Algorithms for Passive Dynamical Modeling and Passive Circuit Realizations

by

Zohaib Mahmood

B.Sc., University of Engineering and Technology, Lahore, Pakistan (2007)
S.M., Massachusetts Institute of Technology (2010)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2015

© Massachusetts Institute of Technology 2015. All rights reserved.

Author

Department of Electrical Engineering and Computer Science
January 08, 2015

Certified by

Luca Daniel
Emanuel E. Landsman Associate Professor of Electrical Engineering
Thesis Supervisor

Accepted by

Leslie A. Kolodziejski
Chairman, Department Committee on Graduate Theses
Algorithms for Passive Dynamical Modeling and Passive Circuit Realizations

by

Zohaib Mahmood

Submitted to the Department of Electrical Engineering and Computer Science on January 08, 2015, in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

The design of modern electronic systems is based on extensive numerical simulations, aimed at predicting the overall system performance and compliance since early design stages. Such simulations rely on accurate dynamical models. Linear passive components are described by their frequency response in the form of admittance, impedance or scattering parameters which are obtained by physical measurements or electromagnetic field simulations. Numerical dynamical models for these components are constructed by a fitting to frequency response samples. In order to guarantee stable system level simulations, the dynamical models of the passive components need to preserve the passivity property (or inability to generate power), in addition to being causal and stable. A direct formulation results into a non-convex nonlinear optimization problem which is difficult to solve.

In this thesis, we propose multiple algorithms that fit linear passive multiport dynamical models to given frequency response samples. The algorithms are based on convex relaxations of the original non-convex problem. The proposed techniques improve accuracy and computational complexity compared to the existing approaches. Compared to sub-optimal schemes based on singular value or Hamiltonian eigenvalue perturbation, we are able to guarantee convergence to the optimal solution within the given relaxation. Compared to convex formulations based on direct Bounded-Real (or Positive-Real) Lemma constraints, we are able to reduce both memory and time requirements by orders of magnitude. We show how these models can be extended to include geometrical and design parameters.

We have applied our passive modeling algorithms and developed new strategies to realize passive multiport circuits to decouple multichannel radio frequency (RF) arrays, specifically for magnetic resonance imaging (MRI) applications. In a coupled parallel transmit array, because of the coupling, the power delivered to a channel is partially distributed to other channels and is dissipated in the circulators. This dissipated power causes a significant reduction in the power efficiency of the overall system. In this work, we propose an automated eigen-decomposition based approach to designing a passive decoupling matrix interfaced between the RF amplifiers and the
coils. The decoupling matrix, implemented via hybrid couplers and reactive elements, is optimized to ensure that all forward power is delivered to the load. The results show that our decoupling matrix achieves nearly ideal decoupling. The methods presented in this work scale to any arbitrary number of channels and can be readily applied to other coupled systems such as antenna arrays.

Thesis Supervisor: Luca Daniel
Title: Emanuel E. Landsman Associate Professor of Electrical Engineering
Acknowledgments

I would like to thank Professor Luca Daniel for his constant support and guidance over the last several years. I am grateful to him for all the creative discussions we have had which inspired several new ideas presented in this research. It would not have been possible for me to complete this research without his advice and constant mentoring.

I am grateful to Professor Jacob White and Professor Elfar Adalsteinsson for being on my PhD Committee. In addition to my thesis supervisor and committee members I would like to thank my mentors including Professor Lawrence L Wald, Professor Roberto Suaya, Professor Stefano Grivet-Talica and Professor Giuseppe Carlo Calafiore. I am obliged to all of them for their time and feedback on my research.

I am greatly indebted to Brad Bond for getting me started with this new field. Our first interaction started when he was a Teaching Assistant for the course 6.336 Introduction to Numerical Simulation. I never missed his office hours. I am thankful to Prof. Luca and Dr. Brad for making 6.336 such an exciting experience. Later we continued to collaborate very closely on different topics including nonlinear system identification and some of the research presented in this thesis. I am grateful to him for always welcoming technical discussions. He has been a great source of learning and inspiration for me.

Working with the Computational Prototyping Group has been a great pleasure. I would like to thank all current and past members of our group for very useful discussions.

I would like to thank the MIT Office of the Provost for awarding me the Irwin Mark Jacobs and Joan Klein Jacobs Presidential Fellowship, which supported my research during my first year at MIT. I am grateful to all the funding sources that funded various projects during my PhD.

Last, but definitely not the least, I am wholeheartedly thankful to my family for their love, care and encouragement throughout my life. Without their support, help
and confidence in me, this journey could not even have started.
Contents

1 Introduction ......................................................... 13
   1.1 Motivation .................................................. 13
   1.2 Overview and Contributions of this Thesis .................... 15
   1.3 Organization of this Thesis .................................. 16

2 Background ......................................................... 17
   2.0.1 Notations .................................................. 17
   2.1 Linear Time Invariant (LTI) systems ......................... 17
   2.2 Passivity ................................................... 18
      2.2.1 Manifestation of Passivity for a Simple RLC Network .. 20
   2.3 Tests for Certifying Passivity ................................ 20
      2.3.1 Tests Based on Solving a Feasibility Problem ............ 20
      2.3.2 Tests Based on Hamiltonian Matrix ..................... 21
      2.3.3 Sampling Based Tests - Only Necessary ................. 22
   2.4 Convex Optimization Problems ................................ 22
      2.4.1 Convex and Non-Convex Functions ....................... 22
      2.4.2 Convex Optimization Problems ......................... 23
      2.4.3 Semidefinite Programs ................................ 24

3 Existing Techniques .............................................. 25
   3.1 Traditional Approaches ...................................... 25
   3.2 Automated Approaches ...................................... 27
   3.3 Projection Based Approaches ................................ 27
3.3.1 The Traditional Projection Framework .................................. 27
3.3.2 Stable Projection for Linear Systems ................................. 28
3.3.3 Parameterization of Projection Methods .............................. 29
3.4 Rational Fitting of Transfer Functions ................................. 30
  3.4.1 One-step Procedures .................................................. 32
  3.4.2 Two-steps Procedures ................................................ 32
  3.4.3 Parameterized Rational Fitting ....................................... 34

4 Passive Fitting for Multiport Systems - Method I ........................ 37
  4.1 Rational Transfer Matrix Fitting in Pole Residue Form ............. 38
  4.2 Passive Fitting for Multiport LTI Systems ............................ 39
    4.2.1 Problem Formulation ................................................. 39
    4.2.2 Conjugate Symmetry ............................................... 39
    4.2.3 Stability ............................................................. 40
    4.2.4 Positivity ........................................................... 41
    4.2.5 The Constrained Minimization Problem ............................ 43
  4.3 Implementation ........................................................... 44
    4.3.1 Step 1: Identification of Stable Poles ........................... 44
    4.3.2 Step 2: Identification of Residue Matrices ...................... 45
    4.3.3 Equivalent Circuit Synthesis ..................................... 48
    4.3.4 The Complete Algorithm .......................................... 49
  4.4 Proposed Unconstrained Minimization Setup .......................... 50
    4.4.1 Notations ............................................................ 50
    4.4.2 Computing The Equivalent Function ............................... 51
    4.4.3 Computing The Jacobian Matrix ................................... 53
  4.5 Results ........................................................................... 55
    4.5.1 Wilkinson Combiner in a LINC Amplifier .......................... 55
    4.5.2 Power & Ground Distribution Grid .................................. 58
    4.5.3 On-Chip RF Inductors ............................................... 60
    4.5.4 1-Port Structure ...................................................... 62
5 Passive Fitting for Multiport Systems - Method II

5.1 Background

5.1.1 Problem Description

5.1.2 Computation of the $H_\infty$ Norm

5.1.3 Localization Methods

5.2 Passivity Enforcement using Localization Methods

5.2.1 Initialization

5.2.2 Passivity Enforcement Using The Ellipsoid Algorithm

5.2.3 Passivity Enforcement Using The Cutting Plane Method

5.2.4 Piecewise Linear Lower Bound on the Global Optimum

5.3 Results

5.3.1 Example 1: A 4-Port PCB Interconnect Structure

5.3.2 Example 2: A 4-Port Transformer

5.3.3 Example 3: Handling Large Passivity Violations

5.3.4 Example 4: Wide Bandwidth Passivity Violation

5.3.5 Example 5: A packaging interconnect

5.3.6 Example 6: A SAW Filter

5.3.7 General Discussion

6 Globally Passive Parameterized Model Identification

6.1 Motivation

6.2 Background

6.2.1 Interpolatory Transfer Matrix Parameterization

6.2.2 Positivity of Functions

6.3 Optimal Parameterized Fitting

6.3.1 Problem Formulation
Chapter 1

Introduction

1.1 Motivation

Generation of accurate and passive dynamical models for linear multiport analog circuit blocks is a crucial part of the design and optimization process for complex integrated circuit systems. Quite often, these models also need to capture dependence of the system on design and geometrical parameters while providing apriori passivity and stability certificates for the entire parameter range of interest. These identified models are interfaced with commercial circuit simulators where they are used to perform transient simulations being interconnected with other circuit blocks. In order to guarantee robust and stable numerical simulations, these models must satisfy physical properties such as causality, stability and passivity [119,126]. Causality and stability can be easily enforced during the model fitting [33,34,59,63]. Passivity enforcement is instead more challenging and requires special care. In this thesis we focus on generating guaranteed passive linear dynamical models.

Let us consider the typical design flow for an analog circuit block, say a distributed power amplifier, containing on-chip multiport passive interconnect structures, such as multi-primary transformers for power combining. The full system design is completed in two steps. As the first step, the interconnect passive structures are laid out in an electromagnetic field solver and then simulated for frequency response in the desired frequency band. The second step consists of developing a reduced model
based on the frequency samples or system matrices extracted by the solver which can be incorporated into a circuit simulator (e.g. Spice or Spectre).

![Equivalent Multiport Model](image)

Figure 1-1: The model of a multiprimary transformer interfaced with circuit simulator to perform full distributed power amplifier simulations

Once the models are generated, they are interfaced with the circuit simulators using either equivalent netlists or behavioral descriptions. The final network in Figure 1-1 shows the interconnection of the generated model with other building blocks of the amplifier such as transistors and capacitors. Transient simulations are performed in order to get performance metrics of the complete nonlinear power amplifier. If the generated model encounters a violation of any basic physical property of the structure, such as passivity, it can cause huge errors in the response of the overall system, and the results may become completely nonphysical.

In addition, the designers would greatly benefit from these models if they are parameterized in some of the design parameters, such as width and spacing for an inductor or characteristic impedance in the case of distributed transmission line structures. These parameterized models will greatly reduce the design cycle by allowing the designer to instantiate the structure with various design parameters without performing another full-wave electromagnetic simulation. However these models are useful only if the final parameterized models come with apriori passivity certificates in the entire parameter range of interest.

These models can also be used to realize passive networks. Although this is a well studied problem for single and two port networks. The problem of synthesizing a network from its given frequency response is highly challenging specially for multiport
devices. In this thesis we propose strategies to automatically design realizable networks from the given response. We have explored applications including the design and practical realization of decoupling matrices (or multiport matching networks) to decouple Radio Frequency (RF) phased arrays for Magnetic Resonant Imaging (MRI) applications.

1.2 Overview and Contributions of this Thesis

In this thesis we propose various algorithms to automatically generate both individual non-parameterized and parameterized passive multiport models, and passive circuit realizations. In particular the main contributions of this thesis are summarized as follows:

- We have proposed an algorithm to identify individual passive dynamical models. The proposed algorithm identifies the unknown system in two steps. The first step is to identify a common set of stable poles for the multiport structure. The second step is to identify residue matrices which conform to passivity conditions. These passivity conditions are enforced as Linear Matrix Inequalities (LMIs) [82, 90].

- We have proposed another passive model identification algorithm which 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 $H_{\infty}$ norm of the system transfer function. We solve the resulting convex non-smooth optimization problem via computationally efficient localization based algorithms, such as the ellipsoid and the cutting plane methods [85,86,89].

- We have proposed a new strategy to develop globally stable and passive parameterized models. In this approach a collection of individual passive models is approximated by a single closed form model which conforms to passivity conditions in a continuous parameter range of interest [88].
We have extended our passive modeling algorithms and proposed automated strategies to design a robust decoupling matrix (or multiport matching network) interfaced between the RF amplifiers and the coils in a multi channel transmission array. Our decoupling matrix mixes the input signals and is optimized to ensure all forward power is delivered to the load [83,84].

The matlab implementations of various algorithms proposed in this thesis is posted on public domain as free open source software at http://www.rle.mit.edu/cpg/ and [1].

1.3 Organization of this Thesis

This thesis is organized as follows: Chapter 2 covers the relevant background on linear systems and passivity. Chapter 3 describes a brief overview of the existing techniques along with their advantages and shortcomings. Chapter 4 explains our convex formulation identifying passive system models in pole residue form. Chapter 5 presents our algorithm identifying passive linear system models using localization methods. Chapter 6 presents details on how the proposed algorithm can be extended to identify globally passive parameterized models with apriori passivity certificates. Chapter 7 describes how our passive modeling algorithms can be extended to design decoupling matrices for coupled arrays. Chapter 8 summarizes the thesis. Appendix A describes how some of the convex problems can be cast as semidefinite programs. All the theoretical developments are supported by examples which are provided at the end of relevant chapters.
Chapter 2

Background

2.0.1 Notations

We shall use the following notations in this section.

\[ n_p = \text{Number of ports of the given system} \quad (2.1) \]
\[ \kappa = \text{Number of poles} \quad (2.2) \]
\[ q = \text{Number of states of the compact model} \quad (2.3) \]
\[ F = \text{Number of frequency points} \quad (2.4) \]

2.1 Linear Time Invariant (LTI) systems

Dynamical systems are an extremely useful tool for the time-domain analysis of physical systems, as they provide a relationship between an input signal \( u(t) \) and an output signal \( y(t) \) for the system. In state-space models, the evolution of the system is described by a state vector \( x(t) \), which is controlled by input \( u(t) \) and from which output \( y(t) \) is determined. In the general form, a Linear Time Invariant (LTI) state-space model can be expressed as
\[
\frac{d(x)}{dt} = Ax(t) + Bu(t) \tag{2.5}
\]

\[
y(t) = Cx(t) + Du(t)
\]

where \( A \in \mathbb{R}^{q \times q} \), \( B \in \mathbb{R}^{q \times n_p} \), \( C \in \mathbb{R}^{n_p \times q} \), and \( D \in \mathbb{R}^{n_p \times n_p} \). Here \( q \) and \( n_p \) are the number of states and ports respectively. Linear systems can also be expressed in terms of a transfer matrix, by applying Laplace transform of (2.5), as

\[
H(s) = \frac{Y(s)}{U(s)} \triangleq C(sI - A)^{-1}B + D = \begin{bmatrix}
H_{11}(s) & \cdots & H_{1p}(s) \\
\vdots & \ddots & \vdots \\
H_{p1}(s) & \cdots & H_{pp}(s)
\end{bmatrix} \tag{2.6}
\]

Here \( H_{ij}(s) \) indicates the transfer function from Port \( j \) to Port \( i \).

### 2.2 Passivity

Passivity is one of the most important properties of dynamical systems. 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 [119], 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 [51].

Consider a system described by input (say current) \( u \) and output (say voltage) \( y \). Then \( \langle u, y \rangle_\tau \) describes the total energy of the system up to time \( \tau \), where

\[
\langle u, y \rangle_\tau = \int_{-\infty}^\tau y(t)^T u(t) \, dt. \tag{2.7}
\]
Then the system is passive if

$$\langle u, y \rangle_{\tau} \geq 0, \forall \tau \in R^+ \quad (2.8)$$

Linear systems are usually described as a transfer matrix or as a state space model. Let us consider the passivity of impedance or admittance systems specified by a transfer matrix $H(s)$. Passivity for an impedance or admittance system corresponds to ‘positive realness’ of the transfer matrix. To be *positive real*, the transfer matrix $\hat{H}(s)$ must satisfy the following constraints

$$\overline{H(s)} = H(s) \quad (2.9a)$$

$H(s)$ is analytic in $\Re s > 0 \quad (2.9b)$

$$H(j\omega) + H(j\omega)^\dagger \succeq 0 \ \forall \omega \quad (2.9c)$$

Where $\Re$ denotes the real part and $\dagger$ indicates the conjugate transpose.

The first condition (2.9a), commonly known as *conjugate symmetry*, ensures that the impulse response corresponding to $H(s)$ is real. The second condition (2.9b) 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 system is marginally stable if it has simple poles (i.e. poles with multiplicity one) on the imaginary axis. The third and final condition (2.9c), which is the *positivity condition*, implies positive realness of the symmetric part of the transfer matrix on the $j\omega$ axis.
2.2.1 Manifestation of Passivity for a Simple RLC Network

We consider a simple RLC network as shown in Figure 2-1. In this simple schematic we can analytically compute the equivalent of passivity conditions as follows

\[ Z_{eq}(\omega) = R + jX_{eq}(\omega) \]
\[ = R + j\frac{\omega L}{1 - \omega^2 LC} \]  

(2.10)

Here $\Re Z_{eq} = R$. The passivity condition translates into $R$ being non-negative, i.e. $R \geq 0$ in addition to $L$ and $C$ being non-negative.

Figure 2-1: Manifestation of passivity for a simple RLC network where $Z_{eq}(\omega) = R + jX_{eq}(\omega)$. Passivity implies that $R, L, C \geq 0$

2.3 Tests for Certifying Passivity

There are several tests, all based on positive real lemma, by which a model can be certified to be passive. Section 2.3.1 and 2.3.2 describe conditions which are both necessary and sufficient for passivity [16]. Section 2.3.3 describes a necessary condition for passivity.

2.3.1 Tests Based on Solving a Feasibility Problem

When the system is represented in general state space form,
\[
\frac{Ed(x)}{dt} = Ax + Bu \\
y = Cx + Du
\]

with minimal realization (i.e. every eigenvalue of A is a pole of \(H(s)\)), passivity is implied by the positive real lemma. The positive real lemma states that if there exists a positive definite matrix \(P = P^T > 0\) and \(P \in \mathbb{R}^{q \times q}\) such that the following matrix is negative semidefinite

\[
\begin{bmatrix}
E^T PA + A^T PE & E^T PB - C^T \\
B^T PE - C & -D - D^T
\end{bmatrix} \preceq 0
\]  (2.11)

then \(H(s)\) is positive real and hence the system is passive. In order to certify if a system is passive, the feasibility problem for the existence of a positive definite matrix \(P = P^T > 0\) can be solved. However such a formulation will result into a non-linear non-convex problem which may not be solved efficiently.

### 2.3.2 Tests Based on Hamiltonian Matrix

We can solve a condition equivalent to (2.11) based on Riccati equations and Hamiltonian Matrices. If we assume that \(D + D^T > 0\), then the inequality (2.11) is feasible if and only if there exists a real matrix \(P = P^T > 0\) satisfying the Algebraic Riccati Equation

\[
A^T P + PA + (PB - C^T)(D + D^T)^{-1}(PB - C^T)^T = 0
\]  (2.12)

In order to solve the Algebraic Riccati Equation 2.12 we first form the associated
Hamiltonian matrix $M$ as follows

$$
\begin{bmatrix}
A - B(D + D^T)^{-1}C & B(D + D^T)^{-1}B^T \\
-C^T(D + D^T)^{-1}C & -A^T + C^T(D + D^T)^{-1}B^T
\end{bmatrix}.
$$

(2.13)

Then the system (2.5) is passive, or equivalently, the LMI (2.11) is feasible, if and only if $M$ has no pure imaginary eigenvalues [16].

2.3.3 Sampling Based Tests - Only Necessary

Another class of passivity tests are based on checking the passivity conditions (2.9) at discrete frequency samples. Since passivity requires condition (2.9) to hold for all $\omega$, a reduced model is non-passive if $\lambda_{\min}(\Re\{H(j\omega_i)\}) < 0$ only for some $\omega_i$. Here $\lambda$ denotes the eigen values. Note that this test is based on a necessary condition and can only be used to check data samples or samples from the model for passivity violations and cannot be used to certify the passivity of a model.

2.4 Convex Optimization Problems

In this section we describe general convex optimization problems. Before discussing the optimization problems, we first describe the notion of convexity and convex functions in the following section. For detailed description we refer the readers to [19].

2.4.1 Convex and Non-Convex Functions

A function $f(x)$ is convex if the domain of $f(x)$ is convex and if for all $x, y \in \text{domain } f(x)$ and $\theta \in [0, 1]$, it satisfies

$$
f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y)
$$

(2.14)

Geometrically, this inequality means that the line segment between $(x, f(x))$ and $(y, f(y))$ lies above the graph of $f(x)$ as shown in Figure 2-2.
Figure 2-2: Shows a convex function. In general finding global minimum for convex functions is easy.

One of the nice properties of a convex function is that they have only global minimum which is relatively easier to compute compared to non-convex functions, as shown in Figure 2-3, which may have local minima and hence making the computation of global minimum an extremely difficult task.

Figure 2-3: Shows a convex function. In general finding global minimum for convex functions is extremely difficult.

2.4.2 Convex Optimization Problems

Using the notation of [19], a convex optimization problem is of the form
minimize $f_0(x)$
subject to $f_i(x) \leq 0, \ i = 1, \ldots, m$ \hspace{1cm} (2.15)

$a_i^Tx = b_i, \ i = 1, \ldots, p$

where the objective function $f_0(x)$ is convex, the inequality constraint functions $f_i(x)$ are convex and the equality constraint functions $h_i(x) = a_i^Tx - b_i$ are affine. Hence in a convex optimization problem we minimize a convex objective or cost function over a convex set. Convex optimization problems are particularly attractive since finding global minimum for the cost function is a relatively easy.

2.4.3 Semidefinite Programs

Semidefinite Programs or simply SDPs belong to a special type of convex optimization problems where a linear cost function is minimized subject to linear matrix inequalities.

minimize $c^Tx$
subject to $F_1x_1 + F_2x_2 + \cdots + F_nx_n - F_0 \succeq 0$ \hspace{1cm} (2.16)

where all of the matrices $F_0, F_1, \ldots F_n \in S^k$, here $S^k$ indicates set of symmetric matrices of order $k \times k$. Semidefinite Programs are particularly important since most of the convex optimization problems can be cast as an SDP (described in Appendix A) and can be solved efficiently using public domain solvers such as [2,100].
Chapter 3

Existing Techniques

System level 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 dynamical models that prove efficient when used in standard circuit simulators. Over the last few years many techniques [15,22,23,28,31,33,34,36,46,50,52,53,57,59–64,68,74,79,80,86,87, 90,102–104,107,108,112,119,120,126] have been developed for the generation of linear dynamical models. In this chapter we summarize various existing techniques for non-parameterized and parameterized model generation of multiport passive structures.

3.1 Traditional Approaches

Traditionally, the critical task of generating a model is completed manually that is the circuit designer or system architect approximates the unknown system with empirical or semi empirical formulas. These models rely on the designers’ experience and intuition accumulated after a lifetime of simulations with electromagnetic field solvers and circuit simulators. In these approaches, normally the designer would either approximate the structure with lumped RC and RL networks characterized at the operating frequency as shown in Figure 3-1 or generate a simple schematic from intuition consisting of RLC elements having frequency response ‘close’ to the original structure as shown in Figure 3-1.
Unfortunately, these intuitive approaches have limited applications and may not capture all the important features in a given frequency response. Additionally, in order to generate a complete multiport transfer matrix, quite often these approaches approximate the individual transfer functions separately. Hence, these models may be completely nonphysical, violating important physical properties of the original system such as passivity, since passivity is a property of the entire transfer matrix and cannot be enforced if transfer functions are identified individually. Also, these techniques are not scalable, hence generating a model for a system with more than a few ports may become challenging. Furthermore, the modeling task becomes even more difficult when attempting to generate by hand closed form compact models of the frequency response parameterized by design or geometrical parameters.

In order to make this process more efficient and robust, it is desirable to replace hand-generated macromodels by automatically-generated compact dynamical models that come with guarantees of accuracy, and passivity. In the following sections we discuss some of the commonly used techniques for the identification of compact
dynamical models.

3.2 Automated Approaches

In the recent years, considerable effort has been put into automating the procedure to generate compact parameterized models. There are two commonly used techniques to generate models for linear structures. The first ones are projection based approaches and the second ones are rational fitting approaches. A detailed survey of these approaches is presented in [14]. We describe these techniques one by one in the following sections.

3.3 Projection Based Approaches

3.3.1 The Traditional Projection Framework

Most of the model order reduction techniques can be interpreted within a projection framework. In such a framework the solution to a given large linear multiport system

\[ E \dot{x} = Ax + Bu, \quad y = C^T x, \]  

(3.1)

is approximated in a low-dimensional space \( x \approx V \hat{x} \), where \( V \in \mathbb{R}^{q \times q_r} \) is the right projection matrix, \( \hat{x} \) is the reduced state vector, and \( q \gg q_r \). A reduced set of equations is then obtained by forcing the residual, \( r(V \hat{x}) = EV \dot{\hat{x}} - AV \hat{x} - Bu \), to be orthogonal to the subspace defined by a left projection matrix \( U \), i.e. \( U^T r(V \hat{x}) = 0 \). The resulting state space model has the form

\[ \hat{E} \dot{\hat{x}} = \hat{A} \hat{x} + \hat{B} u, \quad y = \hat{C}^T \hat{x}, \]  

(3.2)

where \( \hat{E} = U^T EV, \hat{A} = U^T AV, \hat{B} = U^T B, \) and \( \hat{C} = V^T C \). The accuracy of
the reduced model created via projection is completely determined by the choice of projection matrices $U$ and $V$. The most common approaches for selecting the vectors are methods based on balanced truncation, moment matching, and singular value decomposition (e.g. proper orthogonal decomposition, principle components analysis, or Karhunen-Loeve expansion). For more details on generating projection vectors see [11,56].

### 3.3.2 Stable Projection for Linear Systems

Traditionally it is assumed that the original large system (3.1) possesses an extremely special structure viz. $E = E^T \succeq 0, A \preceq 0$, and $B = C$. In such cases selecting $U = V$ (known as congruence transform or Galerkin projection) will preserve stability and passivity in the reduced system for any choice of $V$. While all digital RLC type interconnect networks possess the required semidefinite structure, for analog modeling it is, unfortunately, completely unrealistic to restrict consideration to only semidefinite systems. Therefore for the vast majority of analog systems, the congruence transform cannot guarantee stability and passivity of the generated model. One possible computationally cheap solution is to use as a first step any of the available traditional projection based methods (including congruence transforms) and then attempt to perturb the generated model to enforce stability and passivity. One semidefinite formulation of this problem is

$$
\begin{align*}
\text{minimize} & \quad ||\Delta \hat{E}|| + ||\Delta \hat{A}|| + ||\Delta \hat{C}|| \\
\text{subject to} & \quad \hat{E} \succeq 0, \\
& \quad \hat{A} + \hat{A}^T \preceq 0, \\
& \quad \hat{B} = \hat{C}
\end{align*}
$$

(3.3)

where $\hat{E} = U^T E V + \Delta \hat{E}$, $\hat{A} = U^T A V + \Delta \hat{A}$, $\hat{B} = U^T B$, and $\hat{C} = V^T C + \Delta \hat{C}$. Here stability and passivity are enforced in the reduced model by forcing it to be described by semidefinite system matrices, which introduced no loss of generality even if the original system (3.1) is not described by semidefinite matrices [13].
Unfortunately, in most cases any such perturbation could completely destroy the accuracy of the reduced model. Instead of perturbing the reduced model, a better approach that can guarantee accuracy in the reduced model is to perturb one of the projection matrices. That is, given $U$ and $V$, search for a ‘small’ $\Delta U$ such that the system (3.2), defined by reduced matrices $\hat{E} = (U+\Delta U)^T EV$, $\hat{A} = (U+\Delta U)^T AV$, $\hat{B} = (U+\Delta U)^T B$, and $\hat{C} = V^T C + \Delta \hat{C}$ is passive. This problem can similarly be formulated as a semidefinite program

$$\begin{align*}
\text{minimize} & \quad ||\Delta U|| \\
\text{subject to} & \quad \hat{E} \succeq 0, \\
& \quad \hat{A} + \hat{A}^T \preceq 0, \\
& \quad \hat{B} = \hat{C} \quad (3.4)
\end{align*}$$

It can be shown that if the original model (3.1) is stable and passive, then for any projection matrix $V$ there exist projection matrices $U$ such that the resulting reduced model is stable and passive [13].

### 3.3.3 Parameterization of Projection Methods

Generating a parameterized reduced model, such as

$$\hat{E}(\lambda) \dot{x} = \hat{A}(\lambda) x + \hat{B}(\lambda) u \quad (3.5)$$

for a linear system where $\lambda$ is a vector of design parameters, is of critical importance if the models are to be used for system level design trade-off explorations. Two modifications to the previously described projection procedures must be made when constructing parameterized models. First the subspace defined by $V$ must capture the solution response to changes in parameter values. Expanding the subspace is typically achieved for linear systems by generating projection vectors that match the frequency response derivatives with respect to the parameters $\lambda$ in addition to the
frequency [29, 125]. Alternative approaches for handling variability resulting from a large number of parameters are based on sampling and statistical analysis [38, 133].

The second issue involves specifically the case of nonlinear parameter dependence, where the system matrix or vector field must be able to cheaply capture changes in $\lambda$. One way to make a parameterized system matrix $A(\lambda)$ projectable with respect to the parameters is to represent them as a sum on non-parameterized functions that are linear in scalar functions of the original parameters. For instance, for the parameterized linear system

$$\dot{x} = A(\lambda)x$$

we seek to approximate and project as follows:

$$A(\lambda) \approx \sum_{i=0}^{\kappa} \hat{A}_i g_i(\lambda) \rightarrow \hat{A}(\lambda) \approx \sum_{i=0}^{\kappa} (U^T A_i V) g_i(\lambda)$$

such that $\hat{A}_i = U^T A_i V$ are constant matrices and can be precomputed. Here $g_i(\lambda)$ are scalar functions of the original parameter set $\lambda$. The matrix approximation in (3.7) can be achieved using a polynomial expansion if $A(p)$ is known analytically, or via fitting in the case when only samples of the matrix $A_k = A(\lambda_k)$ are known [29].

### 3.4 Rational Fitting of Transfer Functions

Projection methods have been successful for certain classes of linear systems, but in many applications, such as when modeling analog passive components affected by full-wave effects or substrate effects, the resulting system matrices include delays, or frequency dependency. To capture such effects in a finite-order state-space model, one must approximate this frequency dependence, using for instance polynomial fitting of matrices [30], making preservation of passivity through projection even more
challenging. Furthermore, often times only transfer function samples are available, obtained possibly from measurements of a fabricated device or from a commercial electromagnetic field solver. In such a scenario, since original matrices are not available, projection based approaches cannot be used.

An alternative class of methods are based on transfer matrix fitting. There exist different approaches to generate rational transfer function matrices from frequency response data [34, 59, 68, 79, 107]. The problem of finding a passive multiport model from complex frequency response data is highly nonlinear and non convex. Given a set of frequency response samples \( \{ H_i, \omega_i \} \), where \( H_i = H(j\omega_i) \) are the transfer matrix samples of some unknown multiport linear system, the compact modeling task is to construct a low-order rational transfer matrix \( \hat{H}(s) \) such that \( \hat{H}(j\omega_i) \approx H_i \). Formulated as an \( L_2 \) minimization problem of the sum of squared errors, it can be written as

\[
\text{minimize} \sum_i \left| H_i - \hat{H}(j\omega_i) \right|^2
\]

subject to \( \hat{H}(j\omega) \) passive

(3.8)

Even after ignoring the passivity constraint in (3.8), the unconstrained minimization problem is non-convex and is therefore very difficult to solve. Direct solution using nonlinear least squares have been proposed, such as Levenberg-Marquardt [93]. However, there is no guarantee that such approach will converge to the global minimum, and quite often the algorithm will yield only a locally optimal result. Rather than solving non-convex minimization problem, many methods apply a relaxation to the objective function in (3.8) resulting in an optimization problem that can be solved efficiently. These methods 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. These schemes can be broadly clas-
sified into one-step and two-steps procedures. One-step procedures simultaneously enforce passivity during the fitting process by using a convex or quasi-convex relaxation [80, 112]. Two-steps procedures first generate a stable (but not necessarily passive) model. Passivity is enforced during the second step [23, 28, 31, 36, 87, 90].

3.4.1 One-step Procedures

Over the past years considerable effort has been put into finding a convex relaxation to the original problem including the passivity constraint (3.8) such as [80, 112]. These techniques lie on the more optimal but less efficient side of the trade-off. The technique in [112] (available for download at [4]) 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 [112] to the generation of parameterized passive models. Although optimal, this formulations may require possibly large computational costs both in terms of memory and CPU time for systems with a large number of parameters.

3.4.2 Two-steps Procedures

A suboptimal but potentially 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 [59]. 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.

Passivity via Perturbation

Many of the ‘two-steps’ methods employ a perturbation framework in the second step [15, 46, 50, 52, 57, 64, 102, 103, 108], where passivity violations are corrected iteratively. 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 ([60, 62, 102, 103]) through the solution of approximated local problems. Variants of the above schemes have been presented in [53, 61, 74, 104, 120]. A comprehensive comparison of such passive linear dynamical modeling techniques is available in [48]. In the following sections, we describe two approaches relevant to this thesis.

**Passivity Enforcement Using Linear Matrix Inequalities**

Within these ‘two-steps’ methods, some approaches guarantee optimality in the second step exploiting convex or quasi-convex formulations ([23, 28, 31, 36, 87, 90]). Such algorithms enforce passivity by defining constraints based on the positive real lemma or the bounded real lemma [18, 98].

For example, 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) [28]. Passivity enforcement via bounded real lemma can be directly formulated as a semidefinite program (SDP) and solved using standard SDP solvers such as [117] to compute the global optimal solution.

\[
\begin{align*}
\text{minimize} & \quad \sum_i \left| H_i - \hat{H}(j\omega_i) \right|^2 \\
\text{subject to} & \quad \begin{bmatrix} A^T W + W A & W B & C^T \\
B^T W & -I & D^T \\
C & D & -I \end{bmatrix} \preceq 0 \\
\quad & \quad W = W^T \succeq 0
\end{align*}
\]

Problem (3.10) generates a passive model with optimal accuracy with in the class of two-steps based methods (in these methods the poles and residues are calculated in two steps). The main drawback of these methods, however, is the excessive computational cost [28]), 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.

**Passivity via Passive-Subsections**

A model is passive if all of its building blocks are passive. There are approaches, such as [92,94], where the individual building blocks of a non-passive model are checked for passivity individually. However, since such a condition is only a sufficient condition, many passive models will fail the test. Also in these approaches [92, 94] no efficient method or algorithm was presented in order to rectify for passivity violations. For example in [94] 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 identification such that there are no passivity violation in the final model.

**3.4.3 Parameterized Rational Fitting**

There are two possible approaches to generating a parameterized transfer matrix $\hat{H}(s, \lambda)$ from a given set of frequency response and parameter values $\{H_i, \omega_i, \lambda_i\}$. The first approach is to fit simultaneously to the frequency response data and parameter values, i.e. minimizing $\sum |\hat{H}(j\omega_i, \lambda_i) - H_i|^2$. This approach was first proposed in [112] along with the simultaneously enforcement of stability passivity. However, simultaneous frequency and parameter fitting can become quite expensive for a large number of parameters. Alternative fitting approaches rely on interpolating between a collection of non-parameterized models, each generated by a stable and passive fitting procedure. The main challenge for such approaches based on interpolation is to guarantee passivity in the final interpolated model, since one may produce very trivial stable systems where simple interpolation will not preserve stability or passivity. All the current existing interpolation based algorithms [32,40,118] only provide a test to simply check stability after a particular instance of the parameterized model has been
instantiated. The downside of such methods is that the user (i.e. a circuit designer, or a system level optimization routine) would need to run such test every single time a new model is instantiated. Furthermore, if the instantiated model does not pass the stability test, the user would either be stuck with an unstable model, or would need to basically rerun the fitting algorithm to perturb the unstable instantiation until stability is achieved. In other words none of the available interpolation based approaches can guarantee that any instantiation of their identified parameterized models will be a priori stable and passive for any value of the parameters in a predefined range.

In this thesis we present a method for generating parameterized models of linear systems that the user will be able to instantiate for any parameter value either within a limited given range, or for an unlimited range, and be sure a priori to obtain a passive model. Given a collection of systems swept over design and geometrical parameters of interest, we identify a closed form parameterized dynamical model using constrained fitting. The details can be found in Chapter 6. Our algorithm is completely independent from the type of initial non-parameterized identification procedure used for the individual systems, if only stability is sought in the final parameterized model. In other words, the individual (non-parameterized) models may be generated by any stability preserving modeling scheme such as convex optimization based approaches [28, 81, 112], vector fitting based approaches [47, 49, 58, 59] or Loewner matrix based approaches [77]. However, in order to enforce global passivity, the individual non-parameterized models need to have the structure described in Chapter 4.
Chapter 4

Passive Fitting for Multiport Systems
- Method I

In this chapter we present a framework for identifying passive dynamical models from frequency response data. We cast the problem as a standard semidefinite program (SDP) which can be solved by SDP solvers such as [2,100]. We solve the problem in two steps. First a set of common poles is identified using already established techniques [5, 59, 112]. 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 4.5. 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 [28, 112]. We remark that although [28, 112] can generate more accurate models, their usability is limited to very small examples.

We use a framework similar to the one proposed in [82], 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. We have tested the algorithm on a set of challenging examples for which existing approaches do not perform well.
Note that sufficient only passivity conditions used in this chapter were also derived in [92, 94]. However in [92, 94] these 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 [92, 94] to rectify for passivity violations. For example in [94] 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 chapter, being convex, is guaranteed to converge to the global minimum and can be easily implemented using publicly available convex optimization solvers such as SeDuMi [2].

4.1 Rational Transfer Matrix Fitting in Pole Residue Form

The problem of constructing a rational approximation of multi port systems in pole residue form consists of finding residue matrices $R_k$, poles $a_k$ and the matrices $D$ & $F$ such that the identified model, defined by the transfer function $\hat{H}(s)$ in (4.1), minimizes the mismatch with the frequency response samples from the original system as described in (3.8).

$$\hat{H}(s) = \sum_{k=1}^{\kappa} \frac{R_k}{s - a_k} + D + sF$$  \hspace{1cm} (4.1)

here $R_k$, $D$ and $F$ are $n_p \times n_p$ residue matrices (assuming the system has $n_p$ ports) and $a_k$ are poles. Since most of the passive structures have a symmetric response, we consider the case when $R_k$, $D$ and $F$ are symmetric matrices. In the case when the matrices are non-symmetric, we can apply the same formulation to the symmetric part of the matrices.
4.2 Passive Fitting for Multiport LTI Systems

4.2.1 Problem Formulation

To formulate the problem, we expand the summation for \( \hat{H}(s) \) in (4.1) in terms of the purely real and complex poles. Also, since we are mainly interested in the properties of \( H(s) \) on the imaginary axis, we replace \( s \) with \( j\omega \).

\[
\hat{H}(j\omega) = \sum_{k=1}^{\kappa_r} \frac{R^r_k}{j\omega - a^r_k} + \sum_{k=1}^{\kappa_c} \frac{R^c_k}{j\omega - a^c_k} + D + j\omega F
\] (4.2)

Where \( \kappa_r \) and \( \kappa_c \) denote the number of purely real and the number of complex poles, respectively. Also, \( R^r_k \in \mathbb{R}^{n_p \times n_p}, \ R^c_k \in \mathbb{C}^{n_p \times n_p}, \ a^r_k \in \mathbb{R}, \ a^c_k \in \mathbb{C} \ \forall k, \) and \( D, F \in \mathbb{R}^{n_p \times n_p}. \)

In the following sections, we consider one by one the implications of each passivity condition in (2.9) on the structure of (4.2).

4.2.2 Conjugate Symmetry

Let us consider the implications of first condition of passivity on the structure of our proposed model in (4.2). The terms in (4.2) corresponding to the matrices \( D \) and \( F \), and to the summation over purely real poles satisfy automatically the property of conjugate symmetry in (2.9a). On the other hand such condition requires that the complex-poles \( a^c_k \) and complex residue matrices \( R^c_k \) always come in complex-conjugate-pairs

\[
\hat{H}(j\omega) = \sum_{k=1}^{\kappa_r} \frac{R^r_k}{j\omega - a^r_k} + \sum_{k=1}^{\kappa_c/2} \left\{ \frac{\Re R^c_k + j\Im R^c_k}{j\omega - a^c_k} + \frac{\Re R^c_k - j\Im R^c_k}{j\omega - a^c_k} \right\} + D + j\omega F
\] (4.3)

In (4.3) \( \Re \) and \( \Im \) indicate the real and imaginary parts respectively. Note that the summation for complex poles now extends only upto \( \kappa_c/2. \)
Lemma 1 (Conjugate Symmetry) Let $\hat{H}(j\omega)$ be a stable and conjugate symmetric transfer matrix given by (4.3), then $\hat{H}(j\omega)$ satisfy the property of conjugate symmetry $\overline{\hat{H}(j\omega)} = \hat{H}(j\omega)$.

Proof We show that the $\hat{H}(j\omega)$, as in (4.3) satisfies $\overline{\hat{H}(j\omega)} = \hat{H}(j\omega)$ by construction.

\[
\hat{H}(j\omega) = \sum_{k=1}^{\kappa_r} \frac{R_{k}^r}{-j\omega - a_k^r} + \sum_{k=1}^{\kappa_c/2} \left\{ \frac{\Re R_{k}^c + j\Im R_{k}^c}{-j\omega - Ra_k^c - j\Im a_k^c} + \frac{\Re R_{k}^c - j\Im R_{k}^c}{-j\omega - Ra_k^c + j\Im a_k^c} \right\} + D - j\omega F
\]

\[
\Rightarrow \overline{\hat{H}(j\omega)} = \sum_{k=1}^{\kappa_r} \frac{R_{k}^r}{j\omega - a_k^r} + \sum_{k=1}^{\kappa_c/2} \left\{ \frac{\Re R_{k}^c - j\Im R_{k}^c}{j\omega - Ra_k^c + j\Im a_k^c} + \frac{\Re R_{k}^c + j\Im R_{k}^c}{j\omega - Ra_k^c - j\Im a_k^c} \right\} + D + j\omega F
\]

\[
= \hat{H}(j\omega)
\]

Rewriting (4.3) compactly we get:

\[
\hat{H}(j\omega) = \sum_{k=1}^{\kappa_r} \hat{H}_k^r(j\omega) + \sum_{k=1}^{\kappa_c/2} \hat{H}_k^c(j\omega) + D + j\omega F \quad (4.4)
\]

where: $\hat{H}_k^r(j\omega) = \frac{R_{k}^r}{j\omega - a_k^r}$ \quad (4.5)

\[
\hat{H}_k^c(j\omega) = \frac{\Re R_{k}^c + j\Im R_{k}^c}{j\omega - Ra_k^c - j\Im a_k^c} + \frac{\Re R_{k}^c - j\Im R_{k}^c}{j\omega - Ra_k^c + j\Im a_k^c} \quad (4.6)
\]

4.2.3 Stability

The second condition (2.9b), which requires analyticity of $\hat{H}(s)$ in $\Re\{s\} > 0$, implies stability. For a linear causal system in pole-residue form (4.1), the system is strictly stable if all of its poles $a_k$ are in the left half of complex plane i.e. they have negative real part ($\Re\{a_k\} < 0$). Note that the system is marginally stable if conjugate pair poles with multiplicity one are present on the imaginary axis.
4.2.4 Positivity

The positivity condition for passivity (2.9c) is the most difficult condition to enforce analytically. We present here an extremely efficient condition which implies (2.9c). We consider the case when all the building blocks in the summation (4.4), namely:

- purely real poles/residues $\hat{H}_r^k(j\omega)$,
- complex-conjugate pairs of poles/residues $\hat{H}_c^k(j\omega)$,
- and the direct term matrix $D$ are individually positive real. Please note that the $j\omega F$ term has purely imaginary response and therefore does not affect positivity condition.

**Lemma 2** (Positive Real Summation Lemma) Let $\hat{H}(j\omega)$ be a stable and conjugate symmetric transfer matrix given by (4.4), then $\hat{H}(j\omega)$ is positive-real if $\hat{H}_r^k(j\omega)$, $\hat{H}_c^k(j\omega)$ and $D$ are positive-real $\forall k$. i.e.

$$\Re \hat{H}_r^k(j\omega) \succeq 0, \Re \hat{H}_c^k(j\omega) \succeq 0 \forall k \& D \succeq 0 \implies \hat{H}(j\omega) \succeq 0$$

(4.7)

**Proof** The sum of positive-real, complex matrices is positive real.

Lemma 2 describes a sufficient, but not-necessary, condition for (2.9c). However, as it will be shown in the examples, this condition is not restrictive.

In the following sections we derive the equivalent conditions of positive realness on each term separately.

**Purely Real Pole-Residues**

In this section we derive the condition for the purely real pole/residue term $\hat{H}_r^k(j\omega)$ in the summation (4.4) to be positive real. Such a condition can be obtained by rationalizing $\hat{H}_r^k(j\omega)$ defined in (4.5), as following:
\[
\hat{H}_k^c(j\omega) = \frac{R_k^r}{j\omega - a_k^r} + \frac{R_k^i}{j\omega - a_k^i} = \frac{R_k^r \times -j\omega - a_k^i}{j\omega - a_k^r - j\omega - a_k^i} = -\frac{a_k^i R_k^r}{\omega^2 + a_k^r} - j\frac{\omega R_k^r}{\omega^2 + a_k^r}
\]

(4.8)

\[
\Re(\hat{H}_k^c(j\omega)) \geq 0 \implies -\frac{a_k^i R_k^r}{\omega^2 + a_k^r} \geq 0 \quad \forall \omega, k = 1, \ldots, \kappa_r
\]

(4.9)

\[
\Re(\hat{H}_k^c(j\omega)) \geq 0 \implies -\frac{a_k^i R_k^r}{\omega^2 + a_k^r} \geq 0 \quad \forall \omega, k = 1, \ldots, \kappa_r
\]

(4.10)

**Complex Conjugate Pole-Residues**

In this section we derive the positive realness condition for the complex pole/residue term \(\hat{H}_k^c(j\omega)\) in the summation (4.4). Since complex terms always appear conjugate pairs, we first add the two terms for \(\hat{H}_k^c(j\omega)\) in (4.6) resulting into:

\[
\hat{H}_k^c(j\omega) = \frac{\Re R_k^c + j\Im R_k^c}{j\omega - \Re a_k^c - j\Im a_k^c} + \frac{\Re R_k^c - j\Im R_k^c}{j\omega - \Re a_k^c + j\Im a_k^c}
\]

(4.11)

\[
= -2\{\Re a_k^c\}(\Re R_k^c) - 2\{\Im a_k^c\}(\Im R_k^c) + j2\omega(\Re R_k^c)
\]

(4.12)

In order to obtain positive realness condition on \(\hat{H}_k^c(j\omega)\) we rationalize (4.12) to form (4.13). The resulting condition for \(\Re(\hat{H}_k^c(j\omega)) \geq 0\) is given in (4.14)

\[
\hat{H}_k^c(j\omega) = -\frac{2\{\Re a_k^c\}(\Re R_k^c) - 2\{\Im a_k^c\}(\Im R_k^c) + j2\omega(\Re R_k^c)}{((\Re a_k^c)^2 + (\Im a_k^c)^2 - \omega^2)^2 + j2\omega(\Re a_k^c)^2}
\]

(4.13)

\[
\Re(\hat{H}_k^c(j\omega)) \geq 0 \implies -2\{\Re a_k^c\}(\Re R_k^c) - 2\{\Im a_k^c\}(\Im R_k^c) - 2\omega^2\{\Re a_k^c\}(\Re R_k^c) - 2\omega^2\{\Im a_k^c\}(\Im R_k^c)
\]

(4.14)
Direct Term Matrix

Since $D$ is a constant real symmetric matrix, we require $D$ to be a positive semidefinite matrix, i.e.

$$D \succeq 0$$

4.2.5 The Constrained Minimization Problem

We combine all the constraints derived earlier and formulate a constrained minimization problem as follows:

$$\begin{align*}
\text{minimize} & \quad \sum_i \left| H_i - \hat{H}(j\omega_i) \right|^2 \\
\text{subject to} & \quad a^c_k < 0 \quad \forall k = 1, \ldots, \kappa_r \\
& \quad \Re a^c_k < 0 \quad \forall k = 1, \ldots, \kappa_c \\
& \quad \Re \hat{H}_r^k(j\omega) \geq 0 \quad \forall \omega, k = 1, \ldots, \kappa_r \\
& \quad \Re \hat{H}_c^k(j\omega) \geq 0 \quad \forall \omega, k = 1, \ldots, \kappa_c \\
& \quad D \succeq 0
\end{align*}$$

where

$$\hat{H}(j\omega) = \sum_{k=1}^{\kappa_r} \hat{H}_r^k(j\omega) + \sum_{k=1}^{\kappa_c/2} \hat{H}_c^k(j\omega) + D + j\omega F$$

Here $H_i$ are the given frequency response samples at frequencies $\omega_i$; $\hat{H}_r^k$ and $\hat{H}_c^k$ are defined in (4.5) and (4.6) respectively; $a^r_k$ and $a^c_k$ denotes the real and complex poles respectively. The detailed expressions for $\Re \hat{H}_r^k(j\omega) \geq 0$ and $\Re \hat{H}_c^k(j\omega) \geq 0$ are described in (4.10) and (4.14) respectively. The optimization problem described above in (4.15) is non convex. In the following section, we shall see how we can implement the relaxed version of (4.15) as a convex problem in terms of linear matrix inequalities.
4.3 Implementation

In this section we describe in detail the implementation of our passive multiport model identification procedure based on solving the constrained minimization framework developed in Section 4.2.

The optimization problem in (4.15) is non-convex because both the objective function and the constraints are non-convex. The non-convexity in (4.15) arises mainly because of the terms containing products and ratios between decision variables such as ratio of residue matrices, $R_k$, and poles, $a_k$, in the objective function, and product terms and ratios of $R_k$ and $a_k$ in the constraints.

Since the main cause of non-convexity in (4.15) is the coupling between $R_k$ and $a_k$, it is natural to uncouple the identification of unknowns, namely $R_k$ and $a_k$ in order to convexify (4.15). We propose to solve the optimization problem in (4.15) in two steps. The first step consists of finding a set of stable poles $a_k$ for the system. The second step consists of finding a passive multiport dynamical model for the system, given stable poles from step 1. In the following sections we describe how to solve the two steps.

4.3.1 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 [112]. Some schemes such as [5, 59] 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 employ a binary search based algorithm to automatically find the minimum number of poles required to achieve a user defined error bound on the mismatch between given frequency response samples and the frequency response of identified stable model.
4.3.2 Step 2: Identification of Residue Matrices

In this section we formulate the convex optimization problem for the identification of residue matrices using the stable poles from step 1. We first revisit the conditions for passivity (4.10) and (4.14), and later we shall develop the convex objective function.

Purely Real Pole-Residues

Let us consider the positive realness condition on the purely real pole residue term $H_r^r(j\omega)$ as in (4.10). The constraint (4.10) requires frequency dependent matrices to be positive semidefinite for all frequencies. This is in general very expensive to enforce. However, a careful observation of (4.10) reveals that the denominator, which is the only frequency dependent part of (4.10) is a positive real number for all frequency. This allows us to ignore the positive denominator which leaves us enforcing $-a_r^r R_r^r \succeq 0$. Since we are already given stable poles (i.e. $a_r^r < 0$), the constraint in (4.10) reduces to enforcing positive semidefiniteness on $R_r^r$.

$$\Re H_r^r(j\omega) \succeq 0 \implies R_r^r \succeq 0 \quad \forall k = 1, ... , \kappa_r \quad (4.16)$$

Such a constraint is convex and can be enforced extremely efficiently using SDP solvers [2,100].

Complex Conjugate Pole-Residues

In this section we reconsider the positive realness condition on the complex conjugate pole residue pair term $H_c^c(j\omega)$ as in (4.14). As before, a closer examination of the frequency dependent denominator in (4.14) reveals the fact that it is positive for all frequencies. Given that we have a fixed set of stable poles, and the denominator is always positive, we rewrite the constraint (4.14) only in terms of the variables i.e. $\omega$ and $R_c^c$. Also, we replace the constant expressions of $Re a_c^c$ and $Im a_c^c$ in (4.14) with generic constants $c_i$. We finally obtain the following equivalent condition.
\[ \Re \hat{H}_k^c(j\omega) \geq 0 \implies (c_1 \Re R_k^c + c_2 \Im R_k^c) + \omega^2 (c_3 \Re R_k^c + c_4 \Im R_k^c) \geq 0 \ \forall \omega, k = 1, \ldots, \kappa_c \] (4.17)

The problem is however still not solved since the condition in (4.17) is frequency dependent.

**Lemma 3** Let \( X_1, X_2 \in S^{n_p} \) and \( \omega \in [0, \infty) \), where \( S^{n_p} \) is the set of symmetric \( n_p \times n_p \) matrices, then

\[ X_1 + \omega^2 X_2 \succeq 0 \forall \omega \iff X_1 \succeq 0, X_2 \succeq 0 \] (4.18)

**Proof** Direction \( \Rightarrow \)

Given \( X_1 + \omega^2 X_2 \succeq 0 \) we consider the following limits:

\[ \lim_{\omega \to 0} (X_1 + \omega^2 X_2) \succeq 0 \implies X_1 \succeq 0 \]
\[ \lim_{\omega \to \infty} (X_1 + \omega^2 X_2) \succeq 0 \implies X_2 \succeq 0 \] (4.19)

Direction \( \Leftarrow \) follows from the fact that a non-negative weighted sum of positive semidefinite matrices is positive semidefinite. \( \blacksquare \)

We define

\[ X_1^k = c_1 \Re R_k^c + c_2 \Im R_k^c \]
\[ X_2^k = c_3 \Re R_k^c + c_4 \Im R_k^c, \] (4.20)

and apply Lemma 3 to the constraint defined in (4.17) which results into

\[ \Re \hat{H}_k^c(j\omega) \geq 0 \implies X_1^k \succeq 0, X_2^k \succeq 0 \ \forall k = 1, \ldots, \kappa_c \] (4.21)

Since \( X_1^k, X_2^k \) are linear combinations of the unknown matrices, \( \Im R_k^c \) & \( \Im R_k^c \) the
constraint (4.21) is a semidefinite convex constraint and thus can be enforced very efficiently.

**Convex Optimization to Find Residue Matrices**

In this section we summarize the final convex optimization identifying the residue matrices which correspond to to passive $H(j\omega)$, given stable poles $a_k$.

$$\begin{align*}
\text{minimize} & \quad \sum_i \left| \Re H_i - \Re \hat{H}(j\omega_i) \right|^2 + \sum_i \left| \Im H_i - \Im \hat{H}(j\omega_i) \right|^2 \\
\text{subject to} & \quad R^r_k \succeq 0 \quad \forall k = 1, \ldots, \kappa_r \\
& \quad -\Re a^c_k \Re R^c_k + \Im a^c_k \Im R^c_k \succeq 0 \quad \forall k = 1, \ldots, \kappa_c \\
& \quad -\Re a^c_k \Re R^c_k - \Im a^c_k \Im R^c_k \succeq 0 \quad \forall k = 1, \ldots, \kappa_c \\
& \quad D \succeq 0
\end{align*}$$

This final problem (4.22) is convex, since the objective function is a summation of $L_2$ norms. All the constraints in (4.22) are linear matrix inequalities. This convex optimization problem is a special case of semidefinite programming, requiring only few frequency independent matrices to be positive semidefinite. This problem formulation is extremely fast to solve, compared to other convex formulations [81,112] where the unknown matrices are frequency dependent.

**Adaptive Pole Placement**

In a standard two step procedure, 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. Specializing, 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 1. 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 during the next iteration.

**Algorithm 1 Adaptive Pole Placement Algorithm**

**Input:** $\kappa_{\text{initial}}, \kappa_{\text{increment}}, W_{\text{initial}} (W := \text{weights})$

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

**Complexity**

In the problem formulation (4.22), 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 $\eta = \frac{n_p(n_p+1)}{2} (\kappa + 2)$, where $\kappa$ is the number of poles and $n_p$ is the number of ports. If we are given $F$ frequency samples then the complexity of solving our problem is roughly $O(F^\nu \eta^\gamma)$, where $\gamma = 2$ and $\nu = 2.5$ typically.

**4.3.3 Equivalent Circuit Synthesis**

From the circuits perspective, the algorithm identifies a collection of low-pass, band-pass, high-pass and all-pass passive filter networks. These passive blocks can be readily synthesized into an equivalent passive circuit networks, and can be interfaced with commercial circuit simulators by either generating a spice-like netlist, or by using Verilog-A. Alternatively, we can develop equivalent state space realizations for our passive multiport models, for example a Jordan-canonical form can be obtained.
as described in [5] and then diagonalized.

### 4.3.4 The Complete Algorithm

In this section we present the description of the complete framework in Algorithm 2.

**Algorithm 2** Complete Passive Multiport Model Identification

**Input:** The set of frequency response samples \( \{H_i, \omega_i\} \), either the number of poles \( \kappa \) or the rms error bound \( \epsilon \)

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

1. Find stable poles \( a_k \) for the system
2. if \( \hat{\kappa} \) then
   3. \( \kappa_L \leftarrow 1, \kappa_U \leftarrow \kappa_{\text{MAX}} \)
   4. repeat
   5. \( t \leftarrow (\kappa_L + \kappa_U)/2 \)
   6. Find the stable system \( \hat{H}_t \) with \( t \) poles \( a_k \)
   7. Compute the rms error \( \epsilon_t = \sqrt{\sum_i |H_i - \hat{H}_t(j\omega_i)|^2} \)
   8. if \( \epsilon_t < \epsilon \) then
      9. \( \kappa_U \leftarrow t \)
   10. else
     11. \( \kappa_L \leftarrow t \)
   12. end if
   13. until \( \kappa_U = \kappa_L \)
  14. \( \kappa \leftarrow \kappa_U \)
  15. end if

16. Find the stable system with \( \kappa \) poles \( a_k \)
17. Solve the optimization problem (4.22) for \( R_k \)
18. Construct the model in pole/residue form as in (4.2)
19. Synthesize the equivalent passive circuit and generate the corresponding netlist or verilogA model file

This algorithm minimizes a cost function based on \( L_2 \) norm subject to linear matrix inequalities. Such a formulation can be solved very efficiently and is guaranteed to converge to the global minimum. However, the fact that this algorithm provides analytical expressions to enforce passivity in a highly efficient manner has an enormous potential such as in extensions to parameterized passive multiport models (discussed in Chapter 6); or to include designers specific constraints such as ensuring a good match for qualify factors in RF inductor dynamical models.
4.4 Proposed Unconstrained Minimization Setup

We observe that in the constrained minimization problem in (4.22), the constraints either enforce a matrix-valued variable to be positive definite or a linear combination of the matrix-valued variables to be positive definite. Positive definite matrices can be expressed by their Cholesky Decomposition. We propose a change of variables in (4.22). We propose to replace the positive definite constraints with the product of their Cholesky factors. This would imply that, for example $\mathbb{R}R_k$ in (4.22) will be replaced by $L_1L_1^T$, where $L_1$ is the Cholesky factor of $\mathbb{R}R_k$. After replacing the variables in (4.22) with their Cholesky factors, the new problem is unconstrained non-linear least squares minimization problem and is written down as

$$\min_{L_j} \sum_i \left( \left| \sum_j \alpha_j^{(\omega_i)} L_jL_j^T - \Re H^{(\omega_i)} \right|^2 + \left| \sum_j \beta_j^{(\omega_i)} L_jL_j^T - \Im H^{(\omega_i)} \right|^2 \right)$$  (4.23)

In (4.23), $L_j's$ are lower triangular Cholesky factors for our variables and define the new basis functions: $L_jL_j^T$. $\Re H^{(\omega_i)}$ and $\Im H^{(\omega_i)}$ are respectively the real and the imaginary parts of the given frequency response sampled at the frequency $\omega_i$. $\alpha_j^{(\omega_i)}$ and $\beta_j^{(\omega_i)}$ define the weights for the real and imaginary parts respectively, for the $j-th$ basis at the frequency $\omega_i$. Note that (4.23) is defined for matrix variables. To solve (4.23) we need to vectorize the variables and compute the equivalent function and its derivative (the Jacobian matrix).

4.4.1 Notations

We use the following notations for this section
\[ N = \text{Number of ports of the given system} \quad (4.24) \]
\[ P = \text{Number of poles} + 1 \quad (4.25) \]
\[ F = \text{Number of frequency points} \quad (4.26) \]
\[ f(\omega_i) \in \mathbb{R}^{N \times N} = \text{Function to be minimized, evaluated at } \omega_i \quad (4.27) \]
\[ L_j \in \mathbb{R}^{N \times N} = \text{Lower triangular unknown matrices} \quad (4.28) \]
\[ \Re H^{(\omega_i)} \in \mathbb{R}^{N \times N} = \text{Real part of the given frequency response} \quad (4.29) \]

### 4.4.2 Computing The Equivalent Function

For succinct description we shall consider the minimization problem only for the real part of frequency response in the following sections. It is very trivial to extend the formulation to include the imaginary part. The function to be minimized is given by

\[ f(\omega_i) = \hat{H}^{(\omega_i)} - \Re H^{(\omega_i)} \quad (4.30) \]

and

\[ \hat{H}^{(\omega_i)} = \sum_j \alpha_i^{(\omega_i)} L_j L_j^T \quad (4.31) \]

Rewriting (4.31) in matrix form gives us
\[ \hat{H}(\omega_i) = \sum_j L_j \alpha_j(\omega_i) L_j^T \]

\[ = L_1 \alpha_1(\omega_i) L_1^T + L_2 \alpha_2(\omega_i) L_2^T + \ldots + L_P \alpha_P(\omega_i) L_P^T \]

\[ = \begin{bmatrix} L_1 & L_2 & \ldots & L_F \end{bmatrix} \begin{bmatrix} \alpha_1(\omega_i) I_N \\ \alpha_2(\omega_i) I_N \\ \vdots \\ \alpha_F(\omega_i) I_N \end{bmatrix} = L A^{(\omega_i)} L^T \]  

(4.32)

Here \( L \) is the unknown matrix and is given by, \( L = \begin{bmatrix} L_1 & L_2 & \ldots & L_F \end{bmatrix} \) and \( A^{(\omega_i)} \) is the diagonal matrix in (4.32). Writing the function (4.30) using (4.32) for all frequencies:

\[ f(\omega_1) = L A^{(\omega_1)} L^T - \Re H^{(\omega_1)} \]

\[ f(\omega_2) = L A^{(\omega_2)} L^T - \Re H^{(\omega_2)} \]

\[ \vdots \]

\[ f(\omega_F) = L A^{(\omega_F)} L^T - \Re H^{(\omega_F)} \]  

(4.33)

Writing (4.33) in matrix form:

\[ \begin{bmatrix} f(\omega_1) \\ f(\omega_2) \\ \vdots \\ f(\omega_F) \end{bmatrix} = \begin{bmatrix} L \\ L \\ \vdots \\ L \end{bmatrix} \begin{bmatrix} A^{(\omega_1)} \\ A^{(\omega_2)} \\ \vdots \\ A^{(\omega_F)} \end{bmatrix} L^T - \begin{bmatrix} \Re H^{(\omega_1)} \\ \Re H^{(\omega_2)} \\ \vdots \\ \Re H^{(\omega_F)} \end{bmatrix} \]

\[ f = (I_F \otimes L) AL^T - \Re H \]  

(4.34)
Here $I_F$ is an identity matrix of size $F \times F$.

### 4.4.3 Computing The Jacobian Matrix

We use chain rule to compute the derivative of the function (4.34).

$$f = (I_F \otimes L)A L^T - \Re H$$

$$f = f_1f_2 - \Re H$$

where $f_1 = (I_F \otimes L)$

$$f_2 = A L^T$$

Differentiating (4.36) with respect to $L$

$$\frac{\partial f}{\partial L} = (f_2^T \otimes I_{NF}) \frac{\partial f_1}{\partial L} + (I_N \otimes f_1) \frac{\partial f_2}{\partial L}$$

where $\frac{\partial f_1}{\partial L} = (I_F \otimes T_{NP,F} \otimes I_N)(\text{vec}(I_F) \otimes I_{N,NP})$

$\frac{\partial f_2}{\partial L} = (I_N \otimes A)T_{N,NP}$

Here $T$ is the permutation matrix. We define $T_{m,n}$ as the matrix that transforms $\text{vec}(A)$ into $\text{vec}(A^T)$ [39] (The $\text{vec}(.)$ operator vectorized a matrix by stacking its columns).

$$T_{m,n}\text{vec}(A) = \text{vec}(A^T)$$

Note that the size of this matrix is $mn \times mn$. $T$ is composed of 0’s and 1’s, with a single 1 on each row and column. When pre-multiplying another matrix, it simply rearranges the ordering of rows of that matrix (post multiplying by $T_{m,n}$ rearranges columns).

$$\frac{\partial f}{\partial L} = (L A^T \otimes I_{NF})(I_F \otimes T_{NP,F} \otimes I_N)(\text{vec}(I_F) \otimes I_{N,NP}) + (I_N \otimes I_F \otimes L)(I_N \otimes D)T_{N,NP}$$

(4.43)
We need to simplify the expression for the Jacobian matrix in order to minimize the online computations and compute most functions offline. Here we refer online computations as the ones that are performed inside an active solver loop, whereas offline computations are the ones that can be performed once outside the solver loop and their results can be reused. We use the property \((AC \otimes BD) = (A \otimes B)(C \otimes D)\) of Kronecker product to get the following expression.

\[
LA^T \otimes I_{NF} = (L \otimes I_{NF})(A^T \otimes I_{NF}) = T_{N,NF}(I_{NF} \otimes L)T_{NF,NP}(A^T \otimes I_{NF})
\]

(4.44)

Substituting (4.44) in the (4.43) gives us

\[
\frac{\partial f}{\partial L} = T_{N,NF}(I_{NF} \otimes L)T_{NF,NP}(A^T \otimes I_{NF})(I_F \otimes T_{NP,F} \otimes I_N)(\text{vec}(I_F) \otimes I_{N,NP}) + (I_{NF} \otimes L)(I_N \otimes A)T_{N,NP}
\]

(4.45)

Using \(C_1\) and \(C_2\) to describe the matrices computed offline, we get

\[
\frac{\partial f}{\partial L} = T_{N,NF}(I_{NF} \otimes L)C_1 + (I_{NF} \otimes L)C_2
\]

(4.46)

where \(C_1 = T_{NF,NP}(A^T \otimes I_{NF})(I_F \otimes T_{NP,F} \otimes I_N)(\text{vec}(I_F) \otimes I_{N,NP})\)

\[C_2 = (I_N \otimes A)T_{N,NP}\]

Please note that although we have analytically computed the Jacobian using Kronecker products, we do not need to compute and store the intermediate Kronecker products. In the actual implementation, we propose a direct assembly of the Function and its Jacobian using stamping procedures.
4.5 Results

In this section we shall present modeling examples of various multiport passive structures. All examples are implemented in Matlab and run on a laptop with Intel Core2Duo processor with 2.1GHz clock and 3GB of main memory.

4.5.1 Wilkinson Combiner in a LINC Amplifier

In this section we shall present an example illustrating the usefulness of our proposed methodology for modeling and simulating a LINC (Linear amplification with Nonlinear Components) power amplifier [114]. The architecture, as described in Figure 4-1, consists of a signal splitter, two power amplifiers, and a Wilkinson type power combiner. This architecture is designed to operate at 40 GHz. PA1 and PA2 are class B amplifiers designed in 130nm SiGe process using BJTs. The Wilkinson combiner is designed on alumina substrate with characteristic impedance of 50Ω and operating frequency of 40 GHz.

![Figure 4-1: Block diagram of the LINC power amplifier architecture](image)

Input, $v_{in}$, to this architecture is a $64 - QAM$ signal. The signal splitter decomposes the input $QAM$ signal into two phase modulated fixed amplitude signals. Let $v_{in} = V_{in} \angle \phi$ be the input signal; $v_1 = V_0 \angle \phi_1$ and $v_2 = V_0 \angle \phi_2$ be the two signals generated by the splitter then,

$$v_{in} = v_1 + v_2, \quad V_{in} \angle \phi = V_0 \angle \phi_1 + V_0 \angle \phi_2 \quad (4.47)$$

The splitted signals are amplified by individual nonlinear power amplifiers. The
outputs of these two power amplifiers are added using a Wilkinson type power combiner. This 3-port Wilkinson combiner, as shown in Figure 4-2, is simulated inside a full wave public domain field solver [95] available at [3]. Using the frequency response samples generated by the field solver, a closed form state space model of order $q = 30$ is identified using our passive modeling algorithm. To demonstrate the accuracy of this model in frequency domain Figure 4-3 compares the impedance parameters from field solver (dots) and frequency response of our identified passive model (solid lines). The modeling error $e_{i,k}(\omega)$, defined by (4.48), was less than 0.7% for all $i,k$ in the bandwidth of interest between $2GHz - 60GHz$.

$$e_{i,k}(\omega) = \frac{|H_{i,k}(j\omega) - \hat{H}_{i,k}(j\omega)|}{\max_{i,k,\omega}|H_{i,k}(j\omega)|}$$  \hspace{1cm} (4.48)

The algorithm took only 2seconds to generate the entire model, whereas for the same order and similar accuracy the algorithm described in [112] took 83seconds giving us a speed-up of 40×. A model is passive if there are no purely imaginary eigen values of the associated Hamiltonian matrix. Figure 4-4 is a zoomed-in plot of the eigen values of the associated hamiltonian matrix for the identified model. It is clear that the model passes the passivity test since there are no purely imaginary eigen values.

Finally, the overall amplifier architecture is simulated inside a commercial circuit simulator after connecting the linear model for the combiner with the rest of the circuit components including the nonlinear amplifiers, as shown in Figure 4-5.

Figures 4-6(a) and 4-6(b) plots the normalized input ($v_{in}$) and output ($v_{out}$) volt-
Figure 4-3: Comparing real and imaginary part of the impedance parameters from field solver (dots) and our passive model (solid lines). The mismatch, defined by (4.48), is $e_{i,k}(\omega) < 0.7\% \ \forall \ i, k, \omega \in [2, 60] \text{GHz}$

Figure 4-4: Plotting the zoomed-in eigen values of the associated hamiltonian matrix for the identified model of Wilkinson combiner
Figure 4-5: Block diagram of the LINC power amplifier architecture as simulated inside the circuit.

![Block diagram of the LINC power amplifier architecture](image)

(a) Ideal normalized 64-QAM input voltage $v_{in}$  (b) Normalized output voltage $v_{out}$ generated by transient simulation of the overall architecture in Figure 4-5.

Figure 4-6: Normalized input and output 64-QAM signals

Practically speaking, as verified in Figures 4-6(a) and 4-6(b), the passive nature of the identified model for the Wilkinson combiner guarantees that transient simulations for the overall architecture converge, and the final output signal $v_{out}$ is also a $64-QAM$ signal similar to the input $v_{in}$.

### 4.5.2 Power & Ground Distribution Grid

The second example we present is a power & ground distribution grid used in systems on chip or on package. The 3D layout for this power grid is shown in Figure 4-7, and is composed of five $Vdd$ (red or dark grey) and $Gnd$ (green or light grey) segments placed along both $x$ and $y$ axes. External connections, given by solder balls in a flip chip.
technology, are modeled with bond wires running vertically. Important parameters of this power grid are as follows: die size = 10mm × 10mm, wire width = 20µm, wire height = 5µm, vertical separation = 4µm, gnd-vdd separation = 20µm, bond-wire lengths = 500µm and solder ball radius = 20µm. This structure was simulated using 52390 unknowns in the full wave mixed potential integral equation (MPIE) solver, Fast Maxwell [95], to obtain frequency response samples up to 12 GHz. The multiport simulation was arranged by placing eight ports: four at the grid corners and four inside the grid. Ports are illustrated in Figure 4-7 as black strips.

For this example our proposed algorithm identified an 8×8 passive transfer matrix of order m = 160 in 74 seconds, whereas the algorithm in [112] ran out of memory and did not generate the model. To demonstrate the accuracy, Figure 4-8 compares the real and imaginary impedance respectively of our reduced model with the field solver data.

Although the models are passive by construction, the passive nature was verified by the absence of purely imaginary eigen values of the associated hamiltonian matrix.
Figure 4-8: Comparing real and imaginary parts of the impedance from our passive model (solid line) and from the field solver (dots) for a power distribution grid

A necessary condition for a model to be passive requires $\lambda_{\text{min}}(\Re\{\hat{H}(j\omega_i)\}) > 0$. We plot $\lambda_{\text{min}}(\Re\{\hat{H}(j\omega_i)\})$ for our identified model within the bandwidth of interest in Figure 4-9.

### 4.5.3 On-Chip RF Inductors

The third example is a collection of 4 RF inductors on the same chip or package that are used in the design of multichannel receivers. The layout is shown in Figure 4-10. The array is comprised of four inductors laid out in the form of a 2x2 matrix. Important dimensions of this array are as follows: wire width= $10\mu m$, wire height= $4\mu m$, height of inductors above substrate= $20\mu m$, horizontal separation between sides of two adjacent inductors= $400\mu m$, length of sides of each inductor= $800\mu m, 600\mu m, 400\mu m, 200\mu m$, and having 4, 3, 3, 2 turns respectively. The structure has four ports in total, configured at the input of each inductor. This structure was simulated using 10356 unknowns in the full wave field solver, FastMaxwell [95] which
Figure 4-9: $\lambda_{\text{min}}(\Re \{ \hat{H}(j\omega) \})$

Figure 4-10: 3D layout of the RF inductors (wire widths not to scale)
Figure 4-11: Comparing real part and imaginary of impedance from our passive model (solid line) and from field solver (dots) for the RF inductors captures substrate using a Green function complex image method.

For this example a $4 \times 4$ passive transfer matrix of order $m = 92$ was identified. The algorithm took 72 seconds to identify the passive model, compared to the algorithm in [112] which ran out of memory and did not generate the model.

Figure 4-11 shows impedance parameters both from the field solver and from our identified model. The passive nature of this model was verified by the absence of pure imaginary eigen values of the associated hamiltonian matrix.

4.5.4 1-Port Structure

We obtained this single port frequency response from the code distribution package available on the website of the authors of [112]. We have tested [112] 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 (4.49) for
our work and for [28, 58] are very similar. However, despite the fact that this was the only example for which we were able to test [112], the total error for [112] was very large (i.e. 161%).

$$e_{total} = \sqrt{\sum_i |\Re H_i - \Re H(\omega_i)|^2 + \sum_i |\Im H_i - \Im H(\omega_i)|^2}$$  \hspace{1cm} (4.49)

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

### 4.5.5 4-Port Structure

In this section we discuss in detail a 4-port test case. To demonstrate the accuracy of our generated model, we’ve plotted $Y_{11}$ in the form of a 3D plot in Figure 4-12. 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 $e_{i,k}(\omega) = \frac{|H_{i,k}(j\omega) - \hat{H}_{i,k}(j\omega)|}{\max_i,\omega |H_{i,k}(j\omega)|}$ in Figure 4-13 which shows that the peak error is less than 2.5%.

We provide a comparison with the algorithm presented in [28]. 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 4-14 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 4-14 that for same model order, we get orders of magnitude improvement both in terms of speed and memory compared to [28]. On average we observed a significant $80 \times$ speedup. We note that even for very small model orders, very large amount of memory is required by [28], 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 [28].

Although our proposed technique is theoretically more restrictive then the one
Figure 4-12: 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.

Figure 4-13: 4-port structure: Error between given data and the identified transfer matrix. Each curve corresponds to the mismatch for individual transfer function.
Figure 4-14: **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.

in [28] and [112]. 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.

### 4.5.6 16-Port Structure

In this section we present a 16-port example. We computed a model with $\kappa = 9$ poles. The accuracy of the generated model is demonstrated in Figures 4-15 and 4-17. We see that our algorithm scales efficiently with the number of ports, compared to other optimization based algorithms such as [28, 112] 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 [58]. It is clear form Figure 4-16 that [58] lost significant accuracy during the passivity enforcement step, while our passive model
Figure 4-15: **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.

achieved an excellent match with the given samples while simultaneously enforcing passivity. The total error, defined by (4.49) is 16.171% for [58] while it is only 2.391% for our algorithm. Passivity of our identified model was further verified by both Hamiltonian matrix based test and by checking the feasibility of the final solution.

### 4.5.7 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 $q = 300$. In Figure 4-18 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.
Figure 4-16: **16-port structure**: Comparing $Re\{Z(1,1)\}$ from our passive model (solid red line) with [58] (dashed blue line) and given samples (asterisks). Our passive model matches the given samples perfectly while the passive model from [58] loses accuracy after passivity enforcement.

Figure 4-17: **16-port structure**: Error between given data and the identified transfer matrix.
Figure 4-18: **60-port structure**: Magnitude of frequency response samples from given data (green broken lines) and from our models (red solid line).

### 4.5.8 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 (4.22).

We note that the algorithms presented in [28] and [112], ran out of memory for most of the test cases and did not generate the model. For the smaller test cases, where [28] did generate a model, we gained significant speed-up (22× and 116×) at the price of small degradation in accuracy. Compared to [58] we gain accuracy for some test cases, while for others, the algorithm in [58] performs better. This is because for some of the test cases the initially identified stable model in [58] is already passive or have minor passivity violations, in such a scenario, the model generated by [58] 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 [58]. For such examples our algorithm gives better accuracy.

To summarize, we encourage the user to first try the relatively inexpensive algo-
Figure 4-19: Proposed framework for passive model identification

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 [58], then the user may switch to our algorithm to get better results. The proposed flow is demonstrated in the Figure 4-19.
Chapter 5

Passive Fitting for Multiport Systems
- Method II

In this chapter we present efficient algorithms that can generate passive linear dynamical models from scattering representation (S-parameters) with guaranteed convergence to the global optimal solution (in the given relaxation), and can be characterized by moderate computational requirements both in terms of CPU time and memory occupation. An attempt was made in [22] to formulate the perturbation based approaches in a convex framework. This resulted in a convex but non-smooth optimization problem, which was solved in [22] using general descent schemes based on projected or alternate subgradient iterations. Even though the methods in [22] guarantee the optimality of the solution, they can be highly sensitive to the given problem specific parameters, and require 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 our proposed strategy, we solve the convex non-smooth passivity enforcement problem of [22] using efficient localization schemes such as the ellipsoid method [12] and the cutting plane method [71]. The main differentiating factor between our work and [22] is that the algorithms implemented in this chapter 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 [22] require excessive iterations. Our framework generates optimal passive models with the same quality of [28], 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 method are as follows:

- application of efficient localization methods to solve the convex non-smooth passivity enforcement formulation presented in [22];

- 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;

- construction of a lower bound on the objective function that gets tighter with every iteration;

- implementation of efficient modifications and improvements (e.g., deep cuts formulations) to the standard localization methods, see, e.g., [41].

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.

5.1 Background

In this section we review some of the concepts used in this chapter including linear dynamical modeling and localization methods. Additional definitions are provided in Appendix B.
5.1.1 Problem Description

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

$$H(0, s) = C(sI - A)^{-1}B + D.$$  \hfill (5.1)

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 (5.1) 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 [33, 34, 59, 63] can be applied to these samples in order to identify a state-space dynamical model (5.1) with minimal deviation from the raw data. In a least-squares formulation, this amounts to solving

$$\min_{A, B, C, D} \sum_{\psi=1}^{F} \|H(0, j\omega_\psi) - S_\psi\|^2.$$  \hfill (5.2)

Problem (5.2) is addressed in the literature [33, 34, 59, 63], hence we consider the nominal dynamical model (5.1) as our starting point.

System (5.1) is assumed to be asymptotically stable and the state-space realization is assumed to be minimal. A stable system (5.1) is passive if and only if its $H_\infty$ norm is unitary bounded,

$$\|H(0)\|_{H_\infty} = \sup_{\omega \in \mathbb{R}} \sigma_1(H(0, j\omega)) \leq 1,$$  \hfill (5.3)

where $\sigma_1$ denotes the largest singular value [15, 20, 62]. In the cases where (5.3) 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...
Figure 5-1: Algorithmic flow for perturbation based passivity enforcement

framework is summarized in Figure 5-1.

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 [48]. 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

$$H(C_P, s) = (C + C_P)(sI - A)^{-1}B + D$$  \hspace{1cm} (5.4)$$

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

$$\text{minimize } \|C_P\|_F \quad \text{ s.t. } \|H(C_P)\|_{\infty} \leq 1,$$  \hspace{1cm} (5.5)$$

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

\[
\begin{align*}
\min_{C_P} \quad & \|C_P G^T\|_F \quad \text{s.t.} \quad \|H(C_P)\|_{\mathcal{H}_\infty} \leq 1, \\
\end{align*}
\]  
(5.6)

where \(G\) is the Cholesky factor of the controllability Gramian of the system. It is shown in [46] that (5.6) yields a solution that provides minimal impulse response perturbation in the \(L^2\) (energy) norm. Since (5.6) is readily reduced to (5.5) by a change of variable, we will base our derivations on (5.5) without loss of generality.

Using vectorized variable \(x = \text{vec}(C_P) \in \mathbb{R}^n\) \((n = qn_p)\) and rewriting (5.5) we get

\[
\begin{align*}
\min_x \quad & f(x) \quad \text{s.t.} \quad h(x) \leq 1, \\
\end{align*}
\]  
(5.7)

where

\[
f(x) = \|x\|_2 = \|C_P\|_F, \quad h(x) = \|H(C_P)\|_{\mathcal{H}_\infty}.
\]  
(5.8)

Any local minimum is also a global minimum for Problem (5.7), 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 \(x = -x_c = -\text{vec}(C)\). This is because

\[
x = -x_c \implies \|H(C_P)\|_{\mathcal{H}_\infty} = \|H(-C)\|_{\mathcal{H}_\infty} = \|D\|_2 \leq 1
\]  
(5.9)

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 (5.7) has a unique globally optimal solution. Notice further that, as described in [22], \(h(x)\) is convex and continuous but non-smooth.

\subsection*{5.1.2 Computation of the \(\mathcal{H}_\infty\) Norm}

Accurate computation of the \(\mathcal{H}_\infty\) norm is a critical step in solving (5.5). 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, \( \omega \), 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 (5.10).

\[
||H(C_P)||_{\mathcal{H}_\infty} \approx \max_k \sigma_1(H(C_P,j\omega_k)) \quad (5.10)
\]

The memory and time required for computing (5.10) are \( O(n_p^2) \) and \( O(\gamma n_p^3) \) respectively. Here \( \gamma \) is the number of samples. Computing the \( \mathcal{H}_\infty \) norm via (5.10) 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 [15, 20, 62]. The memory and time required for these methods are \( O(q^2) \) and \( O(\tau q^3) \) respectively. Here \( \tau \) 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 work we use [20] to compute the \( \mathcal{H}_\infty \) norm.

### 5.1.3 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 (5.7). The two localization methods that we employ in this work are the ellipsoid algorithm [12] and the cutting plane method [71]. In the following sections we provide an intuitive description of the two algorithms to solve the generic problem defined in (5.7).

**The Ellipsoid Algorithm**

The ellipsoid algorithm, proposed in [72], starts with an initial ellipsoid (defined in Appendix B.0.3) that is guaranteed to contain the global minimum. Figures 5-2
and 5-3 illustrate the first two iterations of the algorithm. In Figure 5-2 the yellow convex region describes the feasible region $h(x) \leq 1$. The blue sphere describes the initial ellipsoid $\epsilon^{(0)}$. 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 $\{x|g^T(x - x^{(0)}) \leq 0\}$, which is shown by the shaded region in Figure 5-2, 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 $\epsilon^{(1)}$ is the smallest ellipsoid containing the intersection of the original ellipsoid $\epsilon^{(0)}$ and the half space $\{x|g^T(x - x^{(0)}) \leq 0\}$. The updated ellipsoid $\epsilon^{(1)}$ is shown in the right half of Figure 5-2.

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 $\{x|g^T(x - x^{(1)}) \leq 0\}$, depicted by shaded region in Figure 5-3, 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 $\epsilon^{(2)}$ is the smallest ellipsoid defined by the intersection of current ellipsoid $\epsilon^{(1)}$ and the half space $\{x|g^T(x - x^{(1)}) \leq 0\}$. The updated ellipsoid $\epsilon^{(2)}$ is shown in Figure 5-3.

This is continued until the size of the updated ellipsoid is small enough such that all of its interior points fall within the $\delta$ neighborhood of the global minimum. The algorithmic details are described in Algorithm 3.
Algorithm 3 Ellipsoid Algorithm

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

1: Set $k = 0$
2: If $h(x(k)) \leq 1$, 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 $h(x(k)) \leq 1$ and $\eta_k \leq \delta$, then return $x(k)$ and quit
5: Update ellipsoid parameters $x(k+1)$ and $P_{k+1}$ using (5.11)
6: Let $k \leftarrow k + 1$ and goto 1.

The updates, as described in [12], are given by

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

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

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 $\text{vol}(\epsilon_{k+1}) < e^{-\frac{n}{2}} \text{vol}(\epsilon_k)$.

**The Cutting Plane Method**

The cutting plane method [71] also belongs to the class of localization methods. This approach uses a *polyhedron* (defined in Appendix B.0.4) instead of an ellipsoid to define the search region. The algorithm starts with an initial polyhedron $P_0$ that is guaranteed to contain the global minimum. At each step the algorithm adds a new
constraint to the current polyhedron $P_k$ such that the updated polyhedron $P_{k+1}$ is smaller and still contains the global minimum.

Figures 5-4 and 5-5 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 $\{x | g^T(x - x^{(0)}) \leq 0\}$ 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 $\{x | g^T(x - x^{(0)}) \leq 0\}$, as shown in Figure 5-4.

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 gradient $g = \nabla f$ of the objective function, corresponding to the dotted line in Figure 5-5. The half space $\{x | g^T(x - x^{(1)}) \leq 0\}$ 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 $\{x | g^T(x - x^{(0)}) \leq 0\}$, as shown in Figure 5-5.

Algorithm 4 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 $f_{\text{best}}$ attained throughout all previous iterations.

**Algorithm 4 Cutting Plane Method**

**Input:** $f, h$ and $P_0$ such that $x^* \in P_0$; initialize the lower bound $L_0 = 0$ and $f_{\text{best}} = f(x_{\text{feasible}})$; accuracy $\delta > 0$, set $k = 0$

1. If $h(x^{(k)}) \leq 1$, 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 $h(x^{(k)}) \leq 1$ and $|f_{\text{best}} - L_{\text{best}}| < \delta$, then return $x^{(k)}$ and quit
4. Update $\mathcal{P}$: add a new cutting plane $a_{k+1}^T z \leq b_{k+1}$, $\mathcal{P}_{k+1} := \mathcal{P}_k \cap \{z | a_{k+1}^T z \leq b_{k+1}\}$
   here $a_{k+1} = g_k$ and if feasible $b_{k+1} = g_k^T x^{(k)}$ else $b_{k+1} = g_k^T x^{(k)} - h(x^{(k)}) + 1$
5. Update $x^{(k+1)} \in \mathcal{P}_{k+1}$, $L_k$
6. Update $L_{\text{best}} = \max_k L_k$, $f_{\text{best}} = \min_k f(x_{\text{feasible}}^{(k)})$
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, $\text{vol}(\mathcal{P}_{k+1}) \leq (1 - 1/n)\text{vol}(\mathcal{P}_k)$ [17].
5.2 Passivity Enforcement using Localization Methods

In this work we employ the localization methods described in the Section 5.1.3 to solve the convex continuous but non-smooth problem described in the Section 5.1.1. 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.

5.2.1 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.

Upper Bound on $R$ ($R_{UB}$)

Computation of an upper bound on $R$ is pictorially described in Figure 5-6. To understand this, consider the following Lemma.

Lemma 4 Let $f(x) = ||x||_2$ be the cost function, and let the feasible set be defined by $h(x) \leq 1$. Suppose that we are given any feasible point $x_f$ such that $h(x_f) \leq 1$. 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
h(x) ≤ 1

Figure 5-6: [Left] R_{UB}, [Right] R_{LB}

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*) ≤ f(x_f).

As described in Section 7.2, 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

$$
\beta^* = \arg\min_{\beta \in [-1, 0]} f(\beta x_c) \quad \text{s.t.} \quad h(\beta x_c) \leq 1
$$

$$
R_{UB} = ||\beta^* x_c||_2
$$

(5.12)

If instead a weighted objective function, such as (5.6), is used the cost function becomes

$$
f(x) = ||W x||_2,
$$

(5.13)

where W = G \otimes I and I is the identity matrix. The derivation of (5.13) is described in Appendix C.1. For this case, we can compute the upper bound on R by R_{UB} = ||W x_c||_2, which can be further shortened by the line search (5.12).
Lower Bound on $R (R_{LB})$

We define $R_{LB}$ to be the radius of an infeasible hypersphere $\epsilon_{LB}$ centered at the initial point $x^{(0)}$, as described in Figure 5-6. We assume that the initial unperturbed system is non-passive, hence the initial point $x^{(0)}$ is infeasible. The hypersphere $\epsilon_{LB}$ 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 $x \in \epsilon_{LB}$ are infeasible we get

$$h(x) > 1 \quad \forall x \in \epsilon_{LB}. \quad (5.14)$$

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

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

$$\geq h(x^{(0)}) + \inf_{z \in \epsilon_{LB}} \partial h(x^{(0)})^T(z - x^{(0)})$$

$$= h(x^{(0)}) - \sqrt{\partial h(x^{(0)})^T P_{LB} \partial h(x^{(0)})}, \quad (5.15)$$

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

$$\inf_{z \in \epsilon_{LB}} \partial h(x^{(0)})^T(z - x^{(0)}) = -\sqrt{\partial h(x^{(0)})^T P_{LB} \partial h(x^{(0)})}, \quad (5.16)$$

see Appendix C.2 for a derivation. From (5.15), 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 $\epsilon_{LB}$ are infeasible,

$$h(x) \geq h(x^{(0)}) - \sqrt{\partial h(x^{(0)})^T P_{LB} \partial h(x^{(0)})} > 1,$$

$$h(x^{(0)}) - 1)^2 > \partial h(x^{(0)})^T R_{LB}^2 I \partial h(x^{(0)}). \quad (5.17)$$

Solving (5.17) gives us $R_{LB}$ as

$$R_{LB} = \frac{h(x^{(0)}) - 1}{\sqrt{\partial h(x^{(0)})^T \partial h(x^{(0)})}} = \frac{h(x^{(0)}) - 1}{\|\partial h(x^{(0)})\|_2}. \quad (5.18)$$
This means that for any initial hypersphere 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.

**Practical selection of \( R \)**

In this section we provide guidelines on selecting the radius \( R \) of the initial hypersphere. From earlier discussions in Section 5.2.1 and 5.2.1, the value of \( R \) must satisfy the following inequality

\[
R_{LB} < R \leq R_{UB} \tag{5.19}
\]

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

\[
R = \min(\zeta R_{LB}, R_{UB}) \tag{5.20}
\]

### 5.2.2 Passivity Enforcement Using The Ellipsoid Algorithm

**Initialization**

The functions \( f(x) \) and \( h(x) \) for passivity enforcement are defined in (5.8). The gradient \( \nabla f(x) \) and sub-gradients \( \partial h(x) \) are derived in [22] and are not repeated here. We initialize Algorithm 3 by defining an initial ellipsoid, \( \epsilon^{(0)} \), that is guaranteed to contain the global optimum \( x^* \). We define \( \epsilon^{(0)}(x^{(0)}, P_0) \) to be the hypersphere, \( P_0 = R^2 I_n \), with the radius \( R \) and centered around the origin. The radius \( R \) is computed by (5.19).
Complexity

The main attractive feature of the ellipsoid algorithm is that it is extremely efficient in terms of the memory usage.

Computationally, Algorithm 3 has two major components a) Computation of the $\mathcal{H}_\infty$ norm (or $h(x)$), described in Section 5.1.2 b) Updating and storing ellipsoid parameters. Their corresponding time and memory complexities are given in Table 5.1.

<table>
<thead>
<tr>
<th>Component</th>
<th>memory</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) $\mathcal{H}_\infty$ norm</td>
<td>$O(q^2)$</td>
<td>$O(\tau q^3)$</td>
</tr>
<tr>
<td>(b) Ellipsoid Parameters</td>
<td>$O(n^2) \equiv O(q^2 n_p^2)$</td>
<td>$O(n^2) \equiv O(q^2 n_p^2)$</td>
</tr>
</tbody>
</table>

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 5.1.2, 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 $O(\tau q^3)$.

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 [41], 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

$$g^T (x - x^{(k)}) + g_{DC}^{(k)} \leq 0 \quad (5.21)$$
In the deep cut ellipsoid algorithm, the updated ellipsoid at each iteration is given by:

\[ \epsilon^{(k+1)} = \epsilon^{(k)} \cap \{ x | g^T (x - x^{(k)}) + g_{DC}^{(k)} \leq 0 \} \]  

(5.22)

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

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

(5.23)

here \( f_{\text{best}}^{(k)} \) is the best objective value of the feasible iterates so far. The updates are given by (41)

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

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

(5.24)

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 [86]. Proof of convergence of the deep cut ellipsoid algorithm is provided in [41].

5.2.3 Passivity Enforcement Using The Cutting Plane Method

Initialization

We define the initial set (a polyhedron) required for the cutting plane method, described in Algorithm 4, to be the smallest hypercube \( P_0 \) enclosing the hypersphere with radius \( R \). The radius \( R \) is computed by (5.19). This hypercube \( P_0 \) is guaranteed to contain the global minimum. \( P_0 \) is centered around the origin with side length \( 2R \). We shall refer to the size of this hypercube as \( R \).
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 $P_{k+1}$. 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 $P_{k+1}$ 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 $P_{k+1} = \{ z | a^T_i z \leq b_i, \ i = 1...q \}$. This gives us a well rounded performance both in terms of computational cost and convergence. We compute the analytic center by solving (5.25) using infeasible start Newton method

$$x_{k+1} = \arg\min_x - \sum_i \log(b_i - a_i^T x)$$  \hspace{1cm} (5.25)

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 5.2, is dominated by the evaluation of the $H_\infty$ 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>
<thead>
<tr>
<th>Component</th>
<th>memory</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) $H_\infty$ norm</td>
<td>$O(q^2)$</td>
<td>$O(\tau q^3)$</td>
</tr>
<tr>
<td>(b) Polyhedron Parameters</td>
<td>$O(mn^2) \equiv O(mq^2 n_p^2)$</td>
<td>$O(mn^2) \equiv O(mq^2 n_p^2)$</td>
</tr>
</tbody>
</table>

Table 5.2: Cost per Iteration

87
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 chapter, this helps in reducing the number of iterations by more than 20\% when compared with the original cutting plane method used in [86]. The epigraph form [17] of our problem (5.7) is given as follows

\[
\begin{align*}
\text{minimize} & \quad t \\
\text{subject to} & \quad f(x) \leq t \\
& \quad h(x) \leq 1
\end{align*}
\] (5.26)

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

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

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

\[
f(x^{(k)}) + g^T(x - x^{(k)}) \leq t \quad \text{and} \quad t \leq f(x^{(k)})
\] (5.27)

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

5.2.4 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

\[
f(z) \geq f(x_i) + g_i^T(z - x_i), \quad \forall z, i = 1, \ldots, k.
\] (5.28)
This implies that the function \( \hat{f} \) defined as

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

(5.29)

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

\[
x^*_{LB} = \arg\min \hat{f}(x) \quad s.t. \hat{h}(x) \leq 1,
\]  

(5.30)

which can be solved using standard linear programming solvers (in order to solve (5.30), per iteration a standard primal-dual interior point method [91] 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 (5.7). 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 \( \delta \)-optimality of the solution with the help of this lower bound (by definition, the distance of a \( \delta \)-optimal solution from the true optimal solution is less than \( \delta \)). For this purpose, note that the linear program (5.30) 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

\[
l_k = \max_{\forall k \mid h(x^{(k)}) \leq 1} \left( f(x^{(k)}) - \sqrt{g_k^T P_k g_k} \right).
\]  

(5.31)
5.3 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 [25, 34, 59], whose complexity scales only as $O(F\kappa^2)$. This cost is negligible with respect 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.

5.3.1 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 [59] 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 5-7. 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 $\delta$ neighborhood of the global minimum. We also plot the progress for the ellipsoid algorithm without deep cuts, shown by red diamonds in Figure 5-7. Note that by using deep cuts we achieve a significant speed-up of $100\times$ over the original ellipsoid algorithm used in [86].

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
Figure 5-7: 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.)
Figure 5-8: Example 1: comparison between passive model and original non-passive model

compute the $H_\infty$ norm [20].

Figure 5-8 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 5-9 compares the frequency dependent singular values of the original and the passive model. As reported in [22], standard techniques such as [46] generated only a suboptimal and less accurate passive model. The alternate gradient method presented in [22] took 900 iterations (833 seconds) to converge. This gives us a speed-up of 35× over [22].

5.3.2 Example 2: A 4-Port Transformer

We recall that the technique presented in [28] 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 3.4.2. With the proposed techniques, we are able to obtain the same optimal solution with a comparable ac-
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 [28] and the ellipsoid algorithm described in this work. We monitor the memory and time requirements for increasing model orders. Figures 5-10 and 5-11 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 up to 1000× in terms of total run time. Figures 5-10 and 5-11 clearly demonstrate the scalability of our algorithm.

5.3.3 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 5-12. This test case corresponds to a SAW filter with 2 ports, whose initial model is characterized by 36 poles, with $C \in \mathbb{R}^{2 \times 72}$ and the corresponding unknown vector $x = \text{vec}(C_P) \in \mathbb{R}^{144}$. 

Figure 5-9: Example 1: singular values of the original non-passive (solid blue) and the perturbed passive (dashed red) models are plotted versus frequency.
Figure 5-10: Example 2: comparison of the memory requirements for our framework and [28].

Figure 5-11: Example 2: comparison of the time required for our framework and [28].
Figure 5-12: Example 3: singular values of the original non-passive (solid blue) and perturbed passive (dashed red) models plotted against normalized frequency.

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 (5.6). We anticipate the passivity enforcement results in Figure 5-13, 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 5-14 shows the convergence of the cost function 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 (5.30). Using this lower bound, we can guarantee that the computed solution is $\delta$–optimal, so that we control how far we are from the actual optimum. For this case, the algorithm converged in 30
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 5-15. 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 5-15. Note that by using deep cuts we achieve a speed-up of $3 \times$ over the original ellipsoid algorithm used in [86].

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 [46,52] failed to generate a passive model. The alternate gradient algorithm presented in [22] took about 15,000 iterations.
5.3.4 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 5-16. 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 $C \in \mathbb{R}^{2 \times 68}$ and the unknown vector $x = \text{vec}(C_P) \in \mathbb{R}^{136}$. The bounds on the initial feasible set are $R_{UB} = 2.4$ and $R_{LB} = 0.04$, respectively.

Figures 5-17 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 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 5-18. The algorithm found a feasible point in 40 seconds at the 1400-th iteration. We also plot...
Figure 5-15: Example 3: convergence of the ellipsoid algorithm with and without deep cuts.

Figure 5-16: Example 4: singular values of the original non-passive (solid blue) and perturbed passive (dashed red) models plotted against normalized frequency.
Figure 5-17: Example 4: cutting plane method using analytic centering. The cost function is plotted against the iteration number for different initial hypercubes.

the progress for the ellipsoid algorithm without deep cuts, shown by the dashed red line in Figure 5-18. Note that by using deep cuts we achieve a speed-up of more than $2 \times$ over the original ellipsoid algorithm used in [86]. For this example, we get an average speed up of $250 \times$ over [22].

5.3.5 Example 5: 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. 5-19. 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 [20]. Memory usage was less
Figure 5-18: Example 4: convergence of the ellipsoid algorithm with and without deep cuts.

than 90 MB.

Figure 5-20 demonstrates the accuracy of our passive model even in the presence of large passivity violations. Compared with the suboptimal technique [46], the proposed scheme is able to achieve passivity with a smaller perturbation amount, hence increased accuracy. Application of the alternative optimal technique [28] 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.

5.3.6 Example 6: A SAW Filter

For this 3-port SAW filter, an initial stable but non passive model with 144 states was computed using vector fitting [59] 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. 5-21. 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.
Figure 5-19: Example 5 (packaging interconnect): singular values of original non-passive (solid blue) and proposed passive (dashed red) models.

Figure 5-20: Example 5 (packaging interconnect): original non-passive and perturbed passive model responses ($S_{8,13}$)
in 30 seconds. However, note that over 82% of the time was spent in computing the $H_{\infty}$ norm. Memory usage was less than 1 MB.

Figure 5-22 demonstrates the accuracy of our passive model even in the presence of large passivity violations. Also in this case, as expected, the proposed model is more accurate than with suboptimal techniques such as [46]. An optimally accurate model could also be obtained with the algorithm [28], which resulted in a problem with 14250 equations with 46156 variables. The required memory, as reported by matlab was over 2 GB, and the corresponding runtime was about 2.5 hours. We conclude that proposed approach is able to compute an optimal passive macromodel with a fraction of memory and orders of magnitude speed-up with respect to [28].

5.3.7 General Discussion

For the two localization-methods presented in this chapter, 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,
Figure 5-22: Example 6 (SAW filter): original non-passive and perturbed passive model responses ($S_{3,3}$)

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 chapter 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 chapter 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.
Chapter 6

Globally Passive Parameterized Model Identification

6.1 Motivation

Globally passive parameterized models are essential if one wishes to explore the design space of a complex system containing interconnected linear and non-linear components. In these models, passivity is required for the whole continuous parameter range, since the user or the optimizer can instantiate the models with any parameter value. The ability to generate passive parameterized models would greatly facilitate the circuit designers. As an example consider a multi-primary transformer which is used for power combining in distributed power amplifiers, as shown in Figure 6-1. The transformer design variables are length, width and spacing for the windings. With a parameterized modeling tool, the designer would be able to create an equivalent circuit block which approximates the dependence of frequency response on design parameters with high fidelity. Such a parameterized modeling tool should also give apriori guarantees of stability and passivity if the final model is to be used in an interconnected environment. The user can interface these equivalent circuit blocks with circuit simulators and run full system simulations, where s/he has the control over design parameters including length, width and spacing. These parameter values can be fine-tuned for optimal power amplifier performance.
Figure 6-1: A multiprimary transformer parameterized in length, width and spacing. The equivalent circuit block with parameter controlling knobs interfaced with circuit simulator is used for design space exploration of a complete distributed power amplifier design.

6.2 Background

6.2.1 Interpolatory Transfer Matrix Parameterization

A common approach to constructing parameterized transfer matrix models is to interpolate between a collection of non-parameterized models, each generated at a different parameter value. Given a set of individual models and parameter values \( \{H_i, \lambda_i\} \), where each model is described in pole-residue form (4.1), meaning \( H_i = \{R_i, A_i, D_i\} \) where

\[
A_i = \begin{bmatrix}
a_{1,i} \\
\vdots \\
a_{\kappa,i}
\end{bmatrix},
R_i = \begin{bmatrix}
R_{1,i} \\
\vdots \\
R_{\kappa,i}
\end{bmatrix}
\]

are a collection of the poles and residues respectively, then the goal is to construct...
functions $\hat{R}(\lambda)$, $\hat{a}_k(\lambda)$, and $\hat{D}(\lambda)$ that interpolate exactly the given pole-residue models,

$$\hat{R}(\lambda_i) = R_i \quad \hat{D}(\lambda_i) = D_i \quad \hat{A}(\lambda_i) = A_i,$$

resulting in a parameterized transfer matrix in pole-reside form

$$\hat{H}(s, \lambda) = \sum_{k=1}^{\kappa} \frac{\hat{R}_k(\lambda)}{s - \hat{a}_k(\lambda)} + \hat{D}(\lambda),$$

(6.3)

We $\hat{A}(\lambda)$ and $\hat{R}(\lambda)$ as follows

$$\begin{bmatrix} \hat{a}_1(\lambda) \\ \vdots \\ \hat{a}_\kappa(\lambda) \end{bmatrix}, \quad \begin{bmatrix} \hat{R}_1(\lambda) \\ \vdots \\ \hat{R}_\kappa(\lambda) \end{bmatrix}.$$

(6.4)

For instance, in [118] piecewise linear functions were used to interpolate exactly the given models, while in [32] interpolation was achieved using Lagrange polynomials. Although such approaches guarantee exact matching at the given set of parameter values used for interpolation, they do so at the expense of being able to ensure reasonable behavior between the points, and they require an extremely large number of coefficients to describe a model interpolating many points. Piecewise linear fits are not smooth, and therefore preclude the use of models for sensitivity analysis where derivatives with respect to the parameter are necessary. Methods based on polynomial interpolation are smooth, but are likely to produce non-monotonic oscillatory curves, resulting in non-physical behavior of the model between the interpolation points.

Additionally, while interpolation approaches can guarantee that the resulting parameterized model $H(s, \lambda)$ is passive (or stable) when evaluated at the set of parameter values $\lambda_i$ used for interpolation (provided the original set of non-parameterized models are all stable), no guarantees can be made about the passivity (or stability) of the model when evaluated at any parameter value differing from the small set used...
for interpolation.

6.2.2 Positivity of Functions

Enforcing passivity in a system inevitably relies on ensuring positivity (or non-negativity) of some quantity in the system, such as functions in (6.4). A positivity requirement on an arbitrary function is in general a non-convex constraint making this a difficult task. However, one way to ensure that functions are globally positive is to require them to be expressible as a quadratic form of a positive semidefinite (PSD) matrix. That is, if we want to ensure that a scalar multivariate function is globally non-negative, i.e. \( f(x) \geq 0 \) is satisfied for all possible \( x \), then we can choose to construct \( f(x) \) such that it can be expressed as \( \phi(x)^T M \phi(x) \) for some PSD matrix \( M \). Here \( \phi(x) \) can be any vector of nonlinear functions of the argument \( x \). If such a construction is used to describe \( f \), then it is guaranteed that \( f(x) = \phi(x)^T M \phi(x) \geq 0 \) for all possible \( x \). If instead we wish \( f(x) \) to be globally negative, we simply require that \( M_k \) is a negative semidefinite matrix.

Positivity of Polynomials

Let us consider the special case when the function \( f(x) \) is a polynomial. A sufficient condition (necessary for univariate case) for the polynomial \( p(x) = f(x) \) to be non-negative is that it can be written as a sum of squares (SOS). i.e.

\[
\text{If } \quad p(x) = \sum_i g_i^2(x) = \text{SOS} \quad \implies \quad p(x) \geq 0 \quad \forall x \quad (6.5)
\]

A sum of squares polynomial can be represented as a quadratic form of a positive semidefinite matrix. i.e.

\[
p(x) = \sum_i g_i^2(x) = \text{SOS} \quad \iff \quad p(x) = \phi^T M \phi, \quad M \succeq 0 \quad (6.6)
\]
here $\phi$ is a vector of monomials. As an example, suppose we want to certify that

$$p(s) = 1 + 4x + 5x^2 \geq 0 \quad \forall x \quad (6.7)$$

We can certify (6.7) if we can find an $M \succeq 0$ such that it satisfies (6.6). Equation (6.8) shows that indeed we can such $M$

$$p(x) = \begin{bmatrix} 1 & x \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix}, \quad M = \begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix} \succeq 0 \quad (6.8)$$

Also by performing Cholesky decomposition of $M$ we can express $p(x)$ as SOS explicitly.

$$p(x) = 1 + 4x + 5x^2$$
$$= \begin{bmatrix} 1 & x \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix}$$
$$= \begin{bmatrix} 1 & x \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix}$$
$$= (1 + 2x)^2 + x^2 \geq 0 \quad \forall x \quad (6.9)$$

### 6.3 Optimal Parameterized Fitting

#### 6.3.1 Problem Formulation

To circumvent the issues resulting from interpolatory fitting approaches described in Section 6.2.1, we formulate the parameterized modeling problem as an approximate fitting problem using optimization. Given a set of $\Omega$ non-parameterized models \{\(H_i, \lambda_i\}\}, we wish to construct a parameterized model $\hat{H}(s, \lambda)$ in the form of (6.3) such that $\hat{H}(s, \lambda_i) \approx H_i(s)$. We assume the models given to us are described in pole residue form (4.1), meaning $H_i = \{A_i, R_i, D_i\}$, as defined in (6.1). This assumption is not restrictive because if models are only available in numerator-denominator form
or in state-space form, it is a trivial task to transform them into pole-residue from. It is advantageous to fit models in pole-residue form because the parameters of such model (e.g. the dominant poles) have a physical meaning and can be expected to vary smoothly as the parameters in the system are varied. A state-space model, on the other hand, is a non-unique representation of the system (e.g. any rotation of the coordinate system would produce completely different coefficients in the system matrices) and therefore there is no reason to think that interpolating between such systems is a reasonable task.

Our goal is to fit parameterized functions describing the poles, residues, and feed through matrix such that

\[
\hat{A}(\lambda_i) \approx A_i, \quad \hat{R}(\lambda_i) \approx R_i, \quad \hat{D}(\lambda_i) \approx D_i. \tag{6.10}
\]

By using optimization-based fitting, as opposed to exact interpolation, we can greatly reduce the number of terms necessary to fit a model to a large number of points. Furthermore, we eliminate the non-physical oscillatory behavior that may arise between interpolation points when fitting high order polynomials to relatively well-behaved curves.

We generally assume that the poles and residues are both complex. Therefore we will identify separately the real and imaginary parts of each

\[
\hat{a}_k(\lambda) = \Re \hat{a}_k(\lambda) + j \Im \hat{a}_k(\lambda) \tag{6.11}
\]

\[
\hat{R}_k(\lambda) = \Re \hat{R}_k(\lambda) + j \Im \hat{R}_k(\lambda) \tag{6.12}
\]

The result of this formulation is a set of five minimization problems, that solve for the real and imaginary parts of the poles, residue matrices, and direct matrix as a
function of the parameters

\[
\begin{align*}
\min \sum_{i=1}^{\Omega} ||\Re A_i - \Re \hat{A}(\lambda_i)||^2, & \quad \min \sum_{i=1}^{\Omega} ||\Im A_i - \Im \hat{A}(\lambda_i)||^2, \\
\min \sum_{i=1}^{\Omega} ||\Re R_i - \Re \hat{R}(\lambda_i)||^2, & \quad \min \sum_{i=1}^{\Omega} ||\Im R_i - \Im \hat{R}(\lambda_i)||^2, \\
\min \sum_{i=1}^{\Omega} ||D_i - \hat{D}(\lambda_i)||^2.
\end{align*}
\] (6.13)

The accuracy of the resulting model, and the difficulty of solving the optimization problems, depends on how we choose to describe the unknown functions \( \hat{A}(\lambda), \hat{R}(\lambda), \hat{D}(\lambda) \).

In the following section we propose a convenient formulation resulting in a semidefinite optimization problem, which can be easily solved using standard freely available software.

### 6.3.2 Rational Least Squares Fitting

Consider the case where the unknown functions to be identified are described as the ratio of two unknown functions, each of which is expressed as a linear combination of basis functions

\[
\begin{align*}
g(\lambda) = \frac{\alpha(\lambda)}{\beta(\lambda)} &= \frac{\sum_{n=1}^{N} \alpha_n \phi_n(\lambda)}{\sum_{m=1}^{M} \beta_m \psi_m(\lambda)}.
\end{align*}
\] (6.14)

Here \( \psi \) and \( \phi \) are predetermined basis functions, and although we use the term ‘rational’, we are not forcing \( \phi \) and \( \psi \) to be polynomials, and it is not even necessary that \( \phi \) and \( \psi \) be the same class of functions.

Given samples \( g_i \), the optimization task consists of solving for coefficients \( \alpha_n \) and \( \beta_m \) in order to minimize

\[
\sum_i \left| \left| g_i - \frac{\alpha(\lambda_i)}{\beta(\lambda_i)} \right| \right|^2.
\]

Unfortunately, attempting to minimize directly such quantity is a difficult (i.e. non-convex) task due to the nature of the nonlinear dependence on the unknown coefficients \( \alpha_n, \beta_m \). Instead, a useful relaxation [81] of this objective transforms it into the
following convex problem

$$
\min_{\alpha, \beta} \Omega \sum_{i=1}^{\Omega} \frac{||\beta(\lambda)g_i - \alpha(\lambda_i)||^2}{\beta(\lambda)}.
$$

(6.15)

Along with the normalization constraint that $\sum m \beta_m = 1$, this problem can be viewed as a weighted least squares minimization of the desired objective $||g_i - \alpha(\lambda_i)/\beta(\lambda_i)||$. Although the problem is still nonlinear in the unknowns, the formulation is convex (specifically, minimization over a ‘rotated Lorentz cone’) and thus can be solved efficiently using freely available software [116].

In order to utilize such a description to solve (6.13), one must select the basis functions describing the five unknown quantities ($\Re \hat{a}, \Im \hat{a}, \Re \hat{R}, \Im \hat{R}, \hat{D}$), and solve a set of five optimization problems in the form of (6.17). Note that the basis functions used to describe the five different quantities need not be the same.

### 6.3.3 Linear Least Squares

If we restrict the denominator in rational formulation (6.14) to be constant unity (i.e. $\beta(\lambda) = 1$), then we can consider functions defined as a linear combination of basis functions

$$
g(\lambda) = \sum_{n=1}^{N} \alpha_n \phi_n(\lambda),
$$

(6.16)

which simplifies the optimization problem to the standard ‘linear least squares’ problem

$$
\min_{\alpha} \sum_{i=1}^{\Omega} ||g_i - g(\lambda_i)||^2.
$$

(6.17)

This optimization problem is an unconstrained linear least squares minimization that is convex and can be solved without any relaxation using freely available software [116], or even solved analytically.
6.3.4 Polynomial Basis Example

As an illustrative example of how one uses optimization problem (6.15) to identify the parameterized model, suppose we wish to fit the real part of the poles, $\Re \hat{A}(\lambda)$, to a function of two parameters $\lambda_1, \lambda_2$ using a polynomial basis. If we choose a second order polynomial basis for the denominator $\psi$, meaning that $\Psi = [1, \lambda_1, \lambda_2, \lambda_1 \lambda_2, \lambda_2^2]$, and a first order polynomial basis for the numerator $\phi$, meaning $\Phi = [1, \lambda_1, \lambda_2]$, then the resulting function expression for each individual pole $\Re \hat{a}_k(\lambda)$ would be

$$\Re \hat{a}_k(\lambda) = \frac{\alpha_0 + \alpha_1 \lambda_1 + \alpha_2 \lambda_2}{\beta_0 + \beta_1 \lambda_1 + \beta_2 \lambda_2 + \beta_3 \lambda_1^2 + \beta_4 \lambda_1 \lambda_2 + \beta_5 \lambda_2^2}.$$  

This expression is then used in optimization problem (6.15) to solve for the unknown coefficients $\alpha, \beta$ for each of the $\kappa$ poles.

If instead we wish to use a polynomial function description to solve the linear least squares problem (6.17), then a second order polynomial basis yields the function

$$\Re \hat{a}_k(\lambda) = \alpha_0 + \alpha_1 \lambda_1 + \alpha_2 \lambda_2 + \alpha_3 \lambda_1^2 + \alpha_4 \lambda_1 \lambda_2 + \alpha_5 \lambda_2^2.$$  

It is important to point out here that exact polynomial interpolation, such as in [32], can be thought of as a very special case of our framework. Specifically, if we choose polynomials as basis functions, select the basis $\phi$ to allow as many degrees of freedom as data points (i.e. $\kappa = \Omega$), and choose the linear least squares formulation, then the unconstrained optimization problem (6.17) is equivalent to exact polynomial interpolation. However, unlike interpolation formulations, our optimization approach is neither confined to using a polynomial basis nor to using as many coefficients as data points, and therefore the complexity of the resulting model does not scale poorly with the number of points used for fitting.

6.3.5 Complexity of Identification

The cost of identifying parameterized model (6.3) using the previously described optimization procedure is extremely cheap because many of the unknown quantities
are uncoupled, and additionally there are multiple redundancies within the system. As previously mentioned, the real and imaginary parts of the poles can be solved for separately, as described in (6.13). If the model is described by \( \kappa \) poles, then solving for \( \Re \hat{A}(\lambda) \) and \( \Im \hat{A}(\lambda) \) can each be separated into \( \kappa \) different optimization problems, because the poles do not depend upon one another. Additionally, since the residue matrices and direct term are symmetric, for a system with \( n_p \) ports, there will only be \( n_p(n_pk + 1)/2 \) unique elements to fit (corresponding to the upper triangular part of the symmetric matrices). Lastly, since the complex poles and residues occur in conjugate pairs, it is only necessary to fit half of the non-zero imaginary pole parts in \( \Im \hat{A}(\lambda) \), and only necessary to fit half of the residue matrices \( \Im \hat{R}(\lambda) \).

6.4 Constrained Fitting for Stability

The previously posed unconstrained optimization problems (6.13) enforce optimal accuracy of the parameterized transfer function with respect to the individual models, but provides no guarantees on global properties, such as stability, for the system. Although point-wise interpolation as described in Section 6.2.1 will yield a model that is stable at the parameter values used for fitting (assuming the original non-parameterized models are each stable), it provides no stability guarantee when evaluating the model at any other parameter value. In this section we formulate additional constraints for the previously defined optimization problems that allow us to generate guaranteed stable parameterized transfer function models.

6.4.1 Stable Pole Fitting

Stability of a model in pole-residue from (4.1) depends only on the real part of the poles \( \Re \hat{A}(\lambda) \). We say the parameterized model \( H(s, \lambda) \) is stable at a particular parameter value \( \lambda \) if all of the poles have negative real part at that parameter value, i.e. \( \Re \hat{A}(\lambda) < 0 \). Thus, enforcing guaranteed stability of \( H(s, \lambda) \) for all parameter values \( \lambda \) can be achieved by enforcing negativity of \( \Re \hat{A}(\lambda) \) for all such \( \lambda \). Note that since we are modeling the residues and direct term separately, enforcing stability only
affects one of the five optimization problems in (6.13).

Enforcing negativity when fitting the real part of the poles $\Re \hat{A}(\lambda)$ requires adding a constraint to the previously unconstrained minimization problems. If solving the linear least squares problem (6.16), the constrained problem becomes

$$
\min_{\Re \hat{A}} \sum_i ||\Re A_i - \Re \hat{A}(\lambda_i)||^2 \quad \text{subject to (6.18)}
$$

$$
\Re \hat{a}_k(\lambda) < 0 \quad \forall \ k.
$$

If we are instead fitting to rational functions as described in (6.14), then the pole function is negative if the numerator $\Re a_k(\lambda)$ is negative and the denominator $\Re \beta_k(\lambda)$ is positive, resulting in the constrained optimization problem

$$
\min_{\Re \hat{A}} \sum_i ||\Re A_i - \Re \hat{A}(\lambda_i)||^2 \quad \text{subject to (6.19)}
$$

$$
\Re \alpha_k(\lambda) < 0, \quad \Re \beta_k(\lambda) > 0 \quad \forall \ k
$$

To enforce these positivity and negativity constraints, we will require the functions to be describable as positive definite quadratic forms in some nonlinear basis. That is, define $\Re a_k(\lambda) = \Lambda(\lambda)^T M_k \Lambda(\lambda)$ where $\Lambda(\lambda)$ is a vector of functions of $\lambda$, and if $M_k$ is a positive definite matrix, then $\Re a_k(\lambda) > 0$ is a positive function for all $\lambda$. This is a standard technique for enforcing positivity of functions as described in Section 6.2.2, and can similarly be used to enforce negativity by requiring $M_k$ to be a negative definite matrix. Problems of the form (6.19) with semidefinite matrix constraints can be solved using semidefinite optimization solvers [116].

For example, if fitting to a quadratic function of two parameters, we would select

$$
\Lambda(\lambda) = \begin{bmatrix} 1 \\ \lambda_1 \\ \lambda_2 \end{bmatrix}
$$

(6.20)
and would define $\Re a_k(\lambda_i) = \Lambda(\lambda_i)^T M_k \Lambda(\lambda_i)$, and $M_k$ would contain the unknown coefficients $\alpha_n$.

6.5 Constrained Fitting For Passivity

In addition to stability, it is desirable that the identified models also be passive. In this section we extend the optimization framework presented in Section 4.3.2 of Chapter 4 to include parameterization while preserving passivity.

6.5.1 Parameterized Residue Matrices

To enforce global passivity, we require that the individual non-parameterized models conform to the passivity conditions described in Chapter 4. Hence in our final parameterized model we want the constraints in 4.22 to be satisfied for all values of the parameter. Given stable approximation of poles $a(\lambda)$ we compute passive residue matrices by solving the following convex optimization problem

$$\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{\Omega} ||\Re R_i - \Re \hat{R}(\lambda_i)||^2 + \sum_{i=1}^{\Omega} ||\Im R_i - \Im \hat{R}(\lambda_i)||^2 \\
\text{subject to} & \quad R^r_k(\lambda) \succeq 0 \quad \forall k = 1, \ldots, \kappa_r \forall \lambda \\
& \quad -\Re a^c_k(\lambda) \Re R^c_k(\lambda) + \Im a^c_k(\lambda) \Im R^c_k(\lambda) \succeq 0 \quad \forall k = 1, \ldots, \kappa_c \forall \lambda \\
& \quad -\Re a^c_k(\lambda) \Re R^c_k(\lambda) - \Im a^c_k(\lambda) \Im R^c_k(\lambda) \succeq 0 \quad \forall k = 1, \ldots, \kappa_c \forall \lambda \\
\end{align*}$$

where $R^c_k(\lambda) = \Re R^c_k(\lambda) + j\Im R^c_k(\lambda)$

$R_i$ and $\hat{R}(\lambda_i)$ are defined in (6.1) and (6.4) respectively. Note that the constraints in (6.21) require matrix valued functions to be positive definite, such a constraint can be enforced using SOS relaxation as described in Section 6.2.
6.5.2 Positive Definite Direct Matrix

One necessary condition for passivity of a model described in pole-reside form 4.1 is that the direct term $D$ be a positive semidefinite matrix.

$$\minimize_{D(\lambda)} \sum_{i=1}^{\Omega} \left|\left|D_i - D(\lambda_i)\right|\right|^2$$
subject to $D(\lambda) \succeq 0$

We notice that since each of the given individual models is passive, then $D_i \succeq 0$ and can therefore be factored into a matrix $V_i$ such that $D_i = V_i^TV_i$. Therefore, in order to enforce positive definiteness in $\hat{D}(\lambda)$, we will instead fit $\hat{V}(\lambda)$ to factors of the given direct terms $V_i$, and then define $\hat{D}(\lambda) = \hat{V}(\lambda)^T\hat{V}(\lambda)$. Since we are enforcing positivity after identification, the result is an unconstrained minimization problem

$$\min_{\hat{V}} \sum_i \left|\left|V_i - \hat{V}(\lambda_i)\right|\right|^2$$  \hspace{1cm} (6.21)

that can be solved using either linear least squares formulation (6.16) or the rational formulation (6.14) to describe $\hat{V}(\lambda)$.

6.6 Implementation

In this section we describe in detail our parameterized model identification procedure based on solving the optimization problems derived in Section 6.4 and 6.5. We also discuss methods for properly generating the individual non-parameterized models and methods for transforming the identified parameterized transfer matrix into a circuit usable in commercial simulators.

6.6.1 Individual Model Identification and Preprocessing

The ability to obtain a reasonable functional approximation between models depends crucially on the assumption that the poles trace out some nice smooth curve as the
parameters vary. For the dominant poles of the system this is a safe assumption, because the dominant poles in the individual linear models have a physical connection to the original large system. However, for the other less significant poles in the individual identified models, there may be many possible pole configurations that all produce a good match when originally creating those individual models. Therefore, when performing the original identification of the individual non-parameterized models, it may be beneficial to aid the pole placement of the identification procedure as shown in Algorithm 5. To perform this task, we identify poles for the first or nominal parameter value. We then use the poles identified for the previous parameter value as the initial guess to identify poles for neighboring parameter values. This way we ensure the smoothness in the path of the poles. Also if we want to enforce global passivity in the final parameterized model, the initial non-parameterized models must be identified using the algorithm described in Chapter 4.

**Algorithm 5** Successive Pole Placement Algorithm  
1: Given $\Omega$ frequency response data sets  
2: Identify first non-parameterized model $H_1(s)$ described in pole-residue form (4.1)  
3: for $i=2:\Omega$ do  
4: Use pole set $A_{i-1}$ as initial guess for pole set $A_i$, and identify model $H_i(s)$  
5: end for  

Once the collection of models $H_i$ in pole-residue form, each containing $\kappa$ poles, has been identified using Algorithm 5, the next task is to construct $\kappa$ functions, fitting each pole trajectory. Therefore, it is crucial that we ‘order’ the $\kappa$ poles such that they are ‘clustered’ accordingly. We employ a recursive Euclidean distance based clustering approach as described in Algorithm 6. A robust implementation of this algorithm is required since we may encounter complicated scenarios, such as poles crossings and bifurcations. Figure 6-2 shows sample clusters for a system with three poles.

### 6.6.2 Parameterized Identification Procedure

The main part of our complete modeling approach, presented in Algorithm 7, is to solve for parameterized functions describing the real and imaginary parts of the poles,
residues, and direct term. These optimization problems were derived in Section 6.4 and 6.5. There are two important details we wish to emphasize in this part. First, to enforce positive definiteness on the direct term $D(\lambda)$, we fit instead to the factor $V(\lambda)$ and then define $D(\lambda) = V(\lambda)^TV(\lambda)$, as described in Section 6.5. Second, We want to emphasize that when solving for the real part of the poles $\Re A(\lambda)$ we must enforce negativity to ensure stability of the resulting model, as was described in Section 6.4.1. The remaining components of the model can be solved for using either the unconstrained minimization described in Section 6.3 if passivity is not required, or the constrained minimization described in Section 6.5 if apriori global passivity is required.

6.6.3 Post-processing Realization

The parameterized models generated by our approach can be readily converted into equivalent circuit blocks using VerilogA. Since our models are guaranteed to be stable in the parameter space, we use voltage sources as the ‘controlling knobs’. These voltage sources are used to change parameter value after the model is instantiated inside the circuit simulator.
Several equivalent state space realizations for our parameterized models can be achieved. For example, a Jordan-canonical form can be obtained as described in [5]. However in order to have a better performance, such a realization needs to be diagonalized before interfacing with the circuit simulators.

6.7 Results

In this section we shall present three examples highlighting different aspects of our proposed methodology in modeling guaranteed stable models. All examples in this section are implemented in Matlab and run on a laptop having Intel Core2Duo processor with 2.1GHz clock, 4GB of main memory, and running windows vista. We have also posted free open source software implementing these procedures online [1].

6.7.1 Single port - Single parameter: Microstrip Patch Antenna

The first example considered is a microstrip square patch antenna, shown in Figure 6-3. It is designed on Rogers RT5800 substrate having thickness of 25mils (1mil = 0.001inch), relative permittivity $\epsilon_r = 2.2$ and loss tangent $\tan\delta = 0.0009$. The layout is shown in Figure 6-3. In order to control the resonant frequency, we select the side length of the square \('L'\) as the model parameter. A collection of individual non-parameterized models were generated by simulating the structure for S-parameter samples from $0.5GHz$ to $2GHz$ using SONNET Lite, where $L$ was varied from $4000mils$ to $5000mils$ with an increment of $100mils$. The resulting models
each have order $\kappa = 5$ and were generated and preprocessed for each value of $L$ as described in Algorithm 5 and Algorithm 6 of Section 6.6.1.

A stable parameterized model was then identified using Algorithm 2 along with a polynomial basis of degree $N = 8$ for each of the model components. For this example the entire fitting procedure was completed in just 2.48 seconds.

To illustrate the guaranteed stability of the resulting parameterized model, Figure 6-4 plots the trajectory of poles in response to changes in parameter $L$ variation. The thick black line, corresponding to our stable parameterized model, is always in the left half plane, meaning the model is stable at all parameter values. On the other hand, the thin green line, corresponding to a parameterized model generated using polynomial interpolation without stability constraints, crosses into the right half plane and is unstable for many parameter values within the shown range.

To verify the accuracy of our parameterized model, Figure 6-5 compares the fre-
Figure 6-5: Comparison of magnitude of frequency responses of patch antenna parameterized model (dashed lines) with the initial non-parameterized models (solid lines-almost overlapping) for different parameter values. Some traces are from parameter values not used for fitting.

The frequency response magnitude of our model (dashed red line) to the response of individual non-parameterized models at a set of different parameter values, some of which were not used for fitting. Furthermore, in order to show that our model response smoothly to changes in the parameter over the entire range of interest, Figure 6-6 plots the frequency response of our stable parameterized model as a function of densely sampled parameter values.

### 6.7.2 Multi port - Single parameter: Wilkinson Power Divider

In this example we consider a multiport wilkinson divider, shown in Figure 6-7. The standard wilkinson divider is designed on alumina substrate with the following specifications: characteristic impedance $Z_0 = 75\Omega$, substrate dielectric constant $\epsilon_r = 9.8$, substrate height $h = 125\mu m$, and metal thickness of $t = 4\mu m$. A natural parameter choice for this example is the center frequency $f_c$. A collection of non-parameterized
models are generated by simulating the structure using a full wave field solver [3,95] while varying the parameter from 15GHz to 25GHz, and using Algorithms 5 and 6 for fitting. The resulting models each have order 33 and are described by 11 poles.

A stable parameterized model with coefficients described by 4th order polynomials is generated using Algorithm 2 to fit to the previously generated individual models. The fitting procedure required just 1.86 seconds to solve all optimization problems in Algorithm 2.

To show the accuracy of our parameterized model, one component of the frequency response (|Z(3, 3)|) is plotted in Figure 6-8 (dashed red line) and compared to the response of individually fitted non-parameterized models, some of which were not used for fitting.

Lastly, Figure 6-9 plots the frequency response of our stable parameterized model as a function of densely sampled parameter values to show the smoothness of our
Figure 6-8: Comparison of magnitude of frequency responses, $|Z(3,3)|$, of wilkinson divider parameterized model (dotted lines) with the initial non-parameterized models (solid lines) for different parameter values. Some traces are from parameter values not used for fitting.
6.7.3 Multi port - Multi parameter: T-Type Attenuator

As our third example we consider a T-type attenuator. The purpose of this example is to show the full flexibility of our algorithm. This is a multiport, multivariate example where we consider two design parameters. We chose rational basis of different degrees to approximate different elements of the model. The frequency response samples were obtained by simulating the schematic in matlab. Individual non-parameterized models of order 14 and described by 7 poles each are generated and preprocessed for each value of the parameters, controlling attenuation $\lambda_1$ and resonant frequency $\lambda_2$, as described in Algorithm 5 and Algorithm 6.

A stable multivariate parameterized model is generated using Algorithm 2 using the rational function description in (6.15). In this example the numerator of the residues $\hat{R}(\lambda_1, \lambda_2)$, direct term factor $\hat{D}(\lambda_1, \lambda_2)$, and poles $\hat{A}(\lambda_1, \lambda_2)$ are described by polynomials of degree 5, 4, and 2 respectively, while the denominators are chosen as $4^{th}$ order polynomials for each term. The identification required 6.6 seconds to solve all optimization problems in Algorithm 2.

To show that our parameterized model is stable with respect to both parame-
ters, Figure 6-10 plots the densely sampled surface traced by real part of one of the dominant poles from our parameterized model as a function of $\lambda_1$ and $\lambda_2$.

![Figure 6-10: Surface traced by real part of one of the dominant poles from our stable multivariate parameterized model as a function of $\lambda_1$ and $\lambda_2$.](image)

An excellent match between initial non-parameterized models and final parameterized models can be observed in Figures 6-11 & 6-12.
Figure 6-11: Comparison of magnitude of frequency responses, $|Z(2, 1)|$, of attenuator multivariate parameterized model (dotted lines) with the initial non-parameterized models (solid lines). Fixed $\lambda_1$ varying $\lambda_2$. 
Algorithm 7 Complete Parameterized Stable model Fitting

1: Given collection of models generated using Algorithm 5 and ordered using Algorithm 6

2: if ENFORCE GLOBAL PASSIVITY then

3: Select basis functions for direct term factor \( \hat{V}(\lambda) \) and solve unconstrained optimization problem (6.21) for \( \hat{V}(\lambda) \)

4: Select basis for real part of the poles \( \Re \hat{A}(\lambda) \) and solve constrained optimization problem (6.18) for \( \Re \hat{A}(\lambda) \)

5: Select basis for imaginary parts of poles \( \Im \hat{A}(\lambda) \) and solve unconstrained optimization problem (6.13) for \( \Im \hat{A}(\lambda) \)

6: Select basis for real and imaginary parts of the residues \( \Re \hat{R}(\lambda) \) and \( \Im \hat{R}(\lambda) \) and solve the constrained optimization problem (6.21) for \( \Re \hat{R}(\lambda) \) and \( \Im \hat{R}(\lambda) \)

7: else

8: Select basis functions for direct term matrix \( \hat{D}(\lambda) \) and solve unconstrained optimization problem (6.13) for \( \hat{D}(\lambda) \)

9: Select basis for real part of the poles \( \Re \hat{A}(\lambda) \) and solve constrained optimization problem (6.18) for \( \Re \hat{A}(\lambda) \)

10: Select basis for real part of the residues \( \Re \hat{R}(\lambda) \) and solve unconstrained optimization problem (6.13) for \( \Re \hat{R}(\lambda) \)

11: Select basis for imaginary parts of poles \( \Im \hat{A}(\lambda) \) and residues \( \Im \hat{R}(\lambda) \) and solve unconstrained optimization problem (6.13) separately for \( \Im \hat{R}(\lambda) \) and \( \Im \hat{A}(\lambda) \)

12: end if

13: Define final parameterized model

\[
H(s, \lambda) = \sum_{k} \frac{\hat{R}_k(\lambda)}{s - \hat{a}_k(\lambda)} + \hat{D}(\lambda)
\]

where

\[
\hat{a}_k(\lambda) = \Re \hat{a}_k(\lambda) + j \Im \hat{a}_k(\lambda)
\]

\[
\hat{R}_k(\lambda) = \Re \hat{R}_k(\lambda) + j \Im \hat{R}_k(\lambda)
\]
Figure 6-12: Comparison of magnitude of frequency responses, $|Z(1, 1)|$, of attenuator multivariate parameterized model (dotted lines) with the initial non-parameterized models (solid lines). Fixed $\lambda_2$ varying $\lambda_1$. 
Chapter 7

Design of Decoupling Matrices for Parallel Transmission Arrays

In this chapter we present a framework to synthesize a multiport network from its frequency response or desired functionality. Although this is a well studied problem for single and two port networks, the problem of synthesizing a network from its given frequency response is highly challenging specially for multiport devices. In this thesis, we have explored the design and practical realization of decoupling matrices (or multiport matching networks) to decouple Radio Frequency (RF) phased arrays for Magnetic Resonant Imaging (MRI) applications.

7.1 Introduction

The additional transmit (Tx) degrees-of-freedom (DOF) of parallel transmit (pTx) coils allow mitigation of the non-uniform B1+ profile at high fields (i.e., 3T and higher) as well as reduction of the specific absorption rate [21, 26, 27, 37, 44, 45, 54, 55, 65, 70, 105, 109–111, 122, 131, 132]. Recently, Guerin et. al. [54, 55] have shown that increasing the number of independent Tx channels can dramatically improve the capability of the pTx pulse design to create uniform flip-angle excitations at much reduced local and global SAR compared to birdcage coil excitations. This is in agreement with studies by other groups that have observed important performance
improvement in terms of flip-angle uniformity when increasing the Tx channel count [24,75]. However, in [55], the authors have found that the B1+ and SAR performance improvement when increasing the channel count was accompanied with an important increase in power consumption. Although it may appear counter-intuitive that the reduction of SAR requires additional radio frequency (RF) power, this is simply due to the fact that the traditional metric for RF power is proportional to the square of the absolute value of the RF waveforms played on the different Tx channels. This definition of RF power is independent of the phase of the power amplifiers (RFPA) and does not take into account power that may be reflected or transmitted from one RFPA to another due to coupling. However this metric is still useful because it determines the total power output of the RFPA. In summary, the results presented in [54, 55] imply that to create highly uniform flip-angle distributions at low SAR requires more powerful RF amplifiers.

This problem of increased power consumption of pTx arrays with many channels is made dramatically worse if there is significant coupling between the Tx channels [55]. Indeed, in a coupled pTx array the power delivered to a channel is partially transmitted to other channels because of coupling. This coupled power is dissipated in the circulators and is lost for excitation, resulting in very low efficiency of the array as a whole [106]. Furthermore, coupling between the Tx channels increases the overlap between their B1+ maps, thus reducing the effective number of degrees of freedom of array for sculpting the flip-angle map [69,97,111,121,130,131].

Several methods have been proposed to decouple antenna arrays and magnetic resonance imaging coil arrays [66]. Conventional decoupling techniques such as partial overlap of Tx loops [101] and capacitive and inductive decoupling [35,73,124,127] are well suited to small arrays but may fail for coils with many Tx elements (i.e., > 8). This is because these approaches only allow decoupling of nearest-neighbor loops, but in large pTx arrays the most important coupling often occurs between next and third-neighbor elements [6]. In order to decouple distant neighbor channels, capacitive ladder networks have been proposed [67], but these networks are difficult to build and are not robust to variation in the load as they are highly sensitive to the specific
tuning and matching of the array.

Note that pre-correction of the digital waveforms by the inverse of the coupling matrix \([115,123]\) does not solve the lost power problem as it does not prevent power from going upstream in the Tx chain and into the circulators where it is lost for excitation (this strategy is not needed anyway as coupling between Tx channels is accurately reflected in their B1+ maps so that the pulse design algorithm already accounts for this effect). Recently, Stang et al., [113] proposed to use active elements in a feedback loop to decouple coil arrays by impedance synthesis. However the power dissipated in the active elements may reduce the overall power efficiency.

It was shown in \([8,9,42,129]\) that a passive network can be connected between an antenna array and its driving power amplifiers to achieve decoupling between array elements. Such techniques have been applied only to antenna arrays and have not been extended to the problem of matching and decoupling of multichannel array used for parallel transmission in magnetic resonance imaging. A similar approach was proposed in [76] for MRI receive coils. However, [76] focuses only on a restrictive class of degenerate solutions to the decoupling equation that may not be suitable for MRI transmit arrays.

There is clearly a need for a robust and scalable decoupling strategy of parallel transmission arrays. In this chapter, we propose an automated approach for the design and realization a high-power decoupling matrix, inserted between power amplifiers and a coupled parallel transmit array as shown in Figure 7-1. The decoupling matrix decouples all array elements, minimizes the power lost in the circulators and ensures maximum power transfer to the load. Our strategy robustly diagonalizes (in hardware) the impedance matrix of the coils using hybrid coupler networks connected in series. This is a known technique in the antenna design community [42]. In this work we improve on the standard lumped-element realization (which is composed of a very large number of lumped elements) of such circuits by using a compact distributed elements realization which is similar to a butler matrix [7].

We have previously shown in [83] and [84] that theoretically our decoupling matrix can achieve near-perfect decoupling between all channels. In this work we
demonstrate the procedure by designing and fabricating a decoupling matrix for a 2-channel coupled array used for parallel transmission in magnetic resonance imaging. The 2-channel design is provided as a proof of the concept. The methods presented in this thesis scale up to any arbitrary number of channels and can be readily applied to other coupled systems such as antenna arrays.

We summarize below the main contributions of this work:

- We propose an automated framework to decouple array elements using a decoupling matrix
- Our strategy is independent of geometrical configuration of the coils and the number of array channels
- The performance of our decoupling matrix is robust to load and component-value variations
- Our decoupling matrix is comprised only of reactive elements and distributed transmission line components. This implies a low insertion loss.
- Our decoupling matrix retains full degrees of freedom for transmit pulse design

7.2 Background

7.2.1 Decoupling Conditions

The impedance matrix of a coupled N-channel array is described by a dense symmetric complex matrix $Z_C = R_C + jX_C \in \mathbb{C}^{NN}$. The off-diagonal elements of $Z_C$, $z_{ij}(i,j)$ indicate the coupling between the elements $i$ and $j$ of the array. Because all sources are independent, the source impedance matrix is diagonal, with output impedance of the corresponding power amplifier as the diagonal elements (typically 50).

The mathematical condition for achieving full decoupling is that the impedance matrix of the load ($Z_{out}$, shown in Figure 7-1) seen by the power amplifiers is a
Figure 7-1: Block diagram of a parallel transmission RF array. A decoupling matrix is connected between the power amplifiers and the array. $V_1$ and $V_2$ are the voltage vectors at the array and the power amplifiers respectively. $Z_{OUT}$ is the impedance matrix of the load seen by power amplifiers. Perfect decoupling and matching is obtained when $Z_{OUT}$ is a diagonal matrix with diagonal entries equal to the characteristic impedance of the power amplifiers.

A diagonal matrix with matched (typically 50) impedance values to those of the RFPAs. For a 50Ω system, the desired $Z_{out}$ is given by

$$Z_{out} = \begin{bmatrix} 50 & \cdots & 50 \\ \vdots & \ddots & \vdots \\ 50 & \cdots & 50 \end{bmatrix}. \quad (7.1)$$

Furthermore, to have the ability to drive array elements individually, it is required that the voltage transformation introduced by the decoupling matrix is full rank. That is, $V_2 = TV_1$, where $T$ is a full rank matrix.

7.2.2 Interconnection of Circuit Blocks

Consider the interconnection of two circuit blocks as shown in Figure 7-2. Let $X_C$ describe the reactance matrix of the coupled array (on the right) while $S_A \in \mathbb{C}^{2N\times2N}$ describes the S-parameter matrix of the block $S_A$ at the operating frequency. If we assume that $S_A$ has the block form (7.2), where $A$ is an orthogonal matrix comprised
of the eigen-vectors of $X_C = A^\top D_X A$, then $X_{OUT} = AX_C A^\top = AA^\top D_X AA^\top = D_X$, here $D_X$ is a diagonal matrix composed of the eigenvalues of $X_C$ \cite{42,96,129}.

$$S_A = \begin{bmatrix} 0 & A \\ A^\top & 0 \end{bmatrix} \quad (7.2)$$

Figure 7-2: Interconnection of two circuit blocks. $X_C$ and $X_{OUT}$ are the reactance matrices at the coils and the output respectively. $S_A$ is the S-parameter matrix of the block $S_A$.

### 7.3 Design of a Decoupling Matrix

The decoupling condition described in Section 7.2.1 requires diagonalization of a fully dense complex matrix. In this work, we diagonalize the impedance matrix of a coupled array by a multiplication with its eigen-vectors implemented via hybrid couplers.

Figure 7-3 shows the schematic of the proposed decoupling matrix. The coupled array is described by a fully dense and complex impedance matrix, $Z_C = R_C + jX_C$. In the decoupling matrix, the first block $S_A$ diagonalizes the imaginary part of the complex impedance ($X_C$). This results into a diagonal reactance matrix which is eliminated by a series connection with opposite reactive elements. At this point, we are left with a dense real matrix which is diagonalized by the block $S_B$. Finally, we can use single port matching networks (such as quarter wavelength transformers)
to match the diagonal resistance values to the impedance of power amplifiers. Such decoupling can be divided into four stages. The following section provides details on each of these stages.

### 7.3.1 The Four Stages of decoupling

#### Stage 1

The first stage constitutes the $S_A$ block. We design the $S_A$ block such that the columns of the orthogonal matrix $A$ are the eigen-vectors of $X_C$. Consequently, this block diagonalizes the imaginary part, $X_C$. Let us denote the intermediate impedance at the output of $S_A$ as $Z_I$, then

$$Z_I = A(Z_C)A^T = A(R_C + jX_C)A^T$$
$$= AR_CA^T + jAX_CA^T$$
$$= R_I + jD_X,$$  \hspace{1cm} (7.3)

where $D_X$ is a diagonal matrix with eigen values of $X_C$.

#### Stage 2

Stage 2 eliminates the diagonal matrix $D_X$. We can achieve this simply by a series connection of reactive elements of opposite signs at the operating (Larmor) frequency. Let us denote the output impedance of second stage as $Z_{II}$, then

$$Z_{II} = R_I + jD_X - jD_X$$
$$= R_I$$  \hspace{1cm} (7.4)
Stage 3

The impedance presented at the input of the stage 3 is a symmetric dense real matrix $R_I$. Similar to Stage 1, we design a block $S_B$ whose S-parameter matrix at the operating frequency is given by

$$
S_B = \begin{bmatrix}
0 & B \\
B^\top & 0
\end{bmatrix}
$$

(7.5)

such that the columns of the orthogonal matrix $B \in \mathcal{R}^{NN}$ are the eigen vectors of $R_I$. As a result, this block diagonalizes $R_I$. Let us denote the intermediate impedance at the output of stage 3 as $Z_{III}$, then

$$
Z_{III} = B(R_I)B^\top = D_R.
$$

(7.6)

Here $D_R$ is a diagonal matrix with eigen values of $R_I$.

Stage 4

The impedance matrix presented at input to stage 4 is a diagonal ‘resistance’ matrix. At this point the coils are completely decoupled, however they are not matched. The task for stage 4 is to match this diagonal matrix to the impedance of power amplifiers $Z_0$, typically $Z_0 = 50\Omega$.

To accomplish this task, we can use any of the well established single port matching techniques to individually match each of the diagonal entry in $R_I$ to the output impedance of the corresponding power amplifier. Some of these techniques include quarter wavelength transformers, $\top$, $\pi$ and $L$-type matching networks. The details can be found in [78,99].
7.3.2 A Full Rank Transformation

We can show that $V_2 = BAV_1$. Here $A$ and $B$ are the off diagonal block matrices of the S-parameter matrix of $S_A$ and $S_B$ respectively at the operating frequency. Since $A$ and $B$ are orthogonal by construction, $T = BA$ defines a full rank transformation.

A proof is given in the Appendix D.

7.3.3 Realization of the $S_{A/B}$

We observed that the S-parameter matrix of a 2-channel (4-ports) 180-degrees hybrid coupler with unequal divide ratio has the same form required to implement the blocks $S_{A/B}$ in (7.7) (The $-j$ being multiplied by the matrix in (7.7) can be removed by a series connection of transmission lines with length equal to 270 degrees at the input ports). Note that this is a rotation matrix. A series of such rotations matrices, known as Givens rotations [43], are computed as part of a standard algorithm to diagonalize (i.e. computing its singular value decomposition) matrices [43].
Figure 7-4: Layout of a 180 – degrees hybrid coupler

\[ S_{180-\text{hybrid}}(\theta) = -j \begin{bmatrix} 0 & 0 & \cos\theta & \sin\theta \\ 0 & 0 & -\sin\theta & \cos\theta \\ \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \end{bmatrix} \]  \quad (7.7)

where

\[ A(\text{or } B) = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \]  \quad (7.8)

Here the angle \( \theta \) indicates the rotation angle and is computed from the standard singular value decomposition algorithm via Givens rotations [43]. This rotation is implemented by the ratio of the impedances of the hybrid coupler. The impedances \( (Z_{0A}, Z_{0B}) \) of various sections of a hybrid coupler are shown in Figure 7-4 and are related by the following expressions.

\[ Z_1 = Z_0 \sqrt{\frac{1 + \gamma}{\gamma}} \]  \quad (7.9)

\[ Z_2 = Z_0 \sqrt{1 + \gamma} \]  \quad (7.10)

\[ \gamma = \left(\frac{\sin\theta}{\cos\theta}\right)^2 \]  \quad (7.11)

Figure 7-4 shows the layout of a 2 × 2 hybrid coupler (4 ports).
Figure 7-5: Interconnection of $2 \times 2$ hybrid couplers to realize a 3-channel $S_A$ block

Note that a single $2 \times 2$ coupler can implement only a 2-channel $S_{A/B}$ block. In order to realize $S_{A/B}$ blocks with $N$ - *channels*, $0.5N(N - 1)$ of these $2 \times 2$ couplers can be interconnected, as dictated by the Givens rotations [43]. For example for $N = 3$, we require three ($0.5N(N - 1) = 3$) $2 \times 2$ couplers. The connections are shown in Figure 7-5.

### 7.3.4 Robustness Analysis

In this section we show that the decoupling matrix is robust to variations in the load and in the values of lumped components. We show that a small bounded perturbation ($\Delta Z_C$) in the load ($Z_C$) will result into a small bounded perturbation ($\Delta Z_{OUT}$) in the output ($Z_{OUT}$). Let $||\Delta Z_C||_2 < \epsilon$, we show that $||\Delta Z_{OUT}||_2 < \epsilon$. Mathematically our decoupling matrix can be expressed as:

$$Z_{OUT} = B(AZ_CA^T - jD_X)B^T$$  \hspace{1cm} (7.12)

Then

$$Z_{OUT} + \Delta Z_{OUT} = B(A(Z_C + \Delta Z_C)A^T - jD_X)B^T$$  \hspace{1cm} (7.13)$$

$$\Rightarrow \Delta Z_{OUT} = BA\Delta Z_CA^TB^T$$  \hspace{1cm} (7.14)$$

$$||\Delta Z_{C}||_2 < \epsilon \Rightarrow ||\Delta Z_{OUT}||_2 < \epsilon$$  \hspace{1cm} (7.15)
This is because both $A$ and $B$ are unitary matrices and the two-norm (or the spectral-norm) of a matrix is unitarily invariant. Following similar reasoning we can show that

$$||\Delta D_X||_2 < \epsilon \Rightarrow ||\Delta Z_{OUT}||_2 < \epsilon$$ (7.16)

where $\Delta D_X$ is a matrix describing perturbation in the lumped component values. A similar procedure may be followed to check sensitivity to variations in the $A$ and $B$ matrices which are implemented via hybrid couplers. This shows that the decoupling matrix is robust to variations load and component-values.

### 7.3.5 Required Components

In order to design a decoupling matrix to decouple $N$-channel array, we need

- $N(N - 1)$ unequal divide hybrid couplers. Note number can be reduced for a relaxed decoupling requirement. E.g. in the case when two elements in the array are already sufficiently decoupled we do not need to introduce a coupler between them.

- $N$ reactive elements

- $N$ single port matching networks

### 7.3.6 Advantages of our Decoupling Framework

The proposed framework is generic, automatic and scalable. This means that our decoupling strategy is independent of array’s geometrical configuration and the number of channels. The decoupling is achieved at the cost of only a small insertion loss. The computed solution is robust to component and load variations.
Figure 7-6: Picture of a coupled parallel transmit array to be decoupled. The array is loaded with a head phantom.

![Diagram of S-parameter matrix of the coupled array at the operating frequency.](image)

\[
\begin{align*}
S_{1,1} &= -8 \text{dB} \\
S_{1,2} &= -7 \text{dB} \\
S_{2,1} &= -7 \text{dB} \\
S_{2,2} &= -8 \text{dB}
\end{align*}
\]

Figure 7-7: S-parameter matrix of the coupled array at the operating frequency.

7.4 Results

7.4.1 A 2-channel example with hardware implementation

We demonstrate our decoupling framework by designing and fabricating a decoupling matrix for a 2-channel array shown in Figure 7-6. The coils are individually tuned and matched at $297.2 \text{MHz}$. The S-parameter matrix of the array (without a decoupling matrix) show significant coupling between the elements as shown in Figures 7-7 and 7-8.

We implemented our decoupling matrix using a hybrid coupler shown in Figure 7-
Figure 7-8: Frequency dependence of the S-parameters of the coupled array. Blue curves indicate the diagonal, while red curves indicate off-diagonal elements.

Figure 7-9: Layout of the hybrid coupler used in our decoupling matrix

9. The hybrid coupler is designed at 297.2 MHz and fabricated on an FR-4 substrate. The insertion loss introduced by the decoupling matrix was less than 0.3 dB. The level of coupling and matching achieved by our decoupling matrix is less than −40 dB, as shown in Figures 7-10 and 7-11.

We tested the performance of our decoupling matrix in the presence of varying load, i.e. the head phantom size as shown in Figure 7-12. The measurements show that despite changing the head size from one extreme (child) to the other (adult, male) the decoupling matrix still performed well, and the worst case decoupling was still better than −28 dB.
Figure 7-10: S-parameter matrix of the decoupled array at the operating frequency using our decoupling matrix

\[
\begin{array}{c|c}
S_{1,1} = -41dB & S_{1,2} = -42dB \\
S_{2,1} = -42dB & S_{2,2} = -42dB \\
\end{array}
\]

Figure 7-11: Frequency dependence of the S-parameters of the decoupled array.

Figure 7-12: Frequency dependent S-parameters of the array with a decoupling matrix for varying phantom size. (a) Nominal phantom, decoupling level is less than \(-40dB\) (b) Larger phantom, decoupling level is less than \(-35dB\) (c) Smaller phantom, decoupling level is less than \(-28dB\).
7.4.2 A 4-channel example (simulations only)

In this example we design a decoupling matrix to decouple a 4-channel transmit array. The performance was tested in simulations only. Note that even though we need \(N(N - 1)\) hybrid couplers to decouple a 4-channel array, we can actually trade-off the decoupling level by using a smaller number of couplers, e.g. by not introducing a coupler (implementing a Givens rotation) for two array elements that are already sufficiently decoupled. In this example, we have used only 8 couplers instead of 12 = \(N(N - 1)\), while still having worst case decoupling better than \(-25dB\). Figure 7-13 (a) and (b) shows the frequency dependent S-parameter matrix of the array without and with a decoupling matrix at the operating Larmor frequency of \(297.2MHz\). Figure 7-13 (c) shows the results of a sensitivity analysis. We varied the values of the lumped components used in the design with in \(\pm 10\%\) of the nominal component values, and found that the worst case decoupling level was still within an acceptable range of \(-25dB\).

7.5 Discussion

We have presented a framework to design a decoupling and matching network, a decoupling matrix, for parallel transmit arrays operating at high frequencies. To our knowledge, this is the first presentation of an automated procedure that decouples pTx arrays with an arbitrary number of transmit channels. Key advantages of our proposed framework are that it is generic, automatic and scalable. This means that the proposed decoupling strategy is independent of array’s geometrical configuration and the number of channels. Such a matrix could be used to decouple pTx coils with many channels (i.e., \(> 16\)), which are difficult to decouple properly using existing methods but have been shown to be beneficial for SAR reduction and manipulation of the transverse magnetization signal.

The decoupling is achieved at the cost of only a small insertion loss. This insertion loss is implementation dependent. A distributed element approach is more suited for higher frequencies (for MRI machines at \(7T\) or higher) where the hybrid couplers
Figure 7-13: Simulated plots for a 4 – channel example (a) S-parameter matrix of the coupled array at the operating frequency. (b) Frequency dependence of the S-parameters of the coupled array. (c) Sensitivity of the worst case decoupling with respect to lumped component variations.
can easily be implemented via transmission lines. The insertion loss in this case will depend on the operating frequency and the dielectric loss tangent of the substrate of the transmission line. Since our decoupling matrix is similar to a Butler matrix in construction, they have similar insertion loss. Butler matrices for MRI applications are well studied [128]. A possible downside of using a distributed elements realization is the physical size. As mentioned before our strategy requires at most $N(N - 1)$ unequal divide hybrid couplers, as a result the overall physical size of the decoupling matrix could be large. A lumped element based realization is also possible [10], however since it requires a large number of lumped elements to realize a hybrid coupler, the insertion loss may increase.

A decoupling strategy is useful if its performance is robust to load and component variations. We have shown both theoretically and experimentally that the decoupling matrix is robust to component and load variations. As shown in the results section, we have tested our implemented decoupling matrix under different loading conditions and found that its performance is satisfactory with load variations. Similarly we performed a sensitivity analysis by varying the values of the lumped components used in the design with in $\pm 10\%$ of the nominal component values, and found that the worst case decoupling level was with in an acceptable range of $-25dB$.

Finally our decoupling matrix creates orthogonal modes of excitations of the Tx channels that, although a priori different from the coil natural basis. This is supported by the fact that the voltage transformation introduced by the decoupling matrix is full rank (i.e., no loss of Tx degrees of freedom).

In the future, we plan on extending the implementation to more than two channels. Since the mathematical principals of decoupling are similar for receive coils, we would like to explore how such a decoupling matrix can be used to decouple receive coil arrays.
Chapter 8

Conclusion

In this thesis we have presented various highly efficient algorithms to identify individual and parameterized multiport passive models from frequency domain transfer matrix data samples. The algorithms are based on convex relaxations of the original non-convex problems. In the first algorithm, we identify a collection of first and second order networks to model individual non-parameterized passive blocks. Passivity of the overall model is guaranteed by enforcing passivity on the building blocks. The problem is solved in two steps. In the first step we identify a set of common poles for the transfer matrix. In the second step we use the common set of stable poles from step one and identify residue matrices which minimize the mismatch between the model and the given data, and simultaneously conforming to passivity conditions. Several examples are presented which advocate the speed and efficiency of the proposed algorithm. In these examples we have tested passivity of the identified models by verifying the absence of purely imaginary eigenvalues of the associated Hamiltonian matrix. The identified models are interfaced with commercial circuit simulators and used for time domain simulations of complete architectures including a LINC power amplifier where multiport passive model was identified for the passive combining network. The proposed algorithm is compared with existing algorithms based on optimization framework. The comparisons show that our algorithm achieved a speed-up of $40\times$ for some examples while for other examples we generated a highly accurate model in decent amount of time whereas the alternative algorithm ran out
of memory and failed to generate a model.

In the second algorithm for the identification of individual non-parameterized passive models, we employ non-smooth localization-based optimization methods to solve the formulation of passivity enforcement presented in [22]. We guarantee that the solution can be found up to any prescribed accuracy within a finite number of iterations. These algorithms are memory efficient and demonstrate significant improvements both in terms of time and memory when compared with similar passive modeling algorithms.

We have also presented an algorithm to identify globally stable and passive parameterized multiport models. In this algorithm we combine individual stable and passive non-parameterized models to develop a closed form parameterized model. The final parameterized model conforms to passivity conditions during identification and comes with apriori global passivity certificates in the continuous parameter range of interest. In several examples we have verified that the models generated by our approach can be safely instantiated for any parameter value and always result in a stable and passive system, as opposed to all existing interpolation approaches. We have also shown that our fitting approach only requires few seconds to identify practical passive circuit components, having formulated the problem as an efficient convex optimization program.

In the last part of this thesis we have presented an automated strategy to realize passive networks to match and decouple coupled multichannel arrays in magnetic resonance imaging (MRI) applications. Our proposed framework is a generic, automatic and scalable technique to design decoupling matrices for pTx arrays with many channels. The proposed decoupling strategy is robust to load variations and is independent of coils’ geometrical configuration or number of channels. We design a decoupling matrix for a 2-channel array. The results show that our decoupling matrix achieves near to ideal decoupling. The 2-channel instantiation is for demonstration and evaluation purposes. The methods presented in this thesis readily scale to any arbitrary number of channels.
Appendix A

Semidefinite Programming

In the standard form of a semidefinite program a linear cost function is minimized subject to linear matrix inequalities.

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad F_1 x_1 + F_2 x_2 + \cdots + F_n x_n - F_0 \succeq 0
\end{align*}
\] (A.1)

describes the standard form of a semidefinite program, here all of the matrices \( F_0, F_1, \ldots, F_n \in S^k \), here \( S^k \) indicates set of symmetric matrices of order \( k \times k \). The problems that we have described in this thesis are not in the standard form, however they can easily be transformed into the standard representation. Some of the relevant transformations are described in the following sections.

A.1 Minimizing Quadratic Function

Suppose we wish to minimize a quadratic function of the form \( \| Ax - b \|^2 \). We can cast this minimization problem into an equivalent semidefinite program as
\[
\min_{x} \|Ax - b\|^2 \quad \equiv \min_{t,x} \quad t \\
\text{subject to} \quad (Ax - b)^T (Ax - b) \leq t^2 \\
\equiv \min_{t,x} \quad t \\
\text{subject to} \quad t^2 - (Ax - b)^T (Ax - b) \geq 0 \\
\equiv \min_{t,x} \quad t \\
\text{subject to} \quad t - (Ax - b)^T (tI)^{-1} (Ax - b) \geq 0 \\
\equiv \min_{t,x} \quad t \\
\text{subject to} \quad \begin{bmatrix} tI & (Ax - b) \\ (Ax - b)^T & t \end{bmatrix} \succeq 0 \quad (A.2)
\]

Here the last step resulted by applying Schur Complement. The final constraint (A.2) can be transformed into standard SDP constraint as:

\[
\begin{bmatrix} tI & (Ax - b) \\ (Ax - b)^T & t \end{bmatrix} \succeq 0 \quad \Rightarrow \quad \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix} t + \sum_{i=1}^{n} \begin{bmatrix} 0 & A_i \\ A_i^T & 0 \end{bmatrix} x_i - \begin{bmatrix} 0 & b \\ b^T & 0 \end{bmatrix} \succeq 0 \\
(A.3)
\]

here \(A_i\) indicates the \(i\)th column of matrix \(A\). The final equivalent semidefinite program in standard form , described in (A.4) can be solved efficiently using any SDP solver such as [2,100].

\[
\min_{x} \|Ax - b\|^2 \quad \equiv \min_{t,x} \quad t \\
\text{subject to} \quad \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix} t + \sum_{i=1}^{n} \begin{bmatrix} 0 & A_i \\ A_i^T & 0 \end{bmatrix} x_i - \begin{bmatrix} 0 & b \\ b^T & 0 \end{bmatrix} \succeq 0 \\
(A.4)
\]
A.2 Implementing Linear Matrix Inequalities

Suppose we wish to enforce the following linear matrix inequality as an SDP constraint

\[
\begin{bmatrix}
    x_1 & x_2 \\
    x_2 & x_3 \\
\end{bmatrix} + \begin{bmatrix}
    x_4 & x_5 \\
    x_5 & x_6 \\
\end{bmatrix} \succeq 0
\]  

\[\text{(A.5)}\]

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

\[
\begin{bmatrix}
    c_1 & 0 \\
    0 & c_1 \\
\end{bmatrix} x_1 + \begin{bmatrix}
    0 & c_1 \\
    c_1 & 0 \\
\end{bmatrix} x_2 + \begin{bmatrix}
    0 & 0 \\
    0 & c_1 \\
\end{bmatrix} x_3 + \begin{bmatrix}
    c_2 & 0 \\
    0 & c_2 \\
\end{bmatrix} x_4 + \begin{bmatrix}
    0 & c_2 \\
    c_2 & 0 \\
\end{bmatrix} x_5 + \begin{bmatrix}
    0 & 0 \\
    0 & c_2 \\
\end{bmatrix} x_6 \succeq 0
\]  

\[\text{(A.6)}\]
Appendix B

Definitions

In this appendix 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 [17], and is summarized here as a convenience for the readers.

B.0.1 Gradients and Subgradients

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

$$f(y) \geq f(x) + g^T(y - x).$$  \hfill (B.1)

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

$$\partial f(x) = \{g : g \text{ is a subgradient of } f \text{ at } x\}$$  \hfill (B.2)

is called the subdifferential of $f$ at $x$. 

155
B.0.2 Hyperplanes

A hyperplane is a set of the form

\[ \{x \mid a^T x = b\} \quad (B.3) \]

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 B-1. 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.

B.0.3 Ellipsoids

An ellipsoid \( \varepsilon \) is described by its center \( x_c \in \mathbb{R}^n \) and the matrix \( P \).

\[ \varepsilon(x_c, P) = \{x \mid (x - x_c)^T P^{-1} (x - x_c) \leq 1\}, \quad (B.4) \]

where \( P = P^T > 0 \) 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 \( \varepsilon \).
B.0.4 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

\[ \mathcal{P} = \{ x | a^T_j x \leq b_j, j = 1, ..., 5 \} \]

A polyhedron is thus the intersection of a finite number of halfspaces as shown in Figure B-2.
Appendix C

Derivations

C.1 Derivation of the weighted objective function

In this section we derive (5.13) starting from (5.6)

\[
\begin{align*}
\minimize_{C_P} ||C_P G^T||_F & \equiv \minimize_{C_P} ||\text{vec}(C_P G^T)||_2 \\
& \equiv \minimize_{C_P} ||(G \otimes I)\text{vec}(C_P)||_2 \\
& \equiv \minimize_x ||(G \otimes I)x||_2 \\
& \equiv \minimize_x ||Wx||_2 \\
& \equiv \minimize_x f(x)
\end{align*}
\]

C.2 Derivation of (5.16)

In this section we derive (5.16). The left hand side of (5.16) is equivalent to the following convex optimization problem

\[
\begin{align*}
\minimize_{x} g^T x \quad \text{s.t.} \quad x^T P^{-1} x \leq 1
\end{align*}
\]
The Lagrangian, $L(x, \lambda)$, and the dual function, $f_D(\lambda)$, of (C.1) are given by

$$L(x, \lambda) = g^T x + \lambda x^T P^{-1} x - \lambda$$

$$f_D(\lambda) = \inf_x L(x, \lambda)$$

$$f_D(\lambda) = -\frac{1}{4\lambda} g^T P g - \lambda \quad \text{(C.2)}$$

Since the problem (C.1) is convex and $P \succ 0$, it satisfies Slater’s conditions [19]. This implies that strong duality holds. Hence we compute the global optimal solution to (C.1) by solving the its dual problem which is given by

$$\max_\lambda -\frac{1}{4\lambda} g^T P g - \lambda \quad \text{s.t.} \quad \lambda \leq 0 \quad \text{(C.3)}$$

The global optimal solution ($\lambda^*$) for (C.3) is given by

$$\lambda^* = \frac{1}{2} \sqrt{g^T P g} \quad \text{(C.4)}$$

Substituting (C.4) in (C.2) gives us the right hand side of (5.16), which is

$$f_D(\lambda^*) = -\sqrt{g^T P g} \quad \text{(C.5)}$$
Appendix D

The Voltage Transformation by our Decoupling Matrix is Full Rank

We compute the voltage transformation relation between the voltages $V_1$ and $V_2$ and show that the transformation is indeed full rank. We note that the above blocks are connected in cascade, i.e. the output of one block is connected to the input of the next. For such a connection the best choice to represent the blocks is via the ABCD-parameters, given in (D.1).

\[
\begin{bmatrix}
    V_2 \\
    I_2
\end{bmatrix} =
\begin{bmatrix}
    a & c \\
    b & d
\end{bmatrix}
\begin{bmatrix}
    V_1 \\
    -I_1
\end{bmatrix}
\]

This is because we can compute the ABCD-parameters of the entire chain simply by multiplication of the ABCD-parameter matrices of the individual blocks. Substituting the individual ABCD-parameter matrices in (D.1).

\[
\begin{bmatrix}
    V_2 \\
    I_2
\end{bmatrix} =
\begin{bmatrix}
    I & jD_1 \\
    jD_2 & I
\end{bmatrix}
\begin{bmatrix}
    B & 0 \\
    0 & B
\end{bmatrix}
\begin{bmatrix}
    I & -jD_x \\
    0 & I
\end{bmatrix}
\begin{bmatrix}
    A & 0 \\
    0 & A
\end{bmatrix}
\begin{bmatrix}
    V_1 \\
    -I_1
\end{bmatrix}
\]

(D.2)
\[
\begin{bmatrix}
V_2 \\
I_2
\end{bmatrix} = \begin{bmatrix}
BA & -jBD_XA + jD_1BA \\
jD_2BA & D_2BD_XA + BA
\end{bmatrix}
\begin{bmatrix}
V_1 \\
-I_1
\end{bmatrix}
\] (D.3)

Where \(D_1\) and \(D_2\) are diagonal matrices. Under no load condition, \(I_1 = 0\), this implies

\[
V_2 = BAV_1 = TV_1
\] (D.4)

Since both \(B\) and \(A\) matrices are orthogonal by construction, \(T\) is a full-rank matrix. Hence the voltage transformation introduced by the decoupling matrix is full rank.
Bibliography


[85] Zohaib Mahmood, Giuseppe C. Calafiore, Stefano Grivet-Talocia, and Luca Daniel. Optimal passivity enforcement of state-space models via localization


173