Torino, Torino 10129, Italy e-mail: giuseppe.calafiore@polito.it

Abstract—This work presents a novel approach for passivity enforcement of state-space macromodels, based on nonsmooth localization algorithms applied to a convex formulation of the passivity constraints. The main advantages of proposed scheme are guaranteed optimality and limited required computational resources. Compared to convex formulations based on a direct implementation of Bounded Real Lemma we are able to reduce both the memory and time requirements by orders of magnitude.

I. INTRODUCTION

Macromodeling techniques have become ubiquitous in modern Computer Aided Design (CAD) flows. Such methods allow the identification of compact dynamical models of complex interconnect structures. These models cast in various different forms [3], involve a coupling between all transfer matrix elements, or equivalently all individual elements of the state-space matrices. Therefore, the numerical complexity of passivity enforcement schemes is significantly larger than the complexity of state of the art rational fitting schemes. Moreover, since passivity is more restrictive than stability, there is a potential loss of accuracy that needs to be carefully controlled. These issues triggered significant research efforts in the last two decades.

We can roughly divide passivity enforcement methods in two classes: optimal and suboptimal schemes. Optimal methods provide, within a given parameterization form, a single passive model that is closest in some norm to the original data. In this class we include methods based on Positive or Bounded Real Lemma passivity constraints. Methods such as [5], [6] optimize poles and residues simultaneously in a single step, where as methods such as [7]–[9] optimize the poles and residues separately in two steps. The optimality of these methods, which is guaranteed by their convex formulation, comes at the price of possibly large computational requirements, both in terms of runtime and memory.

Suboptimal passivity enforcement schemes are instead based on approximate, linearized, or local forms of passivity constraints. In this class we include methods based on Hamiltonian perturbation and residue or state-space perturbation via localized constraints at specific frequencies [10]–[14]. These methods are much faster, but only approximate. They are not able to guarantee the best possible accuracy in the final passive models, and they require generally iterations, posing the problem of convergence due to lack of convexity.

In this paper, we present an algorithm that tries to combine the advantages of both approaches. We apply a localization scheme, namely the ellipsoid algorithm [15], to a convex nonsmooth formulation of the passivity constraints [9]. Due to convexity, we are able to prove that the optimal passive model will be obtained. Furthermore, the implementation of ellipsoid algorithm requires a reduced memory footprint, and the number of iterations (which will always converge to the optimum) is shown to be moderate. Therefore, the proposed approach is competitive with respect to both optimal methods (it produces the same result), and suboptimal methods (in terms of efficiency). These features are confirmed through application to various examples.

II. BACKGROUND

A. Passivity Enforcement via State Space Perturbation

A linear dynamical state space model for a system with \( p \) ports is given by

\[
\dot{x}(t) = Ax(t) + Bu(t), \\
y(t) = Cx(t) + Du(t).
\]

Here \( x(t) \in \mathbb{R}^n \) describes the state variables, \( u(t), y(t) \in \mathbb{R}^p \) describe the input and output vectors respectively. Matrices
A ∈ \mathbb{R}^{n,n}, B ∈ \mathbb{R}^{n,p}, C ∈ \mathbb{R}^{p,n} and D ∈ \mathbb{R}^{p,p} are obtained via rational fitting [1]. The transfer matrix corresponding to the state-space system (1) is given by

\[ H(s) = C(sI - A)^{-1}B + D. \tag{2} \]

System (1) and (2) are assumed to be stable. In scattering representation a stable system (2) is passive (or dissipative) if and only if its subgradient point. The algorithm defines a hyperplane orthogonal to a subgradient \( g ∈ \partial h \) of the constraint function. The half-space

\[ S_C = \{ z \mid g^T (z - δ_C^{(v)}) \leq 0 \} \tag{7} \]

is the one where the constraint function decreases and hence intersects with the feasible region. The algorithm constructs an ellipsoid \( χ_{ν+1} \) that includes \( χ_ν \cap S_C \), and such that \( \text{vol}(χ_{ν+1}) < \text{vol}(χ_ν) \), where \( \text{vol}(\cdot) \) denotes the volume of the argument.

Suppose instead that the center \( δ_C^{(ν)} \) of the ellipsoid \( χ_ν \) is feasible. The algorithm then defines \( g ∈ \nabla f \) as the gradient of the objective function, which decreases in the half-space (7), which in turn contains the global minimum \( δ_C \). The next ellipsoid \( χ_{ν+1} \) is constructed as in the previous case. This process is continued until the size of the updated ellipsoid is small enough such that all of its interior points fall within an \( \varepsilon \)-neighborhood of the global minimum \( δ_C \). Note that localization methods are not descent algorithms. Hence, they keep track of the best feasible solution attained through all iterations. The update steps of ellipsoid algorithm, given in [15], [18], are analytical, involving only matrix vector products. The ellipsoid algorithm is efficient in terms of memory, but it may take several iterations to converge.

III. PASSIVITY ENFORCEMENT VIA LOCALIZATION METHODS

In this paper, we employ the ellipsoid algorithm of Sec. II-B to solve the convex non-smooth problem (6).

A. Initial Set

One of the main challenges in using the ellipsoid algorithm is to define an initial set \( χ_0 \) that is guaranteed to contain the global minimum. This initial set needs to be as small as possible, because for larger initial sets the algorithm may take more iterations to converge. We define \( χ_0 \) in the form of a hypersphere with radius \( R \). We also compute an upper and a lower bound on \( R \), which help us to pick a value of \( R \) that is appropriate (see [19] for a detailed derivation).

1) An Upper Bound on \( R (R_{UB}) \): Since we have defined the objective function in terms of an \( ℓ^2 \) norm of the decision variable, the \( ℓ^2 \) norm of any feasible point \( δ_C^{(ν)} \) will define an upper bound on \( R \). Hence, a hypersphere centered at origin with radius equal to the Euclidean distance of \( δ_C^{(ν)} \) from the origin is guaranteed to contain the global minimum. As described in Section II, one of such feasible points is \( δ_C^{(ν)} = -\text{vec}(C) \), therefore we set \( R_{UB} = ||\text{vec}(C)|| \).

2) Lower Bound on \( R (R_{LB}) \): We assume that the initial unperturbed system is not passive, hence \( δ_C^{(0)} \) is infeasible. We define \( R_{LB} \) to be the radius of an infeasible hypersphere \( ε_{LB} \) centered at the initial point \( δ_C^{(0)} \), such that

\[ h(δ_C) > 1 \quad ∀δ_C ∈ ε_{LB}. \tag{8} \]
Since $h(\delta_C)$ is convex, we have $\forall \delta_C \in \epsilon_{LB}$

$$h(\delta_C) \geq h(\delta_C^{(0)}) + \partial h(\delta_C^{(0)})^T (\delta_C - \delta_C^{(0)})$$

$$= h(\delta_C^{(0)}) - R_{LB} \sqrt{\partial h(\delta_C^{(0)})^T \partial h(\delta_C^{(0)})},$$

(9)

where $\partial h$ denotes a subgradient [9] of the constraint function. From (9), we note that all points in $\epsilon_{LB}$ are infeasible if

$$h(\delta_C) \geq h(\delta_C^{(0)}) - R_{LB} \sqrt{\partial h(\delta_C^{(0)})^T \partial h(\delta_C^{(0)})} > 1.$$  

(10)

Solving (10) gives us $R_{LB}$ as

$$R_{LB} = \frac{h(\delta_C^{(0)}) - 1}{\sqrt{\partial h(\delta_C^{(0)})^T \partial h(\delta_C^{(0)})}} = \frac{h(\delta_C^{(0)}) - 1}{||\partial h(\delta_C^{(0)})||_2},$$

(11)

so that all the interior points of any hypersphere with radius $R < R_{LB}$ are infeasible. Hence, we must select $R > R_{LB}$ in order to guarantee that the hypersphere with radius $R$ includes the global minimum.

3) Practical Considerations: From earlier discussions, the value of $R$ must satisfy the following inequality

$$R_{LB} < R \leq R_{UB}.$$  

(12)

In our implementation we select $R = R_{UB}$.

B. Computational Complexity

The main attractive feature of the ellipsoid algorithm is its low memory usage. The ellipsoid algorithm has two major components: a) computing the $\mathcal{H}_\infty$ norm, and b) updating and storing ellipsoid parameters. The corresponding computing requirements are summarized in Table I. The algorithm requires only a modest storage of $O(q^2)$, where $q$ denotes the degrees of freedom ($\delta_C \in \mathbb{R}^q$). Notice that the cost per iteration for the ellipsoid algorithms is dominated by the computation of the $\mathcal{H}_\infty$ norm, which is $O(n^3)$.

<table>
<thead>
<tr>
<th>Component</th>
<th>Cost per Iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) $\mathcal{H}_\infty$ norm</td>
<td>$O(n^3)$</td>
</tr>
<tr>
<td>(b) Ellipsoid Parameters</td>
<td>$O(q^2)$</td>
</tr>
</tbody>
</table>

TABLE I

COST PER ITERATION

Fig. 1. Packaging interconnect: singular values of original non-passive (solid blue) and proposed passive (dashed red) models.

Fig. 2. Packaging interconnect: original non-passive and perturbed passive model responses ($S_{8,13}$)

IV. RESULTS

We consider two simple but challenging test cases. Given an initial nonpassive model identified by vector fitting [1], passivity was enforced using the proposed localization approach, an existing optimal passivity enforcement technique [7], and a suboptimal technique [10].

A. A packaging interconnect

The first example is a $p = 16$ port coupled packaging interconnect, for which an original non-passive model ($n = 598$ states) exhibits very large passivity violations over a large bandwidth ($\sigma_1 = 72$) as shown in Fig. 1. The bounds computed on the initial hypersphere’s radius are $0.52 < R \leq 1.97$. We selected $R = 2.0$. The ellipsoid algorithm took less than 500 iterations to find an accurate feasible solution, with a total runtime of about 50 minutes. However, note that over 80% (4.5 seconds per calculation) of the time was spent in computing the $\mathcal{H}_\infty$ norm. Here, we have used a standard algorithm to compute the $\mathcal{H}_\infty$ norm [16]. Memory usage was less than 90 MB.

Figure 2 demonstrates the accuracy of our passive model even in the presence of large passivity violations. Compared with the suboptimal technique [10], the proposed scheme is able to achieve passivity with a smaller perturbation amount, hence increased accuracy. Application of the alternative optimal technique [7] resulted in a problem with 217365 equations and 772865 variables. The required memory, as reported by Matlab was over 40 GB, and the problem could not be set up because the solver ran out of memory.

B. A SAW Filter

For this 3-port SAW filter, an initial stable but non passive model with 144 states was computed using vector fitting [1] from measured data. The original non-passive model exhibited large passivity violations at DC ($\sigma_1 = 25.83$) and around a normalized frequency of 0.156 ($\sigma_1 = 6.51$), as shown in Fig. 3. The bounds computed on the initial hypersphere’s radius are $0.06 < R \leq 1.0$. We selected $R = 1.0$. The ellipsoid algorithm converged in less than 300 iterations to an accurate feasible solution in 30 seconds. However, note that over 82% of the time was spent in computing the $\mathcal{H}_\infty$ norm. Memory usage was less than 1 MB.
ACKNOWLEDGMENT

This work was supported in part through the NCN-NEEDS program, which is funded by the National Science Foundation, contract 1227020-EEC, and the Semiconductor Research Corporation, TASK 2436.001; and in part through the MITOR MIT-Torino program and the MIT-SkolTech program.

REFERENCES