Efficient Approach for Order Selection of Projection-based Model Order Reduction

by

Gnanesh Baggu

Thesis submitted to the University of Ottawa

in partial fulfillment of the requirements for the

Master of Applied Science degree

in

Electrical and Computer Engineering

School of Electrical Engineering and Computer Science

Faculty of Engineering

University of Ottawa

Abstract

The present thrust in the electronics industry towards integrating multiple functions on a single chip while operating at very high frequencies has highlighted the need for efficient Electronic Design Automation (EDA) tools to shorten the design cycle and capture market windows. However, the increasing complexity in modern circuit design has made simulation a computationally cumbersome task.

The notion of model order reduction has emerged as an effective tool to address this difficulty. Typically, there are numerous approaches and several issues involved in the implementation of model-order reduction techniques. Among the important ones of those issues is the problem of determining a suitable order (or size) for the reduced system. An optimal order would be the minimal order that enables the reduced system to capture the behavior of the original (more complex and larger) system up to a user-defined frequency.

The contribution presented in this thesis describes a new approach aimed at determining the order of the reduced system. The proposed approach is based on approximating the impulse response of the original system in the time-domain. The core methodology in obtaining that approximation is based on numerically inverting the Laplace-domain of the representation of the impulse response from the complex-domain (s-domain) into the time-domain.

The main advantage of the proposed approach is that it allows the order selection algorithm to operate directly on the *time-domain* form of the impulse response. It is well-known that numerically generating the impulse response in the time-domain is very difficult and its not impossible, since it requires driving the original network with the Dirac-delta function, which is a mathematical abstraction rather than a concrete waveform that can be implemented on a digital computer. However, such a difficulty is avoided in the proposed approach since it uses the Laplace-domain image of the impulse response to obtain its time-domain representation.

The numerical simulations presented in the thesis demonstrate that using the time-domain waveform of the impulse response, computed using the proposed approach and properly filtered with a Butterworth filter, guides the order selection algorithm to select a smaller order, i.e., the reduced system becomes more compact in size. The phrase "smaller or more compact" in this context refers to the comparison with existing techniques currently in use, which seek to generate some form of time-domain approximations for the impulse response through driving the original network with pulse-shaped function (e.g., Gaussian pulse).

Dedication

I would like to dedicate this thesis wholeheartedly to my parents and my professor Dr. Emad Gad, who have been my source of inspiration and gave me strength in the thoughts of giving up, who continually provide their moral, emotional and financial support.

I also dedicate this to my loving sisters, brothers, relatives and friends who shared their words of advice and enouragement throughtout this work.

Acknowledgements

First and foremost, i would like to express my sincere gratitude to my thesis supervisor, Professor Emad Gad . Without his help, this thesis would not have come to this completion. I am grateful to him for his invaluable advice in guiding this research as well as his care and attention towards me. It was truly an invaluable experience working with him and I can never forget the valuable lessons of life learned from him. Thank you so much for being my supervisor, sir.

I am also sincerely grateful to my co-supervisor Professor Michel Nakhla, for his guidance, suggestions and support throughout the thesis. I wish to thank him for his encouragement and kindness to me. I am lucky to share this work with a person of his stature.

I am very thankful to Professor, Behzad Nouri, for his valuable suggestions and insights in this work. I wish to thank him for motivating and feeling the concern on me.

I cannot forget the valuable words of advice and knowledge provided by Professor, Pavan Gunupudi. His support laid the foundation for my path towards this work.

I would like to extend my thanks to my classmates and seniors with whom I spent very good time understanding subject presented in the class. Specailly i would like to thank Sravani, Abhishek, Naveen, Reshma and Kimi for their time spent with me and illuminating my life and adding more dimension to the aspect of knowledge.

Last but not least, i give special thanks to my family for their blessings, support, encouragement and unconditional love. I cannot express my admiration for their kindness and sacrifices. This is all for them.

Thank you all sincerely,

Table of Contents

List of Tables

List of Figures

Nomenclature

Abbreviations

AWE Asymptotic Waveform Evaluation

BIBO Bounded Input Bounded Output

CAD Computer Aided Design

DAE Differential-Algebraic Equations

FFT Fast Fourier Transform

FNN False Nearest Neighbors

LTI Linear Time Invariant

MATLAB Matrix Laboratory

MIMO Multiple Input Multiple Output

MNA Modified Nodal Analysis

MOR Model Order Reduction

NILT Numerical Inversion of Laplace Transform

POD Proper Orthogonal Decomposition

PRIMA Passive Reduced Order Interconnect macro modelling Algorithm

PVL Padé via Lanczos

SVD Singular Value Decomposition

TBR Truncated Balanced State Space Representation

TL Transmission Line

VLSI Very Large Scale Intergrated Circuit

Mathematical Symbols

$\dot{\mathrm{x}}(t)$ The derivative of the vector $\mathrm{x}(t)$

$\hat{\mathbf{A}}$ The size reduced of the matrix $\mathbf{A}$

$\mathbb{C}_+$ Positive Complex Plane

$\mathbb{R}^{N\times N}$ A real matrix of size $N \times N$

$\mathbf{A}^{-1}$ Inverse of the real matrix $\mathbf{A}$

$\mathbf{A}^T$ Transpose of the real matrix $\mathbf{A}$

$\mathbf{I}_m$ An $m \times m$ identity matrix

$d_n(t_i, t_j)$ Distance between two points on the trajectory

$\mathbf{A}$ A real matrix $\mathbf{A}$

$\mathrm{colsp}(\mathbf{Q})$ Column space of the matrix $\mathbf{Q}$

Chapter 1

Introduction

1.1 Introduction

The increasing sophistication of modern consumer electronics and the high demand for computational resources coupled with the prevalence of the internet-based applications in all walks of life have produced immensely complex electronic designs. Managing the complexity of these designs has become one of the main thursts of developing *Computer - Aided Design (CAD)* tools. CAD is, therefore, an essential component in the design cycle of all electronic circuits.

The idea of representing a large complex system with a smaller system referred to as *Model - Order Reduction (MOR)*, has received a sustained and renewed attention from the CAD community for the best part of the last two decades. The idea in MOR is to capture a system described by a model of differential equations, with another system. That is also represented by differential equations, albeit with a much smaller order.

For the sake of later references, I present here the form of the system of differential equations of the original system that is the target of MOR. This system typically arises from the Modified Nodal Analysis approach to circuit formulation and takes the form,

$$\mathbf{C}\frac{d}{dt}\mathbf{x}(t) \; + \; \mathbf{G}\; \mathbf{x}(t) \; = \; \mathbf{B}\; \mathbf{u}(t) \tag{1.1}$$

where $\mathbf{C}$ is matrix that collects the contributions or the stamps of different memory elements such as capacitors, inductors, $\mathbf{G}$ is a matrix that plays the same role for the memoryless elements, e.g, resistors, dependent sources, etc. $\mathbf{x}(t)$ is a vector that comprises all the circuit variables at any time t, and $\mathbf{u}(t)$ is a vector that represents all the independent stimuli in the circuit. The sizes of these matrices and vectors are such that $\mathbf{C}$ and $\mathbf{G} \in \mathbb{R}^{N\times N}$, and $\mathbf{x}(t) \in \mathbb{R}^{N\times 1}$ where N is the size of the system. The size of the input vector $\mathbf{u}(t) \in \mathbb{R}^{\mathrm{nx1}}$ where $n \; \ll \; N$ is the number of independent sources. The goal of MOR is to replace the system with another system, in the form

$$\hat{\mathbf{C}}\frac{d}{dt}\hat{\mathbf{x}}(t) \; + \; \hat{\mathbf{G}}\hat{\mathbf{x}}(t) \; = \; \hat{\mathbf{B}}\mathbf{u}(t) \tag{1.2}$$

where the size of the reduced system matrices are $\hat{\mathbf{C}}$ and $\hat{\mathbf{G}} \in \mathbb{R}^{q\times q}$, and $\hat{\mathbf{x}}(t) \in \mathbb{R}^{q\times 1}$, where q is the size of the new system and $q \; \ll \; N$.

1.2 Objectives

The primary objectives of this thesis are

(1) Consider the problem of MOR.

(2) Describe a general background to the different approaches used to derive the reduced order model.

(3) Focus on a particular approach used in MOR, namely, the approach based on Krylov-subspace projections.

(4) Examine the various steps used in that approach.

(5) Propose a new technique to carry out one of those steps, namely the step of determining the best order that is appropriate for the reduced model to have.

1.3 Contributions

The main contribution of this thesis is the implementation of a new process aimed at achieving the fifth objective listed above. More precisely, the contribution of this thesis is to propose and implement a new procedure to determine or select the best order of the reduced model generated using the Krylov Subspace method. The proposed procedure is inspired by the recent revival of interest in the concept of Numerical Inversion of the Laplace Transform, NILT, which has been presented in 1971 for the numerical simulation of the transient response of Linear circuits. This thesis puts the NILT framework to serve as the main engine used in determining the best order for the Krylov-based model to have.

The thesis demonstrates that the proposed procedure presents a higher degree of accuracy at a lower computational cost compared with the existing approaches that achieve the same goal. In particular, the numerical simulations presented in the thesis demonstrate that using the time-domain waveform of the impulse response, computed using the proposed approach based on NILT and properly filtered with a Butterworth filter, guides the order selection algorithm to select a smaller order, i.e., the reduced system becomes more compact in size. The phrase "smaller or more compact" in this context refers to the comparison with existing techniques currently in use, which seek to generate some form of time-domain approximations for the impulse response through driving the original network with pulse-shaped function (e.g., Gaussian pulse).

1.4 Organization of the thesis

Chapter 2.1 provides necessary information about various model order reduction techniques. It also elaborates on the projection-based model order reduction techniques which are used in the implementation of this thesis. In Chapter 3, the order selection algorithm, which represents the basis of this thesis is reviewed . Chapter 4 presents the necessary background for NILT and investigates the problem about using Gaussian pulse as input for the network. It also presents the main contribution of the thesis, showing its advantage over the previously used method. Chapter 5 presents the successful implementation of the proposed method with numerical results. A conclusion and suggestion for future work are presented in Chapter 6.

Chapter 2

Model Order Reduction

The model order reduction has many definitions based on the context one prefers to describe it. But, the main idea behind all of them is that, given a large dynamical system, a small system is to be found which approximates the input-output behavior of the original system. It was originally introduced in the area of control theory, but due to its feasibility, it has been carried over to other engineering branches of which VLSI circuit design is the subject of focus in this thesis. This chapter provides an introduction and explanation of the main concepts of model order reduction and the literature covering the MOR techniques.

2.1 Introduction

Models of modern dynamical systems have become so complicated, such that an approximation must be used to achieve simpler models. Computers are widely used for modeling of these real-world systems.Hence, simulations or, more generally, computational science has become an important method for understanding the behavior of such systems. Besides the classical disciplines grounded in either theoretical or experimental sciences, computational science has grown to be a pillar in modern scientific research. It is the goal of this chapter to review a central part of scientific computation, namely the simulation of

complex systems through the concept of model-order reduction.

The review of MOR presented in this chapter is not meant to be an in-depth coverage of the various approaches, but mainly to insert pointers to the major developments in that field.

The contribution presented in this thesis bears on a specific approach to MOR, namely the projection-based approach. The reader may notice that the treatment of this approach carries significantly more depth compared to others. This is indeed an in-depth explanation that will serve to illuminate the main purpose and place of the contribution.

The background on the projection-based MOR will be presented in section 2.6. Section 2.2 discusses the feature that one seeks in a reduced-order model. Section 2.3 presents outlines from system theory related to the dynamic linear systems which are the target of MOR in this thesis. Section 2.4 presents an itemized list of the various approaches and section 2.5 explains the approach mentioned in section 2.4.

2.2 Requirements of Model Order Reduction

Since MOR is widely used for the approximation of the original systems, a number of conditions should be satisfied to ensure feasibility of the resulting models. They may be summarized as

- **Accuracy:** The reduced model should accurately follow the original system's terminal behavior. In other words, the approximation error must be small.
- **Compactness:** The reduced number of variables must be significantly lower than the original number of variables.
- **Properties Preservation:** The main physical properties of the original system such as *passivity* and *stability* must be preserved.

- **Computational Efficiency:** The MOR algorithm that is used to extract a model should be relatively inexpensive and computationally efficient for simulating and storing in the computer's memory.
- **Algorithm Re-usabilitby:** The algorithm that is developed should be fairly automated and must be repeatable at any phase of design with a reasonable cost.

2.3 Physical Properties of Linear Dynamical Systems

As discussed in the previous section, the preservation of the physical properties of a system is necessary. Since the main subject of this thesis is discussed upon linear dynamical systems, this section describes the important properties of the linear dynamical systems.

2.3.1 Stability of Linear Systems

Stability is one of the most important properties of dynamical systems. Hence, it is vital [15] that reduced models inherit the stability properties of the original system. Due to the application of MOR in circuit analysis, the stability of Linear Time Invariant (LTI) descriptor (DAE) systems is explained here. The DAE systems are represented as,

$$\boldsymbol{\alpha} \; : \; \begin{cases} \boldsymbol{E}\dot{\boldsymbol{x}}(t) \; = \; \mathbf{A}\;\mathbf{x}(t) \; + \; \mathbf{B}\;\mathbf{u}(t), \quad x(t_0) \; = \; x_0, \\ \mathbf{y}(t) \; = \; \mathbf{C}\;\mathbf{x}(t) \; + \; \mathbf{D}\;\mathbf{u}(t) \end{cases} \tag{2.1}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the descriptor variable and $\dot{\boldsymbol{x}}(t)$ denotes the derivative of $\mathbf{x}(t)$ with respect to the time variable t. $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times p}$, $\mathbf{C} \in \mathbb{R}^{q\times n}$ and $\mathbf{D} \in \mathbb{R}^{q\times p}$ define the model dynamics. $\mathbf{u}(t) \in \mathbb{R}^p$ is a control input vector, $\mathbf{y}(t) \in \mathbb{R}^q$ is the measurement variable, n is the system order, and $\mathbf{E} \in \mathbb{R}^{n\times n}$ is generally a singular matrix, i.e. rank (E) $= r \leq n$. p and q are the number of system inputs and outputs respectively.

2.3.1.1 Internal Stability

A system is said to be internally stable if it is either *asymptotically stable* or *marginally stable*. They are described as,

- **Asymptotic Stability:** For any given initial condition with zero stimuli, if the response of the system converges to zero, then the system is said to be *Asymptotically Stable*. In others words, we can say that a system is asymptotically stable if and only if all the eigenvalues stay in the open left half-plane.
- **Marginal Stability:** For any given initial condition with zero stimuli, if the response of the system remains bounded, then the system is said to be *Marginally Stable*. In others words, we can say that a system is Marginally stable if and only if all the eigenvalues stay in the closed left half-plane, without doubling of eigenvalues on the imaginary axis.

2.3.1.2 External Stability

A system is said to be externally stable if for a bounded input $\mid \mathbf{u}(t) \mid < K_1$ on any interval (t_0, ∞), the response is also bounded such as $\mid \mathbf{y}(t) \mid < K_2$ on (t_0, ∞). A sufficient and necessary condition for bounded-input bounded-output (BIBO) [9,11,58] is given based on its impulse response as,

$$\int_{t_0}^{\infty} \mid h(t) \mid dt \leq M < \infty \tag{2.2}$$

for some constant M.

2.3.2 Passivity

Systems which do not generate energy themselves are called passive systems. It can also be stated as systems in which energy supplied to the system is always greater than energy dissipated by the system [3, 64].

2.4 Order Reduction Algorithms

A number of methods have been developed for model order reduction till date. Of these, the widely used are described in the following sections.

A. **Polynomial Approximations of transfer function:**

i) Explicit moment matching techniques and asymptotic waveform evaluation (AWE)

B. **Projection based techniques:**

i) Krylov-subspace methods

ii) Singula Value Decomposition(SVD) based methods

- Truncated balanced state-space representation (TBR)
- Proper Orthogonal Decomposition (POD)

C. **Non-project based MOR methods:**

i) Hankel optimal model reduction

ii) Singular perturbation

iii) Transfer function fitting methods

iv) Model reduction via convex optimization

2.5 Polynomial Approximations of Transfer Functions

The transfer function $\mathbf{H}(s)$ of a linear multiport network is a complex-valued matrix and it relates input to output at each complex-frequency point as $\mathbf{Y}(s) = \mathbf{H}(s)\ \mathbf{U}(s)$. For a SISO system, $\mathbf{H}(s)$ is a complex-valued scalar function, and is defined as,

$$\mathbf{H}(s) \ = \ \frac{\mathbf{Y}(s)}{\mathbf{U}(s)} \tag{2.3}$$

Let us suppose that (2.3) is an asymptotically stable transfer function with singularities $\in \mathbb{C}_-$. At any frequency point $s_0 \in \mathbb{C}_+$, it can be expanded using Taylor series as,

$$\mathbf{H}(s) = \sum_{n=0}^{\infty} m_n (s - s_0)^n, \tag{2.4}$$

where,

$$m_n = \frac{1}{n!}\ \mathrm{x}\ \frac{d_n \mathbf{H}(s)}{ds^n}, \Big|_{s=0}. \tag{2.5}$$

The system transfer function $\mathbf{H}(s)$ can be obtained from Laplace transform of impulse response as

$$\mathbf{H}(s) \triangleq \int_0^{\infty} h(t) e^{-st} dt = \int_0^{\infty} h(t) \left(\sum_{n=0}^{\infty} \frac{(-1)^n}{n!} t^n \right) dt = \sum_{n=0}^{\infty} \left(\frac{(-1)^n}{n!} \int_0^{\infty} t^n h(t) dt \right) s^n. \tag{2.6}$$

Comparing (2.5) and (2.6) the coefficient can be defined as,

$$m_n = \frac{(-1)^n}{n!} \int_0^{\infty} t^n h(t) dt \tag{2.7}$$

Similarly, Laplace transformation can also be applied to electrical networks as shown in (1.1), the corresponding *Laplace Domain* equations look like,

$$\begin{cases} \mathbf{C}s\mathbf{X}(s) + \mathbf{G}\mathbf{X}(s) = \mathbf{B}\mathbf{U}(s) \\ \mathbf{Y}(s) = \mathbf{L}\mathbf{X}(s), \end{cases} \tag{2.8}$$

Using the equations (2.8) and (2.3), the transfer function can be written as,

$$\mathbf{H}(s) = \mathbf{L}(\mathbf{G} + s\mathbf{C})^{-1}\mathbf{B}. \tag{2.9}$$

Let $s_0 \in \mathbb{C}_+$ at which the matrix pencil $(\mathbf{G} + s_0\mathbf{C})$ is non-singular. Equation (2.9) can be written as,

$$\mathbf{H}(s) = \mathbf{L}(\mathbf{I} + (s - s_0)\mathbf{A})^{-1}\mathbf{R}, \tag{2.10}$$

where,

$$\mathbf{A} \triangleq (\mathbf{G} + s_0\mathbf{C})^{-1}\mathbf{C} \tag{2.11}$$

and

$$\mathbf{R} \triangleq (\mathbf{G} + s_0\mathbf{C})^{-1}\mathbf{B} \tag{2.12}$$

Now, the i^{th} moment of the function at s_0 is defined as

$$\mathbf{M}_i(s_0) = \mathbf{L}\mathbf{M}_i(s_0) = \mathbf{L}\ \mathbf{A}^j\mathbf{R},\ for\ i\ =\ \{0, 1, 2, 3, ..\}. \tag{2.13}$$

2.5.1 AWE Based on Explicit Moment Matching

In Asymptotic Waveform Evaluation (AWE) [44] algorithm explicit moment matching based on Padè approximation was used to obtain the reduced order rational function. The

Padè approximant is obtained by computation of first 2m moments of $\mathbf{H}$(s) [12, 43, 45]. With AWE we have to be aware that, (a) accuracy is not guaranteed and (b) AWE is unstable for higher order moment approximation.

2.6 Projection Based Methods

In the previous section, we mentioned that explicit moment matching methods suffer from numerical ill-conditioning. Hence, few better methods which are based on implicit moment matching [58] have been introduced. These are the projection based model order reduction techniques [14, 16, 17, 39, 54]. The central idea of projection based methods is to reduce the number of state variables by projecting the state variables vector $\mathbf{x}$ with the dimension n, onto a subspace spanned by the column vectors of an orthogonal matrix $\mathbf{Q}$ whose order $m \ll n$. Let us suppose that there is a variable z in the reduced space such that its projection back to the original space will lead to a minimal error $\boldsymbol{\xi}(t)$,

$$\mathbf{x}(t) = \mathbf{Q}\mathbf{z}(t) + \boldsymbol{\xi}(t), \tag{2.14}$$

where, $\mathbf{x}(t) \in \mathbb{R}^n$ and $\mathbf{Q}\ \mathbf{z}(t) \in \text{colsp}(\mathbf{Q})$ (colsp means column space). The error vector $\boldsymbol{\xi}(t)$ must be confined to the *orthogonal complement subspace* of colsp($\mathbf{Q}$), where

$$colsp(\mathbf{Q})^{\perp} \triangleq \{w \mid w \in \mathbb{R}^n, w^T v = 0, \forall\ v \in colsp(\mathbf{Q})\}. \tag{2.15}$$

Mathematically, it is straightforward that the orthogonal component of the column space of $\mathbf{Q} \in \mathbb{R}^m$ is the null space of $\mathbf{Q}^{\mathrm{T}}$. Substituting (2.14) in (1.1), we get

$$\mathbf{C}\frac{d}{dt}(\mathbf{Q}\mathbf{z}(t) + \boldsymbol{\xi}(t)) + \mathbf{G}(\mathbf{Q}\mathbf{z}(t) + \boldsymbol{\xi}(t)) - \mathbf{B}\mathbf{u}(t) = 0 \tag{2.16}$$

and

$$\mathbf{C}\frac{d}{dt}\mathbf{Q}\mathbf{z}(t) + \mathbf{G}\mathbf{Q}\mathbf{z}(t) - \mathbf{B}\mathbf{u}(t) = \mathbf{C}\frac{d}{dt}\boldsymbol{\xi}(t) + \mathbf{G}\boldsymbol{\xi}(t). \tag{2.17}$$

Consider $\mathbf{x}(t) \approx \tilde{\mathrm{x}} = \mathbf{Q}\mathbf{z}$, using this solution residual error can be written as,

$$\mathcal{R}(\boldsymbol{\xi}(t)) \triangleq \mathbf{C}\frac{d}{dt}\mathbf{Q}\mathbf{z}(t) + \mathbf{G}\mathbf{Q}\mathbf{z}(t) - \mathbf{B}\mathbf{u}(t) = \mathbf{C}\frac{d}{dt}\boldsymbol{\xi}(t) + \mathbf{G}\boldsymbol{\xi}(t). \tag{2.18}$$

Multiplying (2.18) by $\mathbf{Q}^{\mathrm{T}}$ and using the property $\mathbf{Q}^{\mathrm{T}}\ \boldsymbol{\xi}(t) = 0$,

$$\mathbf{Q}^T\mathcal{R}(\boldsymbol{\xi}(t)) \triangleq (\mathbf{Q}^T\mathbf{C}\ \mathbf{Q})\frac{d}{dt}\mathbf{z}(t) + (\mathbf{Q}^T\mathbf{G}\ \mathbf{Q})\mathbf{z}(t) - (\mathbf{Q}^T\mathbf{B})\mathbf{u}(t) = \mathbf{C}\frac{d}{dt}\mathbf{Q}^T\boldsymbol{\xi}(t) + \mathbf{G}\mathbf{Q}^T\boldsymbol{\xi}(t) = 0, \tag{2.19}$$

and therefore,

$$(\mathbf{Q}^T\mathbf{C}\ \mathbf{Q})\frac{d}{dt}\mathbf{z}(t) + (\mathbf{Q}^T\mathbf{G}\ \mathbf{Q})\mathbf{z}(t) - (\mathbf{Q}^T\mathbf{B})\mathbf{u}(t) = 0. \tag{2.20}$$

The approximated output is obtained as

$$\mathbf{y}(t) = \mathbf{L}^T\mathbf{Q}\mathbf{x}(t) \tag{2.21}$$

Hence, from the reduced order model shown in (2.20) and its output equation (2.21), the reduce MNA matrices are

$$\hat{\mathbf{C}} = \mathbf{Q}^T\mathbf{C}\mathbf{Q}, \qquad \hat{\mathbf{G}} = \mathbf{Q}^T\mathbf{G}\mathbf{Q},$$
$$\hat{\mathbf{B}} = \mathbf{Q}^T\mathbf{B}, \text{ and } \quad \hat{\mathbf{L}} = \mathbf{L}\mathbf{Q}. \tag{2.22}$$

The next step is to find an orthogonal basis to span the reduced projection space. There are many methods which came up with the idea for computing the basis of which Krylov-

subspaces are worth studying.

2.6.1 Krylov-Subspace Methods

Krylov-subspace methods are the most commonly used model order reduction methods [6, 10, 16, 20, 54]. These methods construct the orthogonal projection matrix as follows:

$$\begin{aligned} colsp\{\mathbf{Q}\} &= \mathcal{K}_m(\mathbf{A}, \mathbf{R}) \\ &= span\{\mathbf{R}, \mathbf{AR}, ...\mathbf{A}^{m-1}\mathbf{R}\}(span\{x_1, x_2, ..x_n\}\ means\ set\ of\ vectors\ forming\ a\ space) \end{aligned} \tag{2.23}$$

$\mathbf{A}$ and $\mathbf{R}$ are defined in (2.11) and (2.12).

2.6.1.1 Arnoldi Algorithm

The Arnoldi method is used to produce a set of orthogonal vectors to form a basis for Krylov subspace as shown in (2.23). It was introduced by W.E. Arnoldi in 1951 in the field of mathematics [5]. Arnoldi process uses *"modified Gram-Schmidt"* [50] orthogonalization for the calculation of projection matrices. The complete algorithm of the Arnoldi process is shown in Algorithm 1.

Algorithm 1 Arnoldi algorithm

Input: G, C, L, B, u

Output: Q

1: *Solve:* $\mathbf{G}\hat{q}_0 = \mathbf{B}$

2: $q_1 = \hat{q}_0 \ / \ \| \ \hat{q}_0 \ \|$

3: **for** j = 1,...u **do**

4: r = Aq_j

5: **for** i = 1,...,j **do**

6: $h_{ij} = q_j^* r, \quad r = r - q_j \ h_{ij}$

7: **end for**

8: $h_{j+1,j} = \| \ r \ \|$

9: **if** $h_{j+1,j} = 0$ **then**

10: return (q_1, ..., q_j, H $\in \mathbb{F}^{j \times j}$

11: **end if**

12: $q_{j+1} = r/h_{j+1,j}$

13: **end for**

14: return (q_1, ..., q_{k+1}, H $\in \mathbb{F}^{(k+1)\times k}$)

2.6.1.2 Lanczos Process

Lanzos process [30] is an iterative procedure which is used for successive reduction of any square matrix into a sequence of tridiagonal matrices. It was first introduced for solving the problem of eigenvalues and was extended into MOR as a projection based method through Padè via Lanczos (PVL) method. The algorithm of the Lanczos process is explained in Algorithm 2.

Algorithm 2 Lanczos process

Input: G, C, L, B, u

Output: Q, P

1: *Solve:* **AG = C**

2: q = x/‖ x ‖ $\quad$ (Q_1 = [q])

3: r = Aq

4: α_1 = q*r

5: r = r - α_1q

6: β_1 = ‖ r ‖

7: **for** j = 2,3,...u **do**

8: $\quad$ v = q; q = r/β_{j-1}; (Q_j = [(Q_{j-1}, q])

9: $\quad$ r = Aq - β_{j-1}v

10: $\quad$ α_j = q*r

11: $\quad$ r = r - α_jq

12: $\quad$ β_j = ‖ r ‖

13: $\quad$ **if** β_j = 0 **then**

14: $\quad\quad$ return (Q ∈ $\mathbb{F}^{n\times j}$; α_1,....,α_j; β_1,....β_{j-1})

15: $\quad$ **end if**

16: **end for**

2.6.1.3 PRIMA

PRIMA (*passive reduced-order interconnect macro moeling algorithm*) [39] is a direct extension of the block Arnoldi technique. The main advantage of this method is preserving the passivity of the system which is obtained with a simpler formulation. Let an RLC dynamic (MIMO) system be described as,

$$\mathbf{C}\frac{d}{dt}\mathbf{x}(t) \;+\; \mathbf{G}\;\mathbf{x}(t) \;=\; \mathbf{B}\;\mathbf{u}(t) \tag{2.24}$$
$$\mathbf{y}(t) \;=\; \mathbf{L}\;\mathbf{x}(t)\;,$$

A change of variable is applied on (2.24) using an orthogonal projection matrix $\mathbf{Q}$ to obtain the reduced-order model based on a congruence transformation as:

$$\hat{\mathbf{C}}\frac{d}{dt}\mathbf{z}(t) \;+\; \hat{\mathbf{G}}\mathbf{z}(t) \;=\; \hat{\mathbf{B}}\mathbf{u}(t) \tag{2.25}$$
$$\hat{\mathbf{y}}(t) \;=\; \hat{\mathbf{L}}\mathbf{z}(t)\;,$$

The reduced system approximates the original system in response while preserving the main properties of the original system. The reduced system of order q preserves the first $\lfloor\, q/p \,\rfloor$ block moments [10, 39] of the original system.

2.6.2 Truncated Balance Realization (TBR)

Control Theory based truncated balance realization (TBR) methods [13, 19, 24, 31, 32, 34, 35, 41, 42, 51, 56–58, 63, 65] are one of the alternatives to find the reduced models of LTI systems. The weak uncontrollable and unobservable states are truncated using an SVD approach in these TBR methods. The definitions of controllability and observability are as follows,

Controllability: If a dynamic system is fully flexible for any kind of control input, the system is said to be controllable.

Observability: At any point of time, if we can observe the changes inside the system, it is said to be observable.

2.6.2.1 Standard / Conventional TBR

The controllability Grammian W_c and the observability Grammian W_o provide the key information for the TBR procedures. These Grammians are Hermitian positive definite matrices obtained [19, 35] from solving the Lyapunov equations of the system. Those Lyapunov equations are of the form,

$$AW_c + W_cA^T = -BB^T \tag{2.26}$$
$$A^TW_o + W_oA = -CC^T$$

The eigenvalues of W_c and W_o are the *Hankel singular values*. Small values of Hankel singular values mean that those internal dynamic modes are close to non-controllable or non-observable or both [58]. The algorithm of TBR procedure is explained in Algorithm 3.

Algorithm 3 Conventional TBR algorithm

Input: Original Model (A, B, C, D)

Output: Reduced Model ($\hat{A}$,$\hat{B}$,$\hat{C}$,$\hat{D}$)

1: *Solve:* A W_c + $W_c$$A^T$ = -B B^T for W_c

2: *Solve:* A^T W_o + W_oA = -C C^T for W_o

3: Calculate Cholesky factors W_c = L_c $L_o{}^T$ and W_o = L_o $L_o{}^T$

4: Calculate SVD of Cholesky facotrs UΣV^T = $L_o{}^T$ L (Σ = diagonal positive and U,V have orthonormal columns)

5: Calculate balance transformation matrices T = $L_c V\Sigma^{-1/2}$, T^{-1} = $\Sigma^{-1/2}U^TL_o{}^T$

6: Form the balanced realization as $\tilde{A}$ = T^{-1}AT, $\tilde{B}$ = T^{-1}B, $\tilde{C}$ = CT

7: Select the reduced order and partition $\tilde{A}$, $\tilde{B}$ and $\tilde{C}$

8: Truncate $\tilde{A}$, $\tilde{B}$ and $\tilde{C}$ to form reduced realization $\hat{A}$, $\hat{B}$ and $\hat{C}$, $\hat{D}$ = D

2.6.3 Proper Orthogonal Decomposition (POD) methods

Proper Orthogonal Decomposition is also called as Karhunen-Loève decomposition [21] or principal component analysis [26]. It is a technique that derives reduced models by linear projection and it also helps in analyzing multi-dimensional data. Based on the data used for constructing orthonormal projection basis, the POD has two different approaches Time-domain POD and Frequency-domain POD. In the time-domain POD, the projection matrix is constructed from the snapshots of the state vectors at N different time points and is used to form the data matrix $\mathbf{X} = [\mathrm{x}(t_1), \mathrm{x}(t_2), ...\mathrm{x}(t_N)] \in \mathbb{R}^{n \times N}$. The time samples are obtained from the transient simulation of the system with some input stimulus [8]. For the frequency-domain POD, the snapshots are obtained from the frequency response at certain frequency points of interest. The POD approaches are hailed because of the following reasons,

- The time domain samples $\mathbf{X} = [\mathrm{x}(t_1), \mathrm{x}(t_2), ...\mathrm{x}(t_N)]$ and the frequency domain response is easy to compute. In both cases, the advantage of systems matrices' sparsity can be taken.
- Easy to implement.
- Very reliable in practice.
- It can also be applied straightforwardly to non-linear systems.

The algorithm of Frequency-domain POD is explained in Algorithm-4

Algorithm 4 Frequency-Domain POD algorithm

Input: Original Model (A, B, C, D)

Output: Reduced Model ($\hat{A}$,$\hat{B}$,$\hat{C}$,$\hat{D}$)

1: Select interested real frequency points $w_k, k = 1, .., N$

2: Compute the response of original system at those frequencies $X(w_i) = (jw_iI_n - A)^{-1}B$ for $i = 1, .., N$ and store them in the complex snapshot, $X_s = [X(w_1), ..., X(w_N)]$

3: Construnct the correlation matrix $R = \frac{1}{N}(X_s^H X_s), (\in \mathbb{C}^{N\times N})$

4: Solve eigenvalue matrix problem, $RT_i = \lambda_i T_i$

5: Form the basis vector $v_i = \frac{1}{N}\sum_{i=1}^{N} T_i X(w_i)$

6: Construct projection matrix, $V = [v_i, ..., v_m]$ where m $\leq$ N

7: Form the reduced systems matrices, $\hat{A} = V^T AV$, $\hat{B} = V^T B$, $\hat{C}$ = CV, and $\hat{D}$ = D

2.6.4 Non-Projection Based MOR methods

The name itself suggests that Non-Projection Based MOR methods do not use any projection matrix to find the reduced models. The commonly used methods of this kind are stated below.

2.6.4.1 Hankel Optimal Model Reduction

The task of Hankel optimal model reduction of a matrix transfer function $\boldsymbol{G}(s)$ calls for finding a stable reduced system $\hat{\boldsymbol{G}}(s)$ of order less than a given positive integer m, such that the Hankel norm $||\ \boldsymbol{\xi}\ ||$ = $\boldsymbol{G}(s)$ - $\hat{\boldsymbol{G}}(s)$ is minimal. Since Hankel operator $\mathbb{H}$ represents a "part" of the total LTI system with transfer matrix $\boldsymbol{G}(s)$, Hankel norm is never larger than H-Infinity norm. Hence, Hankel optimal model reduction setup can be viewed as a relaxation of the "original" (H-Infinity optimal) model reduction formulation. While no

acceptance solution is available for H-Infinity case, Hankel optimal model reduction has an elegant and algorithmically effiicient solution [4, 18, 19, 52].

2.6.4.2 Singular Perturbation

In the projection based MOR methods, a co-ordinate transformation of the original system's state space to a lower dimensional subspace is performed. In TBR, by setting the last $(n - m)$ states to zeros, state transformation leads to a balanced system. Besides setting states to zeros, we can set the derivatives of the states as zeros. Such a procedure is called *state residualization*, which is similar to the *singular perturbation approximation* [59].

2.6.4.3 Transfer Function Fitting Method

The difficulty in obtaining analytical models for high-speed modules because of the increase in operating frequencies led to the characterization based on terminal responses. Linear devices and subsystems can also be characterized in the frequency domain which is more feasible in applications than time-domain characterization. For time-domain simulations to be feasible, a state-space model that approximates the sampled transfer function of the system can be constructed. Since we are leading to a minimal model starting from the characterization of the original system, these methods can also be treated as model order reduction methods. Since they are approximating the transfer function of the original system, these are called *Transfer Function Fitting Methods*.

2.6.5 Other Methods for Order Reduction

The necessity for the reduced models and the limitations of the current MOR methods are leading for the advancement of a variety of MOR methods regardless of the above presented categories. The newer techniques are aiming at combining the advantages of different methods. Because of the growing demand for the reduced models, it's not just

necessary to provide the reduced models but we also have to ensure that the resulting model is the best possible minimal model and any model with the order below that will lead to inaccurate results. In this regard, *Order Selection* plays a crucial role since it can determine the order for the best possible order for the reduced model. Next chapter provides a detailed explanation of the Order selection.

2.7 Discussion

This chapter described the landscape of the various approaches used to approximate large dynamical systems with reduced-order models. Those approaches are collectively known in the relevant literature as Model-Order Reduction (MOR) technique. One of the fundamental tasks involved in the MOR is the selection of the appropriate reduced order for the approximant model. In many of the existing approaches to MOR, there is a well-defined relation between the reduced order and the accuracy of the approximation, making it possible to specify the order of the reduced system based on the desired accuracy. One of the approaches that can serve as an example in this regard is the TBR algorithm.

Nonetheless, for the projection-based MOR techniques, selecting the most appropriate order has not been an easy goal even though the approach itself has been in usage for a long time. The following chapter will present an approach that was recently introduced for that purpose. The objective in this thesis is to examine that approach and propose an alternative technique that is aimed at alleviating its computational difficulties.

Chapter 3

Order Selection

Selection of order for the reduced model is a practical and important problem in prominently used order reduction techniques. It is important for the macro model based approximations to pre-define the accuracy while not over-estimating the order, which might lead to inefficient simulations. The importance of an optimum order is even more increased if the reduced macro model is used as part of a larger simulation task such as design centering, optimization, statistical analysis, etc. In such cases, the computational cost of the task due to overestimating the order of the reduced order might be significantly more than the computational cost of pre-estimating the optimum order. Techniques like *a-priori* [1] for predicting an optimum order are generally heuristic in nature. This chapter reviews a recently introduced technique aimed at determining an optimum order for reduced-order models obtained through the Krylov-based projection methods. The approach reviewed here has been introduced in [37, 38].

3.1 Preliminaries

This section discusses the fundamentals that are behind the order selection algorithm presented in [37, 38].

The set of differential algebraic equations used to represent the dynamical behavior of the system states [7, 29, 49, 55] for electrical networks are obtained using modified nodal analysis (MNA) circuit formulation [25, 36, 45, 61] and are of the form:

$$\begin{aligned} \mathbf{C}\frac{d}{dt}\mathbf{x}(t) \; + \; \mathbf{G}\; \mathbf{x}(t) \; &= \; \mathbf{B}\; \mathbf{u}(t) \\ \mathbf{y}(t) \; &= \; \mathbf{L}\; \mathbf{x}(t)\;, \end{aligned} \tag{3.1}$$

The solution of the system in (3.1), is $\mathbf{x}(t)$, a function of time t and is a vector of size n, may be considered as a trajectory in an n-dimensional space, which represents the original space of the larger system.

Figure 3.1 illustrates the concept of a trajectory by considering one in 3D space, where $\mathbf{x}(t) = [\; \mathrm{x}_1(t),\; \mathrm{x}_2(t),\; \mathrm{x}_3(t)\;]$.

The goal of projection-based MOR is to represent the n-dimensional trajectory $\mathbf{x}(t)$ with another trajectory $\mathbf{z}(t) \in \; \mathbb{R}^m$ where $m \ll n$. The relation between $\mathbf{x}(t)$ and $\mathbf{z}(t)$ is such that [10, 20, 60]

$$\mathbf{x}(t) \; = \; \mathbf{Q}\mathbf{z}(t) \tag{3.2}$$

where $\mathbf{Q} \in \; \mathbb{R}^{n\times m}$ is an orthogonal matrix, i.e,

$$\mathbf{Q}^T\; \mathbf{Q} \; = \; \mathbf{I}_m \quad (\textit{An identity matrix of order } m \times m) \tag{3.3}$$

There are two main issues that are related to the orthogonal matrix

1. The first issue is the computation of this matrix. This issue has been addressed through several techniques and is out of the scope of this thesis.

2. The second issue is related to the optimum order for this matrix, namely m which is the number of columns in $\mathbf{Q}$.

For the first issue, this thesis adopts the Krylov subspace techniques where $\mathbf{Q}$ is chosen such that

$$Colspan\ \{\mathbf{Q}\} \ = \ Colspan\{\ \mathbf{G}^{-1}\mathbf{B},\ \mathbf{A}\mathbf{G}^{-}1\mathbf{B}, ..., \mathbf{A}^{m-1}\mathbf{G}^{-1}\mathbf{B}\} \tag{3.4}$$

where $\mathbf{A} \ = \ \mathbf{G}^{-1}\mathbf{B}$.

As to the second issue, the choice of m for optimum performance, the concept of "False Nearest Neighbor" has to be elaborated.

3.2 False Nearest Neighbors (FNN)

FNN [28,33,47,48] is the concept that is framed around defining a measure for the closeness between two points on the trajectory, for the trajectory traced by $\mathbf{x}(t) \in \ \mathbb{R}^n$. Let $\mathrm{x}(t_i)$ and $\mathrm{x}(t_j)$ be two such points in this trajectory. The Euclidean distance between these two points defined by

$$d_n\ (t_i,\ t_j) \ = \| \ \mathrm{x}(t_i) \ - \ \mathrm{x}(t_j) \ \| = \ \sqrt{\sum_{\rho=1}^{n}(x_\rho(t_i) - x_\rho(t_j))^2}, \tag{3.5}$$

serves as a measure for the closeness between $\mathrm{x}(t_i)$ and $\mathrm{x}(t_j)$.

A similar measure for the distance between points in $\mathbb{R}^m$ on a trajectory corresponding to $\mathbf{z}(t)$ may be defined in an analogous manner

$$d_m\ (t_i,\ t_j) \ = \| \ z(t_i) \ - \ z(t_j) \ \| = \ \sqrt{\sum_{\rho=1}^{n}(z_\rho(t_i) - z_\rho(t_j))^2}, \tag{3.6}$$

The process that maps $\mathbf{z}(t)$ (reduced-dimension trajectory) into the $\mathbf{x}(t)$ (large dimension trajectory) is the one shown in equation (3.2). This equation also reads, through premultiplying by $\mathbf{Q}^T$,

$$\mathbf{Q}^T\mathbf{x}(t) = \mathbf{z}(t) \tag{3.7}$$

which reveals that $\mathbf{z}(t)$ is a "compressed" version of $\mathbf{x}(t)$. Indeed, the space with n dimension is compressed to dimension m. A suitable way to visualize this compression is by picturing $\mathbf{x}(t)$ as a trajectory in 3-dimensional space ($n = 3$) and letting $\mathbf{z}(t)$ be a trajectory in 2-dimensional space ($m = 2$), as shown by Figure 3.3.

The underlying principle in the FNN is that projection operation must conserve the neighborhoods for each point in the trajectories. If it does not then m is not sufficient for the purpose of MOR and needs to be increased. On the other hand, if it does then m is sufficient and therefore is considered an optimal order for the reduced system.

To elaborate further on the above point, we consider the trajectory shown in 3.1, where two points $\hat{A}$ and $\hat{E}$ coincide. Let these two points in the 2-dimensional space be-crossed at two different time instants, say t_i and t_j, thereby implying that $\mathrm{z}(t_i) = \mathrm{z}(t_j)$. So the question to ask here is $\mathrm{x}(t_i) = \mathrm{x}(t_j)$? If that is the case, then $m = 2$ is perhaps sufficient as an order for the reduced system. If it is not, then m is not sufficiently large and needs to be incremented.

More generally, the idea of FNN and the algorithm based on it moves around the trajectory in the reduced space exploring the points on that trajectory. If these points are reported as being neighbors in $\mathbb{R}^m$, while their corresponding points are not neighbors in $\mathbb{R}^n$, these points are flagged as FNN. If the count of FNN is above zero, then m is not sufficiently large and needs to be increased. If the count of FNN reaches zero, then m is sufficient as an order for the reduced system.

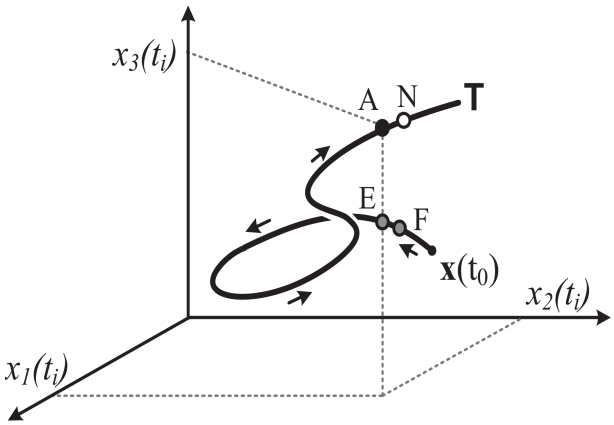

Figure 3.1: Any state corresponding to a time instant can be represented by a point (eg. A, E, F and E) on the trajectory curve (T)

The above discussion leaves an important issue still undefined, namely how to define the neighborhood of a point x_i is defined as the open set U_i given by an (open) ball with radius ε_n. Figure 3.2 illustrates this idea by showing a trajectory and zooming in on a certain point x(t_i) indicating its neighborhood ball.

Suppose the point $z(t_i)$ in the m-dimensional space be the projection of the reference point $x(t_i)$ in n-dimensional space. The points $z(t_j), z(t_l)$ and $z(t_r)$ are considered as true neighbors of $z(t_i)$ because their mapping points are neighbors of the reference point and $z(t_k), z(t_s)$ are considered as false neighbors because they are not neighbors of x(t_i) in the original space.

Finally, it can be said that if a projection to a subspace is made with a sufficient order, the neighborhood structure of the original trajectory is inherited by the projected trajectory, which implies that (a) The original nearest neighboring points remain neighbors in the projection, (b) The near neighbors in the reduced space are true neighbors. These implications are based on the important properties shown in [37, 38]

Contraction Property: In projection using an orthogonal matrix $Q_{n\times m}(m \ll n)$, the inner points of ε_n-neighborhood of any point on the original (n-dimensional) trajectory are preserved as inner points of ε_m-neighborhood in (m-dimensional) reduced subspace, where

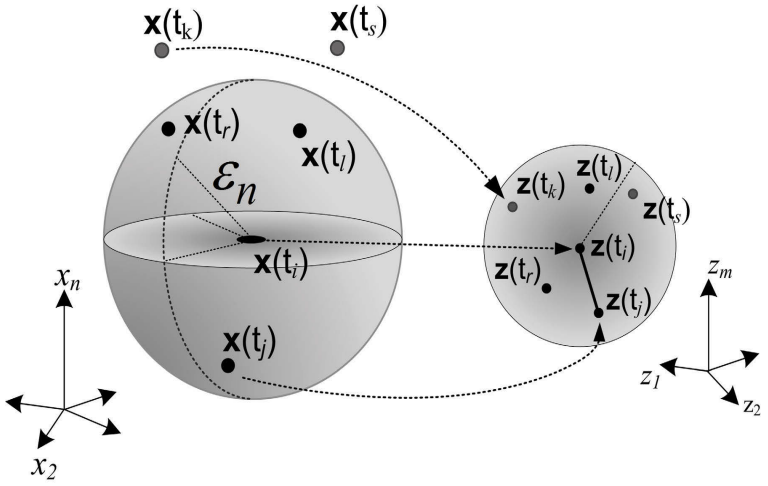

Figure 3.2: Illustration of false neighborhood points

$\varepsilon_n \leq \varepsilon_m$

True Neighbors Property: The accuracy for the macro model is ensured if and only if all the nearest neighboring points on the projected trajectory are true neighbors.

3.3 Unfolding the Projected Trajectory

In the order selection process, the order of the reduced space is consecutively increased starting from a low-dimensional space. This will lead to the expansion of projected trajectory into higher dimensions. In this process, some neighboring points reveal themselves as false nearest neighbors by moving far apart. This projection into higher dimensions can be visualized as unfolding the sections gradually which are folded over. To monitor this unfolding process, the count of false nearest neighbors is utilized. Eventually, the count of false nearest neighbors drops to zero at some order (m_0), and does not reveal any new false nearest neighbors upon increasing the order. This is illustrated in Figure 3.3

In a transition from order m to $m+1$, the changes to two nearest-neighbors (**A** and **B**), is visualized in the above figure. It shows the unfolding of a trajectory which is embedded in a subspace with insufficient order. Upon increasing the dimension from m to $m+1$,

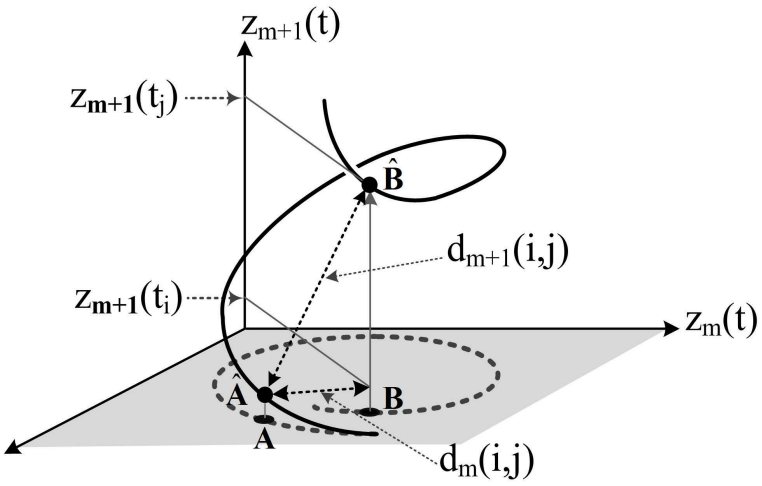

Figure 3.3: Revealing the false nearest neighbors in the unfolding process

the two points moved to their new locations $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{B}}$. Their distance changed from $\mathbf{AB} = \mathbf{d}_m(\mathbf{i,j})$ in m-dimensional space to $\hat{\mathbf{A}}\hat{\mathbf{B}} = d_{m+1}(i,j)$ in the $m+1$ dimensional space. By moving apart from **A** by such distance, **B** revealed itself as a false neighbor of **A** in m dimension. From the Figure 3.3, we have

$$d^2_{m+1}(i,j) = d^2_m(i,j) + (z_{m+1}(t_i) - z_{m+1}(t_j))^2. \tag{3.8}$$

In the order selection method, the displacement vector between two neighborhood points on the new axis is used as the measure for monitoring the behavior of neighboring points. That is if,

$$\Delta_{z_{m+1}}(i,j) = \mid z_{m+1}(t_i) - z_{m+1}(t_j) \mid, \tag{3.9}$$

$\Delta_{z_{m+1}}(i,j)$ is large compared their Euclidean distance $d_m(i,j)$, **A** and **B** are false neighbors. This comparison [2, 28, 46–48] is done by checking the ratio

$$R_{ij} = \left[\frac{d^2_{m+1}(i,j) - d^2_m(i,j)}{d^2_m(i,j)}\right]^{1/2} = \frac{\mid z_{m+1}(t_i) - zm + 1(t_j) \mid}{d_m(i,j)} = \frac{\Delta_{z_{m+1}}(i,j)}{d_m(i,j)}. \tag{3.10}$$

Hence, it can be said that an order m_0 is an optimally minimum reduced order if increasing the order to m_1 ($m_0 \leq m_1$), does not reveal any false nearest neighboring points.

3.4 Computational Steps of Order Selection

The order selection method is summazried as follows. In the following section in the equations "t" is dropped for simplicity in the notation (e.g $z(t_i)$ is referred as $z(i)$).

(1) The Order selection method uses the time series data of projected trajectory z($\cdot$) $\in \mathbb{R}^{m \times N}$,

$$z(\cdot) = \{z(i) \in \mathbb{R}^{m \times 1} \mid z(i) = Q^T \mathrm{x}(i),\ for\ i\ =\ 1, .., N\}. \tag{3.11}$$

This requires the transient simulation of the original circuit with any input which covers a wide frequency range up to the maximum frequency of interest. Since the goal of linear model reduction is to represent the original system up to certain frequency with an accurate model, it is important that the input signals covers the adequate frequency range of interest. Initially, small number (m) orthogonal basis is used to form the projector matrix Q and time series is data is obtained from $z(\cdot) = Q^T \mathrm{x}(\cdot)$.

(2) On the projected trajectory $z(i)$, a set of close points $\prod_i$ to each point is found by the following check:

$$d_{m_s}(i, j)\ \leq\ R,\ for\ j\ =\ 1, .., N\ and\ j\ \neq i, \tag{3.12}$$

where R is the search radius.

In general, any search radius R can be used to get $\prod_i$, but it is good to keep R small which leads to $\prod_i$ with few close points. Because large radius R, leads to large the number of neighbors, which does not impact the accuracy of the result, but will increase

the computational cost. This search algorithm is outlined in "Algorithm-5".

(3) In the next step, the orthogonal basis number is increased from m to $m+1$, and time series data of the projected trajectory in the new dimension is computed as

$$z_{m+1}(\cdot) \;=\; q_{m+1}^{T}\mathrm{x}(\cdot) \tag{3.13}$$

(4) All the points $z(j) \in \prod_i$ to $z(i)$ { i = 1,..N}, that satisfies the ratio (3.10) are marked as false neighbors

$$R_{ij} \;=\; \frac{\Delta_{z_{m+1}}(i,j)}{d_m(i,j)} > \rho_t, \tag{3.14}$$

ρ_t is a pre-specified threshold value. This False Nearest Neighborhood (FNN) search is outlined in "Algorithm-6".

(5) By repeating the above steps (2) - (4), the trajectory curve is projected and unfolded into higher dimensions. Finally, at some order m_0, the false nearest neighbors count in step(4) drops to zero. Hence, increasing the order further does not reveal any new false nearest neighbors. Such m_0 is said to be the optimum minimal order of the reduced space.

The above computational steps are outlined in "Algorithm-7".

There are some noteworthy points in this order selection method. They are:

1) Initially, the search in "Algorithm-5" is performed only once on the time series of order N x m from the projected trajectory. m is the starting (small) order and N is the number of time points. Hence, finding the neighbors in a set of N vectors of size m is done in $\mathcal{O}(N \log(m))$ [27, 53].

2) The FNN algorithm in "Algorithm-6" is computed in $\mathcal{O}(Nn)$ and it is also suitable for parallel implementation which leads in the additional reduction in computational cost.

3) This Order selection method does not require the simulation or formulation of reduced macro model.

Algorithm 5 Neighborhood Search

Input: z $\in \mathbb{R}^{mxN}$

Output: $\prod$

I $\leftarrow$ { i | 1 $\leq$ i $\leq$ N };
for i $\in$ I **do**
    for j $\in$ I - i **do**
        Find $d_m(i, j)$;
        if $d_m(i, j) < R$ **then**
            $\prod_i \leftarrow$ (j, $d_m(i, j)$);
        end if
    end for
end for

Algorithm 6 False Nearest Neighbor (FNN)

Input: $z_{m+1} \in \mathbb{R}^{m\text{x}1}$

Output: FNN count and $\prod$ (updated)

for i $\in$ I (If $\prod_i \neq 0$ **do**
  for j $\in$ $\prod_i$ **do**
    Compute R_{ij} from (3.10)
    if $R_{ij} > \rho_t$ **then**
      FNN + 1;
    end if
    Compute $d_{m+1}(i, j)$ from (3.4)
    $\prod_i \leftarrow (\ j,\ d_{m+1}(i, j))$
  end for
end for

Algorithm 7 Order Selection Algorithm

Input: x $\in \mathbb{R}^{nxN}$

Output: Optimum minimal order m_0

$m \leftarrow$ Small-starting order

$Q_{nxm} \leftarrow$ Projection matrix

Find projected trajectory from (3.7)

$\prod \leftarrow$ From Algorithm - 4

while FNN count > 0 **do**

    $m \leftarrow m + 1$

    Find q_{m+1}

    Find z_{m+1} from (3.9)

    FNN count, $\prod \leftarrow$ From Algorithm-5

end while Optimum minimum reduced order ($m_0 \leftarrow m$)

3.5 Important Remarks

This chapter focused on describing a recent approach [37,38] that addressed the problem of finding the optimal order for the reduced models obtained through the Krylov projection techniques.

As explained throughout the chapter, the algorithm in [37,38] relies on generating the trajectory of the system impulse response and using the notion of False Nearest Neighbor (FNN) to identify the best order for the reduced model.

The main algorithm may thus be summarized in the following steps

1. Consider the original system described by the MNA formulation in the time-domain

$$\boldsymbol{C}\frac{d}{dt}\mathrm{x}(t) \; + \; \mathbf{G}\; \mathbf{x}(t) \; = \; \mathbf{B}\; \mathbf{u}(t) \tag{3.15}$$

2. Let $\mathbf{u}(t)$ be given by the delta function $\mathbf{u}(t) = \delta\ (t)$ and let the response of the system be denoted by $\mathbf{h}(t)$.

3. Generate the trajectory traced by $\mathbf{h}(t)$ for N points $t\ =\ t_0, t_1, ..., t_{N-1}$.

4. Assign $\mathbf{x}\ =\ \{h(t_0), h(t_1), ..., h(t_{N-1}\}$ and pass $\mathbf{x}$ to algorithm 7 to find the optimum order for the reduced-order model.

The steps outlined above can be implemented, but perhaps with some difficulties. The main difficulty in the above steps is generating the time-domain trajectory of the system impulse response since the impulse response is a response to a delta function which is a mathematical abstraction rather than a concrete mathematical function that can be easily programmed in a computer-based implementation.

To overcome this difficulty, the order selection algorithm opts for approximating the impulse response with the response to a Gaussian pulse of the form

$$\mathbf{u}(t) = exp\left[\frac{-1}{2}(t/\tau)^2\right] \tag{3.16}$$

In order for the above Gaussian to approximate the delta function, the parameter τ needs to be assigned increasingly smaller values.

In addition, using a Gaussian pulse as the stimulus acts as a low-pass filter, which dampens the high frequency content of the impulse response. Although this maybe desirable, in the sense that one only needs a reduced model that matches the original model up to certain frequency, the passband of the filter needs to be closer to being ideal in order that the reduced model have a frequency response resembling the frequency response of the original model through the passband. Figure 3.4, shows the frequency domain picture of the Gaussian pulse with $\tau\ =\ 1e-7$.

In the next chapter, the thesis proposes a new technique that is aimed at resolving the

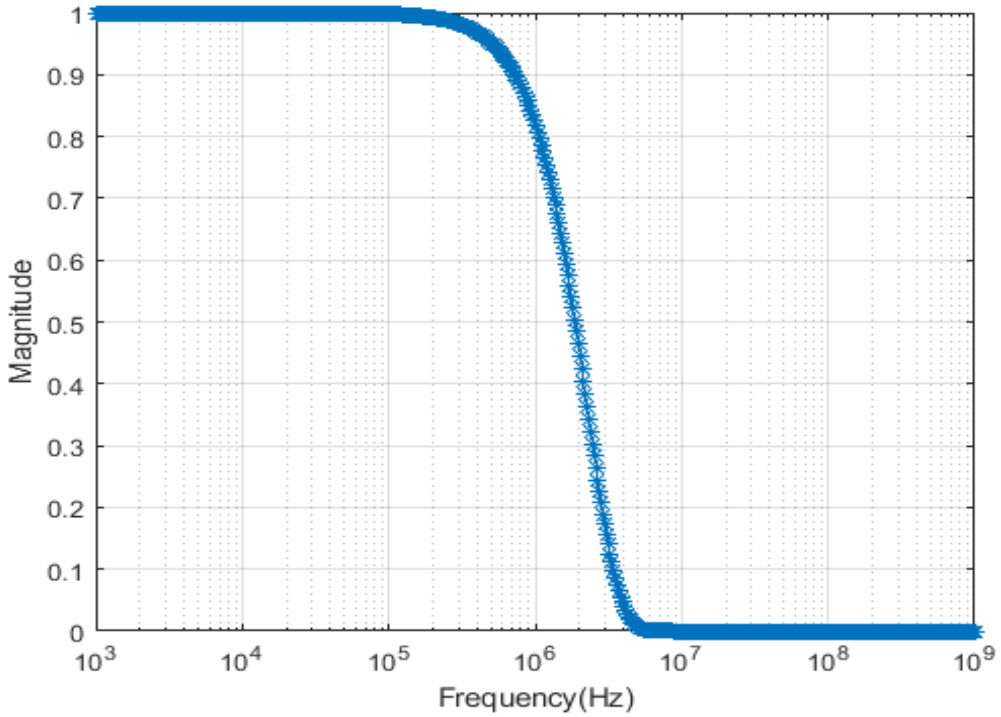

Figure 3.4: Gaussian frequency response

issue of what stimulus function to use as the input with which to drive the order selection algorithm.

Chapter 4

Proposed Approach: NILT-based Order Selection

The previous chapters exposed the main ideas of the concept of MOR. The selection of the most suitable order for the reduced model is one major step that remained an open issue until it was addressed rigorously in the work of [37, 38].

In the work proposed in [37, 38], the network to be reduced is first simulated in the time-domain with a Gaussian pulse as the driving source. The main underlying idea is to approximate the impulse response of the original network, by considering its response to the Gaussian pulse. Although this is a valid idea, the results obtained are still an approximation.

The main objective of this chapter is to propose an alternative technique based on the idea of Numerical Inversion of the Laplace Transform (NILT). To prepare the background work, section 4.1 describes the concept of NILT in a very simple context. Section 4.2 then considers applying it in the context of generating the impulse response of the large linear circuit. Section 4.3 discusses a certain difficulty that arises when the order selection algorithm is used directly on the results generated by the NILT technique.

To overcome these difficulties, Section 4.4 proposes a new idea using the notion of the Butterworth filtering methodology and demonstrates its success in handling the problem encountered in Section 4.2.

4.1 NILT: Numerical Inversion of the Laplace Transform

4.1.1 Development of the method

NILT is based on deriving the time-domain response of a circuit by inverting its Laplace domain version using a numerical-based method.

To introduce NILT assume that a function $\mathbf{v}(t)$ is only known by its Laplace-domain representation $\mathbf{V}(s)$, where

$$\mathbf{V}(s) = \int_0^\infty \mathbf{v}(t)\ e^{st} dt \tag{4.1}$$

Assuming that $\mathbf{V}(s)$ is already known, $\mathbf{u}(t)$ can then be obtained by the following inversion formula

$$\mathbf{v}(t) = \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} \mathbf{V}(s)\ e^{st} ds \tag{4.2}$$

The following presentation follows the description of NILT introduced in [61].

Exact inversion is possible only if the poles of $\mathbf{V}(s)$ are known. This could be done by

1. Finding a rational approximation for $\mathbf{V}(s)$, and
2. Using a root-finding procedure on the denominator of the above rational approximation of $\mathbf{V}(s)$ to compute its poles.

and then repeating the above process for each new problem. The above process, however, becomes intractable for large networks. To avoid this process, NLIT uses an alternate approach by representing the integrand in (4.2) by approximating the exponential function e^{st} by its Padè approximation.

The first step to do this is to remove the variable t from e^{st} by using the transformation

$$z = st \tag{4.3}$$

and then approximating e^z. This is done numerically *only once* in the development of the method, and all further inversions use the results thus obtained. Inserting (4.3) into (4.2) we obtain

$$\mathbf{v}(t) = \frac{1}{2\pi jt} \int_{ć-j\infty}^{ć+j\infty} \mathbf{V}(z/t)\ e^z dz \tag{4.4}$$

Next, we approximate the function e^z by a rational function (Padè approximation)

$$e^z \approx R_{N,M}(z) = \frac{P_N(z)}{Q_M(z)} \tag{4.5}$$

where $P_N(z)$ and $Q_M(z)$ are polynomials of degrees N, and M, respectively. The approximation formally equates a rational function to some terms of a series

$$\frac{\sum_{i=0}^{N} a_i z^i}{1\ +\ \sum_{i=1}^{M} b_i z^i} = \sum_{i=0}^{M+N} c_i z^i + \sum_{i=M+N+1}^{\infty} c_i z^i \tag{4.6}$$

and in our case, the coefficients c_i for $i \leq M+N$ are the coefficients of the Taylor expansion of e^z. Therefore, the Padè approximation $R_{N,M}(z)$ has the first $M+N+1$ terms of its Taylor expansion equal to the Taylor expansion of e^z. However, the expansions differ in the remaining terms.

It is not necessary to solve the system of equations arising from (4.5) since a closed-form

exists [62],

$$R_{N,M} = \frac{P_N(z)}{Q_M(z)} = \frac{\sum_{i=0}^{N}(M+N-i)! \begin{pmatrix} N \\ i \end{pmatrix} z^i}{\sum_{i=0}^{M}(-1)^i(M+N-i)! \begin{pmatrix} M \\ i \end{pmatrix} z^i} \tag{4.7}$$

Several of the first approximations are shown in Table 4.1.1. The poles of all approximations are simple and, for M not differing considerably from N, all are in the right half plane. We will need this fact in the following simplified explanation of the theory; it is valid even if they are in the left half plane. Inserting (4.5) into (4.4), the approximation $\hat{\mathbf{v}}(t)$ to $\mathbf{v}(t)$ becomes

$$\hat{\mathbf{v}}(t) = \frac{1}{2\pi jt} \int_{ć-j\infty}^{ć+j\infty} \mathbf{V}(z/t) R_{N,M}(z) dz \tag{4.8}$$

Now the integral in (4.8) can be evaluated by residue calculus by closing the left path of integration along an infinite arc either to the right or to the left. In order that the path along the infinite arc not contribute to the integral, choose M, N such that the function

$$\mathbf{F}(z) = \mathbf{V}(z/t) R_{N,M}(z) \tag{4.9}$$

has at least two more finite poles than zeros.

Then, using the residues theorem which states that (see Appendix A)

$$\int_C F(z)dz = \pm 2\pi j \sum \text{residues of } F(z) \text{at poles inside the closed path} \tag{4.10}$$

where the positive sign applies when the path C is closed in the left half plane(counterclockwise), whereas the negative one applies for the other case. For $N < M$ we have (using a partial

Table 4.1: Padè Table for the Approximation of e^z.

M \ N	0	1	2
0	$\frac{1}{1}$	$\frac{1+z}{1}$	$\frac{1+z+z^2/2}{1}$
1	$\frac{1}{1-z}$	$\frac{1+z/2}{1-z/2}$	$\frac{1+2z/3+z^2/6}{1-z/3}$
2	$\frac{1}{1-z+z^2/2}$	$\frac{1+z/3}{1-2z/3+z^2/6}$	$\frac{1+z/2+z^2/18}{1-z/2+z^2/18}$

fraction representation of $R_{N,M}(z)$)

$$R_{N,M}(z) = \sum_{i=1}^{M} \frac{K_i}{z - z_i} \tag{4.11}$$

where z_i are the poles of $R_{N,M}(z)$ and K_i are the corresponding residues. Closing the path of integration around the poles of $R_{N,M}(z)$ *in the right half plane* we obtain the following approximation for $\mathbf{v}(t)$

$$\mathbf{v}(t) \approx \hat{\mathbf{v}}(t) = -\frac{1}{t} \sum_{i=1}^{M} K_i \mathbf{V}\left(\frac{z_i}{t}\right) \tag{4.12}$$

This is the basic inversion formula for the Laplace transform. Real-time functions can be evaluated using only the poles z_i in the upper half plane. This reduces the computations to one half. For even integer values of M (the bar denoting complex conjugate), we have

$$\begin{aligned}
\hat{\mathbf{v}}(t) &= -\frac{1}{t} \sum_{i=1}^{M'} K_i \mathbf{V}(z_i/t) - \frac{1}{t} \sum_{i=1}^{M'} \bar{K}_i \mathbf{V}(\bar{z}_i/t) \\
&= -\frac{1}{t} \sum_{i=1}^{M'} 2\Re\left\{K_i \mathbf{V}\left(\frac{z_i}{t}\right)\right\} \\
&= -\frac{1}{t} \sum_{i=1}^{M'} \Re\left\{K_i' \mathbf{V}\left(\frac{z_i}{t}\right)\right\}
\end{aligned} \tag{4.13}$$

where $M' = M/2$ and $K_i' = 2K_i$, K_i being defined by the partial fraction (4.11). When

M is odd, $M' = (M+1)/2$ and $K'_i = K_i$ for the residue corresponding to the real pole.

The process of applying the NILT formula (4.13) to the formula can be summarized as follows.

Let the function $\mathbf{V}(s)$ be given. We wish to know the time domain response at time t:

1. Select appropriate N, M and take z_i, K'_i from the table of poles and residues.
2. Divide each z_i by t and substitute (z_i/t) for each s in $\mathbf{V}(s)$. This provides $\mathbf{V}(z_i/t)$.
3. Multiply each $\mathbf{V}(z_i/t)$ by K'_i and add the products.
4. Retain only the real part and divide by -t.

The above steps give the approximation to $\mathbf{v}(t)$ at the time t.

4.1.2 Application of the NILT method in linear networks

Here, the application of the NILT method to computing the time-domain response of a general linear network is considered. As described earlier, a general linear network is typically described by the MNA formulation in the time-domain as

$$\mathbf{C}\frac{d\mathbf{x}}{dt} + \mathbf{G}\mathbf{x}(t) = \mathbf{B}\mathbf{u}(t) \tag{4.14}$$

The NILT approximation formula (4.13) first requires that the Laplace-domain version for $\mathbf{x}(t)$ be evaluated at specific points in the Laplace-domain. Denoting the Laplace domain version of $\mathbf{x}(t)$ by $\mathbf{X}(s)$, one can write the above formulation in the Laplace domain as

$$(s\mathbf{C} + \mathbf{G})\mathbf{X}(s) = \mathbf{B}\mathbf{U}(s) - \mathbf{C}\mathbf{x}(0^+) \tag{4.15}$$

where $\mathbf{x}(0^+)$ is a vector containing the inital conditions for $\mathbf{x}(t)$ at $t = 0^+$. To compute $\mathbf{X}(s)$, we use the following

$$\mathbf{X}(s) = (\mathbf{G} + s\mathbf{C})^{-1} (\mathbf{B}\mathbf{U}(s) - \mathbf{C}\mathbf{x}(0^+)) \tag{4.16}$$

The development of the NILT method in the previous subsection illustrated that in order to approximate $\mathbf{x}(t)$, $\mathbf{X}(s)$ has to be computed at specific values for the Laplace variable s, namely those values that correspond to the poles of the (N, M) rational approximation of the exponential function e^z in z, where $z = st$. Using the notion described in Section 4.1 it then follows that $\mathbf{x}(t)$ is approximated using,

$$\mathbf{x}(t) \approx \hat{\mathbf{x}}(t) = -\frac{1}{t} \sum_{i=1}^{M'} \Re \left\{ K_i \left(\mathbf{G} + \frac{z_i}{t}\mathbf{C} \right)^{-1} \left[\mathbf{B}\mathbf{U} \left(\frac{z_i}{t} \right) + \mathbf{C}\mathbf{x}(0^+) \right] \right\} \tag{4.17}$$

Assuming that it is required to compute $\mathbf{x}(t)$ at discrete equi-distant timepoints $t = nh$, with $n = 1, 2, 3, \cdots$, where h is the length of the time step, (4.17) takes the following form

$$\hat{\mathbf{x}}(nh) = -\frac{1}{nh} \sum_{i=1}^{M'} \Re \left\{ K_i' \left(\mathbf{G} + \frac{z_i}{nh}\mathbf{C} \right)^{-1} \left[\mathbf{B}\mathbf{U} \left(\frac{z_i}{nh} \right) + \mathbf{C}\hat{\mathbf{x}}(0^+) \right] \right\} \tag{4.18}$$

It is obvious from the preceding discussion, that approximating the time-domain response at a given instant t, requires solving a number of matrix equations equal to M', where $M' = M/2$ for even M or $(M + 1)/2$ for odd M. Moreover, if $\mathbf{x}(t)$ is required at, say Q, time points, $t_0, t_1, ...t_{Q-1}$, then the matrix $\left(\mathbf{G} + \frac{z_i}{t_q}\mathbf{C} \right)^{-1}$ has to be inverted $M'Q$ times for $t = t_0, \cdots, t_{Q-1}$.

Typically, the inversion of the matrix is carried out by using the LU factorization algorithm which scales almost linearly when the matrix has very large dimension and very sparse (few non-zero entries) structure. Following the LU factorization of the matrix $\left(\mathbf{G} + \frac{z_i}{t_j}\mathbf{C} \right)$ for $i = 1, ...M'$ and $j = 0, 1, ...Q - 1$ a process known as the for-

ward/backward substitution involving the right-side vector, which in this case is given by $\left[\mathbf{BU}\left(\frac{z_i}{t_j}\right) + \mathbf{Cx}(0^+)\right]$, is carried out.

In an effort to reduce the computational effort of factorizing a large number of matrices that may be needed, a reinitialization procedure is utilized. In the reinitialization procedure, the origin of the time axis shifted by h, where h is typically the size of time step that is kept constant from step to step, i.e,

$$h = t_{m+1} - t_m, \quad m = 1, 2, \cdots \tag{4.19}$$

It can be shown that the shifting of the time origin by h enables recasting the NILT formula (4.20) as

$$\hat{\mathbf{x}}(nh) = -\frac{1}{h}\sum_{i=1}^{M'} \Re\left\{K_i'\left(\mathbf{G} + \frac{z_i}{h}\mathbf{C}\right)^{-1}\left[\mathbf{BU}\left(\frac{z_i}{h}\right) + \mathbf{C}\hat{\mathbf{x}}((n-1)h)\right]\right\}, \quad n = 1, 2, \cdots, Q-1 \tag{4.20}$$

As can be seen from the above equation, the required number of LU factorization drops from $M'Q$ to only M', since one needs to factorize the matrix $\left(\mathbf{G} + \frac{z_i}{h}\mathbf{C}\right)$ only $\acute{M}$ times. This is again followed by Q forward/backward substitution processes.

4.1.3 Advantages of NILT

Although the NILT approach is applicable only to linear circuits, owing to its essential reliance on the Laplace transform, it provides important advantages over conventional time-domain approaches. It is to be noted here that "conventional time-domain approaches" refers to the numerical solvers of differential equations employed in the time-marching methods and used in transient circuit simulations. Time marching or time stepping methods are generally used to approximate the solution, $\mathbf{x}(t)$ to a system of differential equation of the form

$$\mathbf{f}\left(\mathbf{x}(t), \frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t}, t\right) = 0 \tag{4.21}$$

i.e, $\mathbf{x}(t)$ at discrete time points in t, e.g. $t_0, t_1, t_2, \cdots$.

The main and foremost advantage of NILT is that it can be made to approximate $\mathbf{x}(t)$ to the arbitrary high degree of accuracy without risking to run into numerical instability or restricting the length of the time step. Although a thorough exposition of the issues related to high accuracy (commonly known as high-order) and numerical stability is beyond the scope of this work, it will suffice here to state the necessary and sufficient conditions that guarantee the numerical stability of NILT. The interested reader may refer to the text in [22, 23] for an in-depth discussion of those issues

The necessary and sufficient conditions for stability may be stated as follows. Let $\hat{\mathbf{x}}(t)$ be the NILT generated approximation to the exact solution $\mathbf{x}(t)$. Further, assume that $\hat{\mathbf{x}}(t)$ matches $\mathbf{x}(t)$ exactly at $t = t_n$, that is

$$\hat{\mathbf{x}}(t_n) = \mathbf{x}(t_n) \tag{4.22}$$

The underlying theory of the NILT approximation states that for an approximation obtained at $t = t_n + h$ using NILT, it follows that

$$\hat{\mathbf{x}}(t_n + h) = \mathbf{x}(t_n + h) + O(h^{N+M+2}) \tag{4.23}$$

Further, the successive approximations are stable (in the numerical sense) if the integers N and M and only if they satisfy

$$M - 2 \leq N \leq M \tag{4.24}$$

The above result in (4.23) indicates that starting with an exact value at $t = t_n$, and by

taking a time step of h, the error of approximating $\mathbf{x}(t_n + h)$ is proportional to h^{M+N+2}. Hence the error can be made arbitrarily small by selecting N and M as desired.

On the other hand, to preserve the numerical stability, the values of N and M must be restricted to satisfy (4.24). Note here that this restriction does not restrict the order of approximation that is proportional to $N + M + 2$.

It is important to stress that numerical instability arises from the fact that the error at a single step (which is unavoidable) gets amplified at the next step. Hence, numerical stability refers to the ability of the method to dampen such inevitable error from step to step and prevent it from growing.

4.2 NILT Approximation of the Impulse Response

The idea of using NILT to approximate the impulse response of a large linear network is described in this section. This is a fundamental step towards presenting the main contribution in this thesis, namely, using NILT in the procedure of order selection in the reduced-order model based on Krylov-subspace projections.

To begin with, we carry out the substitution

$$\mathbf{U}(s) = 1 \tag{4.25}$$

to capture the fact that the stimulus of the network in this context is the Dirac delta function $\delta(t)$, so that the response of the system is nothing other than the impulse response which in this case is denoted by $\mathbf{h}(t)$. Thus, we have[1]

$$\mathbf{G}\mathbf{h}(t) + \mathbf{C}\frac{\mathrm{d}\mathbf{h}(t)}{\mathrm{d}t} = \mathbf{B}\delta(t) \tag{4.26}$$

[1]It should be noted here that $\mathbf{h}(t)$, which is a vector collecting all the of impulse responses of all the networks variables, is not to be confused with h, which is a scalar value representing the constant step size used in time marching, as used in (4.19).

Let us now apply the NILT process described in the previous section to find an approximation $\hat{\mathbf{x}}(t)$ for $\mathbf{x}(t)$. Substituting from (4.25) into (4.17) we get

$$\hat{\mathbf{h}}(nh) = -\frac{1}{nh}\sum_{i=1}^{M'} \Re\left\{ K_i' \left(\mathbf{G} + \frac{z_i}{nh}\mathbf{C}\right)^{-1} \left[\mathbf{B} + \mathbf{C}\mathbf{h}(0^+)\right] \right\} \tag{4.27}$$

One of the immediate issues that arise from the above formulation is the computation of $\mathbf{h}(0^+)$, which represents the values of the components of $\mathbf{h}(t)$ at $t = 0^+$. This issue is addressed in the following subsection.

4.2.1 Computing $\mathbf{h}(0^+)$

We consider two possible situations, which depend on whether the matrix $\mathbf{C}$ is invertible or not. We handle the case of invertible $\mathbf{C}$ first.

4.2.1.1 Case I: The matrix C is invertible.

Let us integrate both sides of (4.28) from 0 to a general point of time t, we get

$$\mathbf{G}\int_0^t \mathbf{h}(\tau)\mathrm{d}\tau + \mathbf{C}\int_0^t \frac{\mathrm{d}\mathbf{h}(\tau)}{\mathrm{d}\tau}\mathrm{d}\tau = \mathbf{B}\int_0^t \delta(\tau)\mathrm{d}\tau \tag{4.28}$$

Proceeding with the following manipulation taking the limit as $t \to 0^+$, and noting that for causal systems $\mathbf{t} = 0$ for $t < 0$, we get

$$\mathbf{G}\underbrace{\lim_{t\to 0^+}\int_0^t \mathbf{h}(\tau)\mathrm{d}\tau}_{\to 0} + \mathbf{C}\underbrace{\lim_{t\to 0^+}\int_0^t \frac{\mathrm{d}\mathbf{h}(\tau)}{\mathrm{d}\tau}\mathrm{d}\tau}_{\mathbf{h}(0^+)} = \mathbf{B}\underbrace{\int_0^t \delta(\tau)\mathrm{d}\tau}_{1} \tag{4.29}$$

which yields

$$\mathbf{C}\mathbf{h}(0^+) = \mathbf{B} \tag{4.30}$$

and, with $\mathbf{C}$ being invertible, provides a way of computing $\mathbf{h}(0^+)$ as

$$\mathbf{h}(0^+) = \mathbf{C}^{-1}\mathbf{B} \tag{4.31}$$

4.2.1.2 Case II: The matrix C is non-invertible.

It is often the case that the capacitance matrix is singular and therefore non-invertible, with possibly zero rows and columns. Without loss of generality, we assume that $\mathbf{C}$ is block-structured in the following manner

$$\mathbf{C} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{22} \end{bmatrix} \tag{4.32}$$

where the block $\mathbf{C}_{22}$ is assumed to be invertible. In an analogous manner, we carry this structure to the rest of the MNA formulation, i.e., (4.28) is rewritten as

$$\begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} \\ \mathbf{G}_{21} & \mathbf{G}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{h}_1(t) \\ \mathbf{h}_2(t) \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{22} \end{bmatrix} \begin{bmatrix} \frac{\mathrm{d}\mathbf{h}_1(t)}{\mathrm{d}t} \\ \frac{\mathrm{d}\mathbf{h}_2(t)}{\mathrm{d}t} \end{bmatrix} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} \delta(t) \tag{4.33}$$

In the search for $\mathbf{h}(0^+)$, one could see that the first block of the impulse response, i.e., $\mathbf{h}_1(0^+)$ is actually not needed. This is because the NILT formula (4.27) requires multiplying $\mathbf{h}(0^+)$ by the matrix $\mathbf{C}$, and upon examining the assumed structure of that matrix in (4.32), it becomes clear that this block ($\mathbf{h}_1(0^+)$) is always multiplied by 0, and therefore should not be of any concern for the present task. Thus, the focus, for the time being, is placed on computing $\mathbf{h}_2(0^+)$.

The algebraic part represented by the first block row of (4.33) can be used to give $\mathbf{h}_1(t)$ in terms of $\mathbf{h}_1(t)$ as follows

$$\mathbf{h}_1(t) = \mathbf{G}_{11}^{-1}\left(\mathbf{B}_1\delta(t) - \mathbf{G}_{12}\mathbf{h}_2(t)\right) \tag{4.34}$$

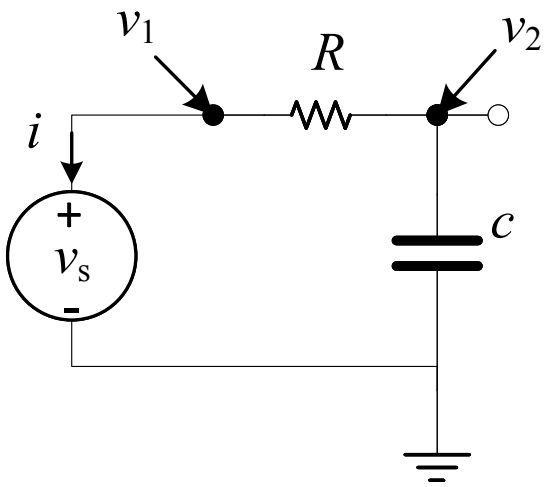

Figure 4.1: A simple test circuit used to validate the idea put in Section 4.2.1 for computing $\mathbf{h}(0^+)$.

Using (4.35) to substitute for $\mathbf{h}_1(t)$ in the second block row of (4.33) produces the following expression for $\mathbf{h}_2(t)$

$$\left(\mathbf{G}_{22} + \mathbf{G}_{21}\mathbf{G}_{11}^{-1}\mathbf{G}_{12}\right)\mathbf{h}_2(t) + \mathbf{C}_{22}\frac{\mathrm{d}\mathbf{h}_2(t)}{\mathrm{d}t} = -\mathbf{G}_{21}\mathbf{G}_{11}^{-1}\left(\mathbf{B}_1\delta(t)\right) + \mathbf{B}_2\delta(t) \tag{4.35}$$

Following the same integration procedure used in Section 4.2.1.1, the expression for $\mathbf{h}_2(0^+)$ is then obtained as

$$\mathbf{h}_2(0^+) = \mathbf{C}_{22}^{-1}\left[\mathbf{G}_{21}\mathbf{G}_{11}^{-1}\mathbf{B}_1 + \mathbf{B}_2\right] \tag{4.36}$$

4.2.2 Experimental Validation for Computing $\mathbf{h}(0^+)$

The goal in this section is to validate the procedure described in Section 4.2.1 for computing $\mathbf{h}(0^+)$ by comparing it in a special case in which $\mathbf{h}(0^+)$ can be written analytically in a closed-form. To this end, we consider the simple RC circuit shown in Figure 4.1.

It is straightforward to see that the Laplace-domain transfer function between the input

voltage, v_s, and the output voltage, v_2 is given by

$$H_2(s) = \frac{V_2(s)}{V_s(s)} = \frac{1}{1 + sRC} \tag{4.37}$$

Therefore, its time-domain version is given by

$$h_2(t) = \frac{1}{Rc} e^{-t/(Rc)} \tag{4.38}$$

and thus

$$h_2(0^+) = \frac{1}{Rc} \tag{4.39}$$

Now, the process outlined in Section 4.2.1 for computing $\mathbf{h}(0^+)$ is applied to the test circuit shown in Figure 4.1.

The matrices for the MNA formulation corresponding to the circuit in Figure 4.1 are as follows

$$\mathbf{G} = \begin{bmatrix} \frac{1}{R} & -\frac{1}{R} & 1 \\ -\frac{1}{R} & \frac{1}{R} & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{4.40}$$

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.41}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.42}$$

$$\mathbf{h}(t) = \begin{bmatrix} h_{v_1}(t) \\ h_{v_2}(t) \\ h_i(t) \end{bmatrix} \tag{4.43}$$

The MNA matrices in (4.40) through (4.43) can be cast in the block-structured format in (4.32) by observing that

$$\mathbf{G}_{11} = \begin{bmatrix} \frac{1}{R} & 1 \\ 1 & 0 \end{bmatrix} \tag{4.44}$$

$$\mathbf{G}_{12} = \begin{bmatrix} -\frac{1}{R} \\ 0 \end{bmatrix} \tag{4.45}$$

$$\mathbf{G}_{21} = \begin{bmatrix} -\frac{1}{R} & 0 \end{bmatrix} \tag{4.46}$$

$$\mathbf{G}_{22} = \begin{bmatrix} \frac{1}{R} \end{bmatrix} \tag{4.47}$$

$$\mathbf{C}_{22} = [c] \tag{4.48}$$

$$\mathbf{B}_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{4.49}$$

$$\mathbf{B}_2 = [0] \tag{4.50}$$

$$\mathbf{h}_1(t) = \begin{bmatrix} h_{v_1}(t) \\ h_i(t) \end{bmatrix} \tag{4.51}$$

$$\mathbf{h}_2(t) = \begin{bmatrix} h_{v_2}(t) \end{bmatrix} \tag{4.52}$$

Substituting from (4.44) through (4.52) into (4.36) gives the following

$$\mathbf{h}_2(0^+) = \frac{1}{Rc} \tag{4.53}$$

Comparing the result obtained in (4.53) with the analytical expression in (4.38) at $t = 0^+$ proves the validity of the suggested procedure outlined in Section 4.2.

To verify that the numerical results obtained from the NILT formula (4.27) agree with the analytical expression in (4.38), Figure 4.2 compares the two results, indicating full agreement.

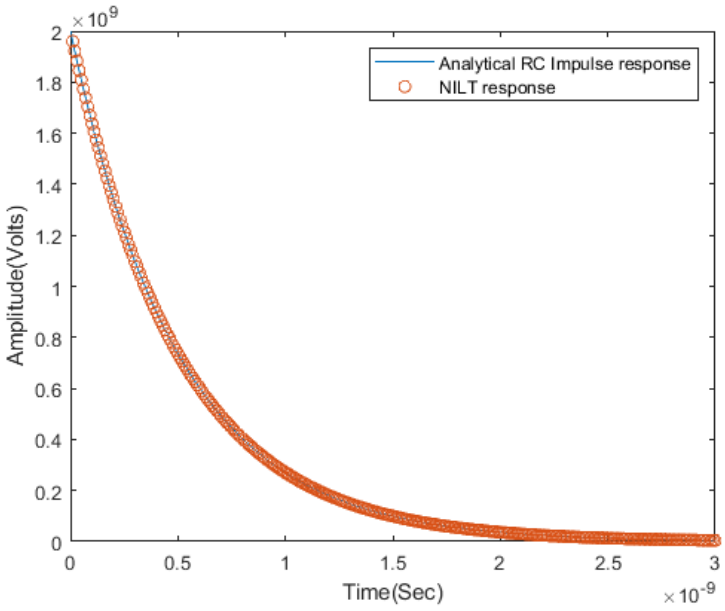

Figure 4.2: Comparing the numerical result obtained from NILT (4.27) with the analytical expression (4.38) $R = 100\Omega$ and $c = 5\text{pF}$.

The verification presented in this section, therefore, provides enough ground for the soundness of reasoning presented in Section 4.2.1.

4.3 Preliminary Results on NILT-based Order Selection

This section presents the preliminary results of applying the core idea proposed by this thesis. Precisely, we consider a large circuit with an original size $N = 2412$ obtained from discretizing an electrically long transmission line using RLC lumped segmentation. Next, the following steps are applied in sequence.

1. The initial conditions of the impulse response $\mathbf{h}(0^+)$ are computed as illustrated in Section 4.2.1.
2. The NILT formula in (4.27) is applied to compute the $\mathbf{h}(nh)$, for $n = 1, 2, 3, \cdots$.
3. The sequence of $\mathbf{h}(nh)$ computed above is supplied as the input to Algorithm-7.
4. The number of FNN reported by the algorithm is monitored until it drops to 0, at which point the required appropriate size of the reduced system is determined.
5. A Krylov subspace projection is performed on the original system with the size determined from the previous step is being used.

Figure 4.3 shows the trace of the number of FNN as the algorithm progresses indicating that this number drops to zero at a value of m around 1600. Figure 4.4 provides a zoomed in view around this number for greater clarification.

This result therefore suggests that to construct a reduced-order system a size of $m = 1590$ is needed.

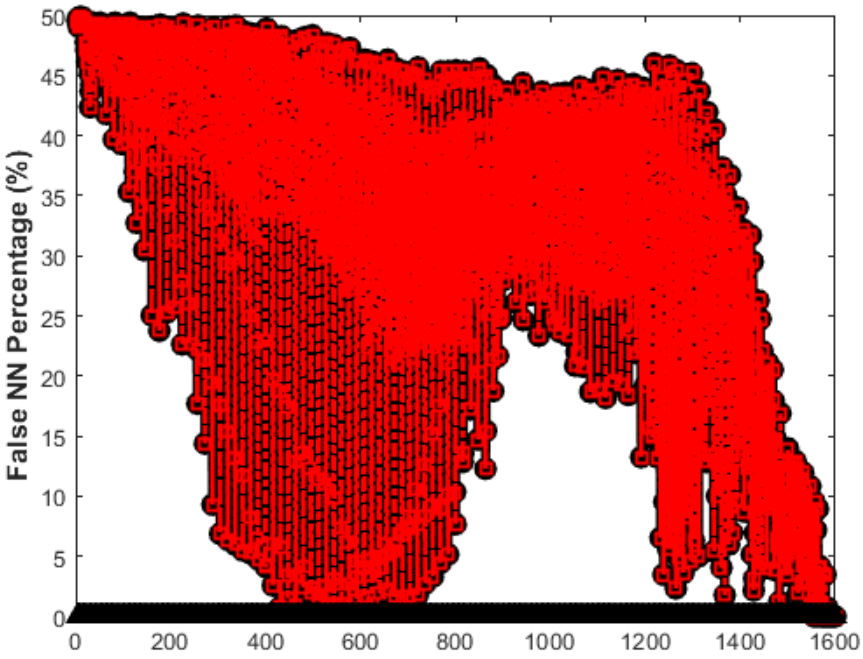

Figure 4.3: The number of FNN produced by Algorithm 7 when supplied with $\mathbf{h}(nh)$, for $n = 1, 2, 3 \cdots$.

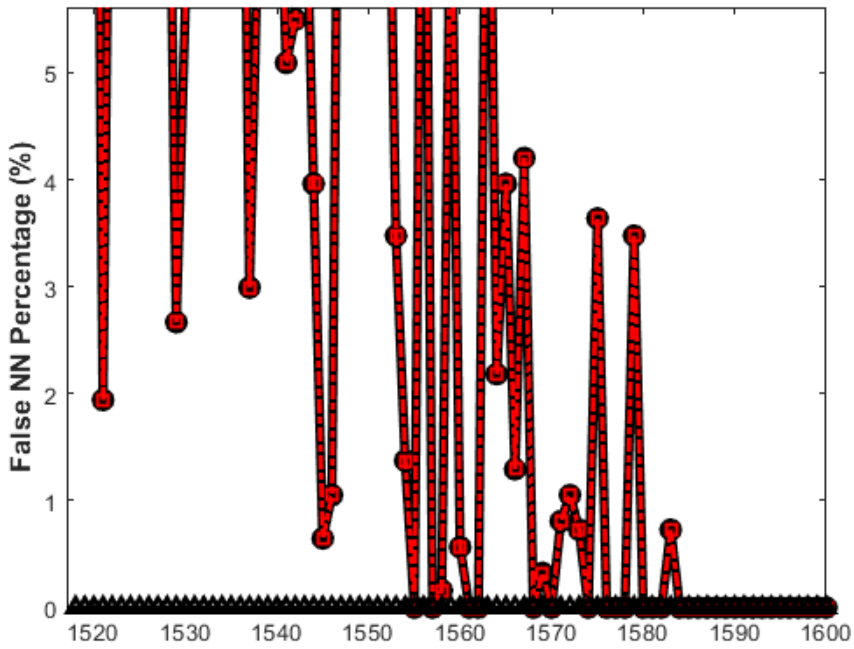

Figure 4.4: A zoom-in on the FNN obtained towards the end in Figure 4.3.

To test the soundness of this result, in the sense of having produced a reduced-model with the suitable size, Algorithm 7 was run again, but this time with a conventional Gaussian impulse with $\tau = 1e-11$ (of the form given by (3.16)) function used as a stimulus for the original network. Again, as in the previous case, the number of FNN is monitored and Figure 4.5, and its zoom-in in Figure 4.6 shows the progression in FNN indicating that the number of FNN drops to zero for a size $m = 378$.

The significant difference in reduced-model sizes as predicted by the above two methods (proposed and conventional) may suggest inefficiency or some source of error.

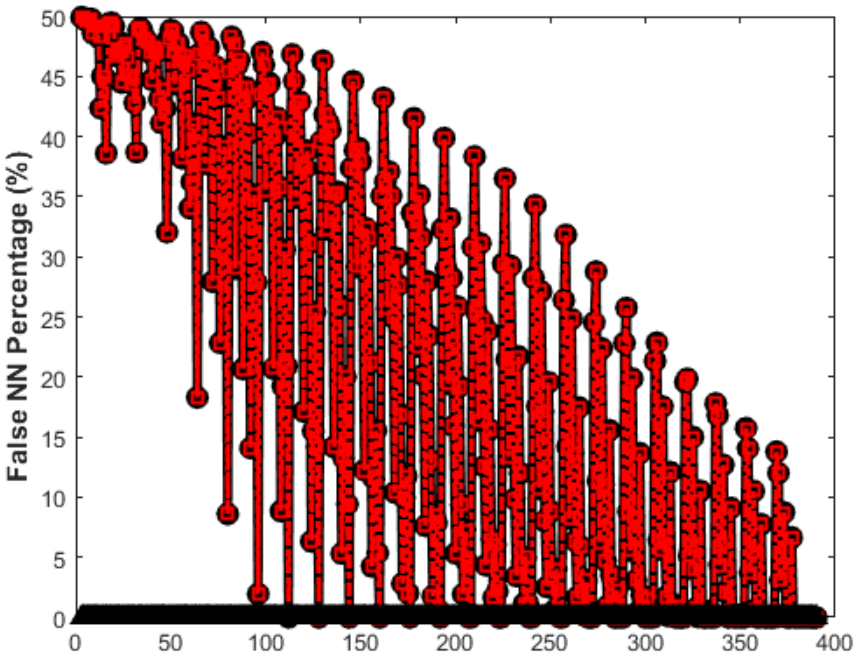

Figure 4.5: The number of FNN produced by Algorithm 7 when supplied with the response to a Gaussian impulse.

To illustrate the reason for this difference, the frequency response of the following three different syste ms

1. The reduced system with size 378 obtained through the Gaussian impulse.

2. The original system of size 2412

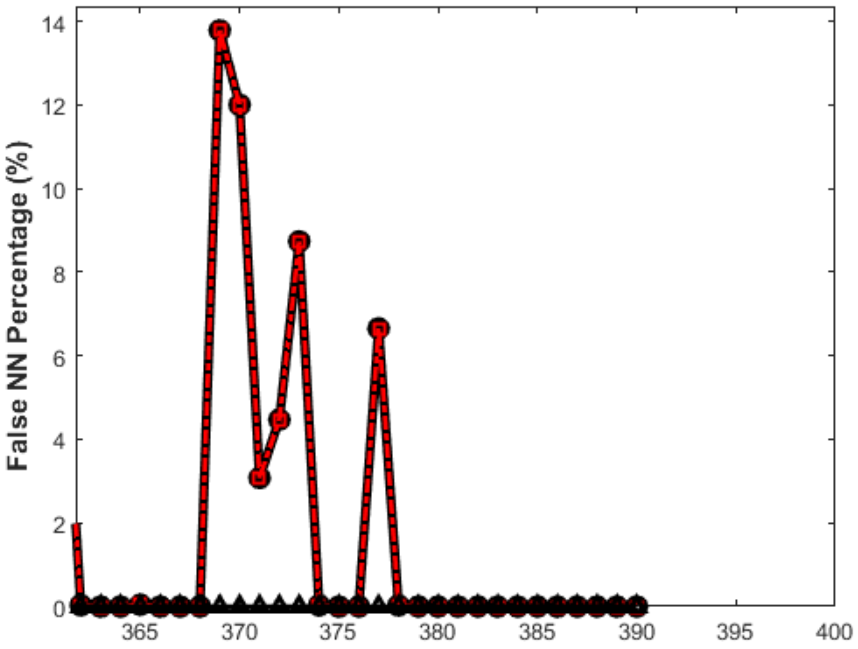

Figure 4.6: A zoom-in on the FNN obtained towards the end in Figure 4.5.

3. The reduced system with size 1600 obtained through the NILT-based computation of the impulse response.

Figure 4.7 shows the comparison between the frequency responses of the three systems. An important point to remark here is that the response of the NILT-based reduced-model matches the entire frequency response of the original system. On the other hand, the frequency response obtained from the reduced-order model whose order has been selected using the Gaussian stimulus matches only up to 10 GHz. To clearly demonstrate this observation, Figure 4.8 shows a zoom-in on the graphs in Figure 4.7.

It is possible now to state the reasons for the "inflated" order selected by the NILT-based impulse response. By operating on the pure impulse response obtained from NILT, the order selection algorithm was "forced" to create a reduced model that matches the entire frequency-domain behavior of the original system. On the other hand, the Gaussian-based order selection operated on a filtered version of the impulse response owing to the fact

that Gaussian pulse stimulus acts as low-pass filter that limits the frequency content of the original system that needs to be captured by the reduced system.

Given that, in practice, the goal of the reduced model is to match the frequency response up to a certain maximum frequency range, matching the entire frequency range (as may happen with the NILT-approximated impulse response) represents an unnecessary waste since it leads to an inflated reduced-order model. Nonetheless, the above experiment was useful in the sense that the insight it afforded provided the information needed to adjust the focus of the thesis to address this issue.

The focus at this point onward is to keep the advantages of using the NILT-generated impulse response to do the order selection, while at the same time filter out high frequency contents from that impulse response. To achieve this goal, the thesis delved into ideas from the theory of digital filtering. Precisely, the idea of using a Butterworth filter received the main focus in the research conducted in this thesis. The following section elaborates further on the basic concepts used to construct Butterworth filters.

4.4 Butterworth Filter

4.4.1 Introduction

Generally speaking, there are three main considerations in designing a filter circuit. They are:

1. The response of the pass band must be maximum flatness.
2. There must be a slow transition from pass band to the stop band.
3. The ability of the filter to pass signals without any distortions within the pass band.

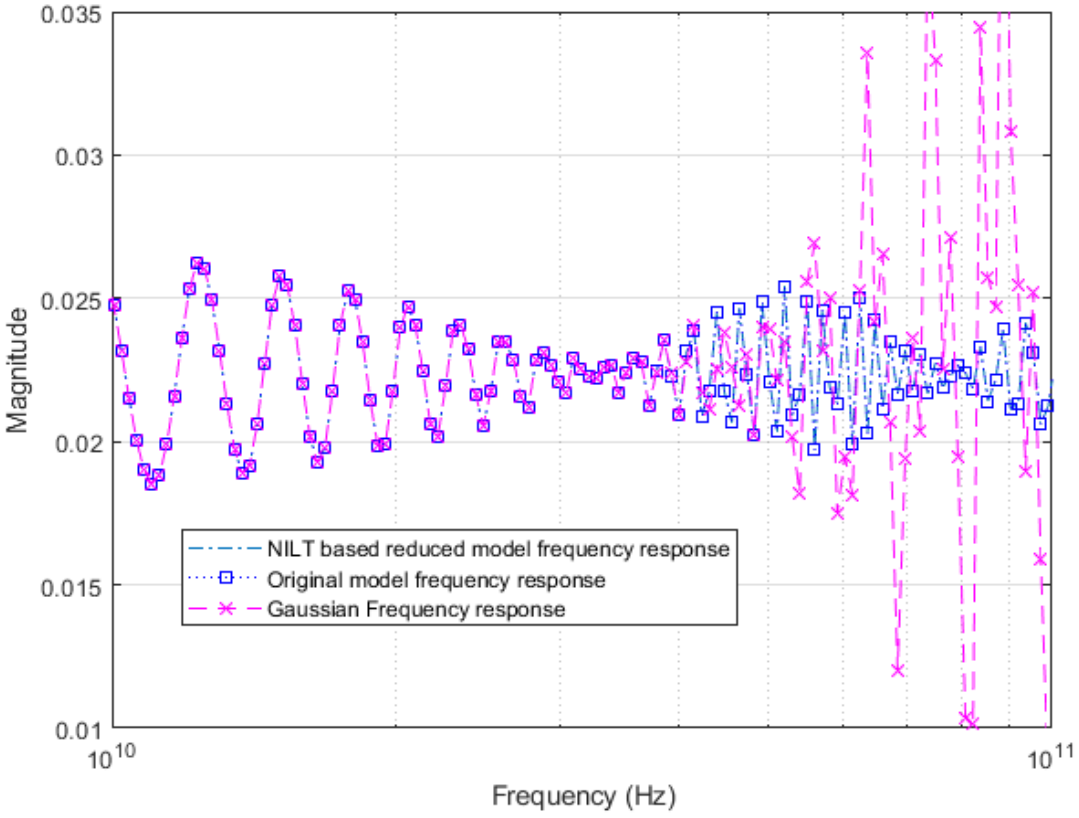

Figure 4.7: Comparing the frequency response for three different systems.

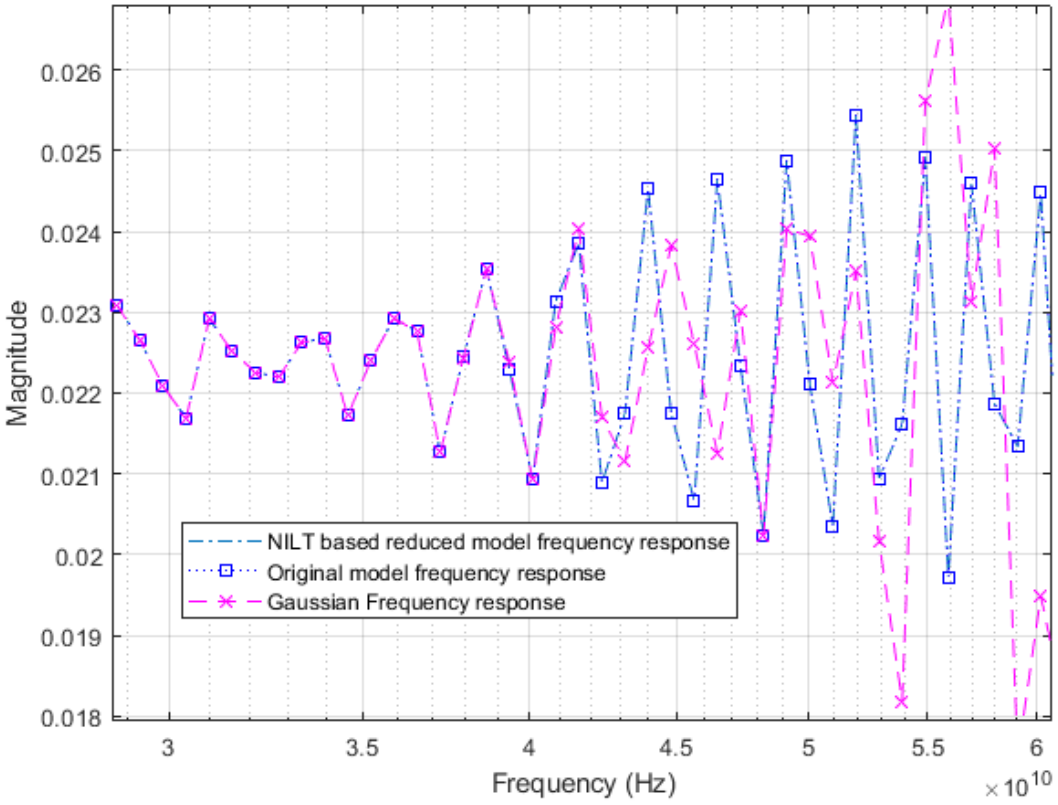

Figure 4.8: A zoomed-in view of the graph in Figure 4.7.

These distortions are generally caused by the phase shifts of the waveforms. In addition to these three the rising and falling time parameters also play an important role. By taking these considerations, a filter is built for individual consideration. For maximum flat response the Butterworth filter is designed. For a slow transition from pass band to stop band the Chebyshev filter is designed and for the maximum flat time delay Bessel filter is designed. But, for the problem in our case, the fast transition from pass band to stop band and maximum flat response in the pass band is needed. Hence, Butterworth filter best suits the problem.

4.4.2 Butterworth Filter

At the expense of steepness in transition medium from pass band to stop band (which is an advantage in our problem), this Butterworth filter will provide a flat response in the output signal. So, it is also referred as a maximally flat magnitude filter. The rate of falloff response of the filter is determined by the number of poles taken in the circuit. The number of poles will depend on the number of the reactive elements in the circuit that is the number of inductors or capacitors used in the circuits.

The amplitude response of the n^{th} order Butterworth filter is given as follows:

$$\frac{V_{out}}{V_{in}} = \frac{1}{\sqrt{1+(f/f_c)^{2n}}} \tag{4.54}$$

where 'n' is the number of poles in the circuit. As the value of the 'n' increases, the flatness of the filter response also increases. f is the operating frequency of the circuit and f_c is the center frequency or cut off frequency of the circuit.

These filters have pre-determined considerations whose applications are mainly at active RC circuits at higher frequencies. Even though it does not provide the sharp cut-off response it is often considered as the all-round filter which is used in many applications.

As we know that to meet the considerations of the filter responses and to have approximations near to ideal filter we need to have higher order filters. This will increase the complexity. We know the output frequency response and phase response of low pass and high pass circuits also. The ideal filter characteristics are maximum flatness, maximum pass band gain and maximum stop band attenuation.

To design a filter, proper transfer function is required. In order to satisfy these transfer function mathematical derivations are made in anolog filter design with many approximation functions. In such designs Butterworth filter is one of the filter types. Low pass Butterworth design considerations are mainly used for many functions.

4.4.3 First Order Low Pass Butterworth filter

The below circuit shows the low pass Butterworth filter

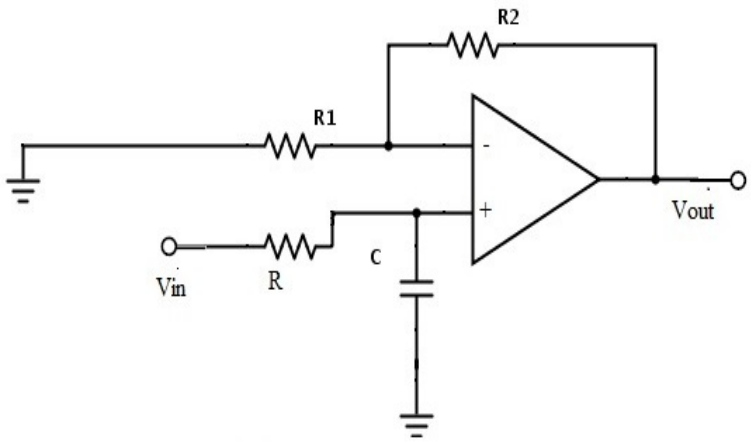

Figure 4.9: Impulse response

The required pass band gain of the Butterworth filter mainly depends on the resistor values of '$R1$' and 'Rf' and the cut off frequency of the filter will depend on R and C elements in the above circuit.

The gain of the filter is given as:

$$A_{max} \;=\; 1 + \frac{R1}{Rf} \tag{4.55}$$

The impedance of the capacitor 'C' is given by the '$-jX_C$' and the voltage across the capacitor is given as,

$$V_C \;=\; \frac{-jX_C}{(R \;-\; jX_C)} \times V_{in} \;\; (X_C \;=\; \frac{1}{2\pi fc}, \text{ capacitve Reactance}) \tag{4.56}$$

The transfer function of the filter in polar form is given as

$$H(j\omega) \;= \mid \frac{V_{out}}{V_{in}} \mid \angle\phi \tag{4.57}$$

where the gain of the filter is

$$\frac{V_{out}}{V_{in}} \;=\; \frac{A_{max}}{\sqrt{1 + (f/f_H)^2}} \tag{4.58}$$

and phase angle is

$$\phi \;=\; -\tan^{-1}(\frac{f}{f_H}) \tag{4.59}$$

At lower frequencies (when the operating frequency is lower than the cut-off frequency, the pass band gain is equal to maximum gain.

$$\frac{V_{out}}{V_{in}} \;=\; A_{max} \text{ (constant)} \tag{4.60}$$

At higher frequencies (when the operating frequency is higher than the cut-off frequency, the gain is less than the maximum gain.

$$\frac{V_{out}}{V_{in}} \;<\; A_{max} \tag{4.61}$$

When the operating frequency is equal to the cut-off frequency the transfer function is equal to $A_{max}/\sqrt{2}$. The rate of decrease in the gain is 20dB/decade or 6dB/octave and can be represented in the response slope as -20dB/decade.

4.4.4 Second Order Low Pass Butterworth Filter

An additional RC network connected to the first order Butterworth filter gives us a second order low pass filter. This second order low pass filter has an advantage that the gain rolls-off very fast after the cut-off frequency in the stop band.

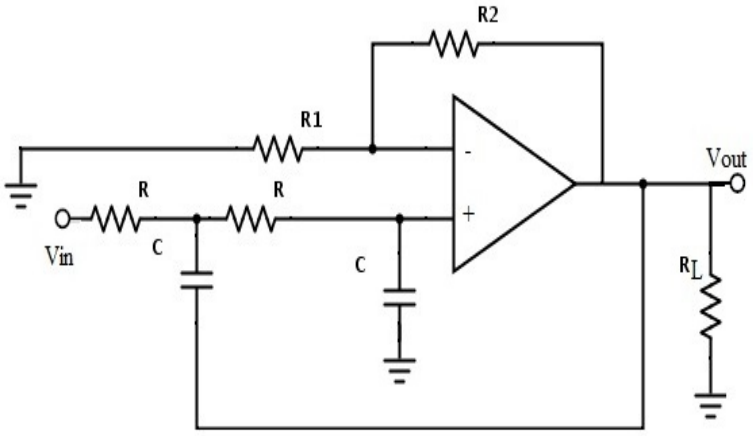

Figure 4.10: Impulse response

In this second order filter, cut-off frequency value depends on the resistor and capacitor values of two RC sections. The cut-off frequency is calculated using the formula below.

$$f_c = \frac{1}{2\pi\sqrt{R^2C^2}} \tag{4.62}$$

The gain rolls off at a rate of 40dB/decade and this response is shown in slope -40dB/decade. The transfer function of the filter can be given as

$$\frac{V_{out}}{V_{in}} = \frac{A_{max}}{\sqrt{1+(f/f_c)^4}} \tag{4.63}$$

The standard form of the transfer function of the second order filter is given as

$$\frac{V_{out}}{V_{in}} = \frac{A_{max}}{s^2 + 2\varepsilon\omega_n s + \omega_n^2} \tag{4.64}$$

where ω_n is the natural frequency of oscillations and ε is the damping factor.

Higher order Butterworth filters are obtained by cascading first and second order Butterworth filters. This can be done as shown in Figure 4.11.

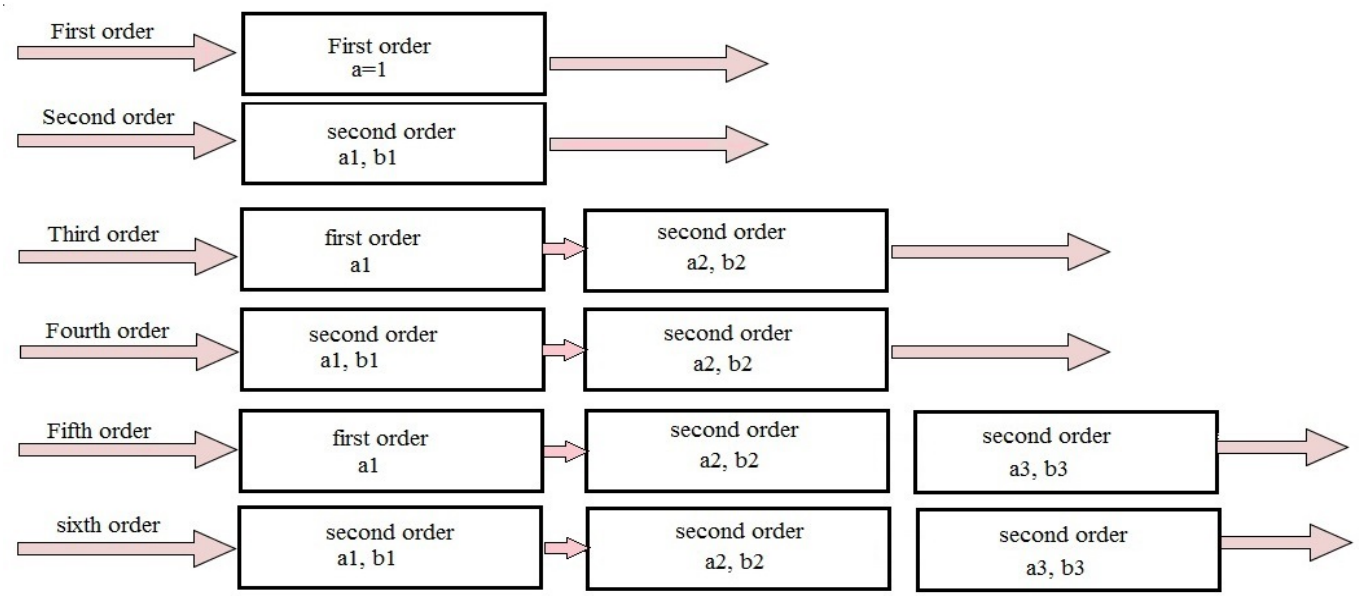

Figure 4.11: Constructing High-order Butterworth filters using first- and second-order filters.

where a_n and b_n are pre-determined filter coefficients and these are used to generate the required transfer functions.

4.4.5 Normalised low Pass Butterworth Filter Polynomials

Normalisation is a process in which voltage, current or impedance is divided by the quantity of the same unit of measure. This process is used to make a dimensionless range or level of particular value.

The denominator polynomial of the filter transfer function gives us the Butterworth polynomial. If we consider the s-plane on a circle with center at the origin, all the poles of the Butterworth filter are located in the left half of that s-plane.

For any order filter the co-efficient of the highest power of 's' should be always 1 and for any order filter the constant term is always 1. For even order filters all the polynomial factors are quadratic in nature. For odd order filters all the polynomials are quadratic except 1^{st} order. The normalized denominator polynomials are given as shown in Figure

n (order)	Normalized Denominator Polynomials in Factored Form
1	$(1+s)$
2	$(1+1.414s+s^2)$
3	$(1+s)(1+s+s^2)$
4	$(1+0.765s+s^2)(1+1.848s+s^2)$
5	$(1+s)(1+0.618s+s^2)(1+1.618s+s^2)$
6	$(1+0.518s+s^2)(1+1.414s+s^2)(1+1.932s+s^2)$
7	$(1+s)(1+0.445s+s^2)(1+1.247s+s^2)(1+1.802s+s^2)$
8	$(1+0.390s+s^2)(1+1.111s+s^2)(1+1.663s+s^2)(1+1.962s+s^2)$
9	$(1+s)(1+0.347s+s^2)(1+s+s^2)(1+1.532s+s^2)(1+1.879s+s^2)$
10	$(1+0.313s+s^2)(1+0.908s+s^2)(1+1.414s+s^2)(1+1.782s+s^2)(1+1.975s+s^2)$

Figure 4.12: The polynomials of the denominator in the Butterworth filter.

As stated previously, obtaining the optimum order for the network is the main objective of this thesis. In this regard, Chapter 3 discussed the method that is implemented in the thesis and the previous section gave information on how the time domain data is obtained. Hence, order selection algorithm is applied on the time domain data which is the impulse response of the network obtained using NILT.

We need an optimum order to obtain a reduced model which approximates the behaviour of the original network until certain frequency. Hence, the time domain data which is needed to apply the order selection algorithm must have frequency content only

up to a prespecified maximum frequency. This is because simply to use the impulse as the input for the network, is to excite every single frequency component of the network. Figure 4.13, shows the frequency response of a network of order 2412. We can clearly see that it is having frequency components of undesired range. Using this data for order selection algorithm makes the algorithm to give an optimum order with which the reduced model approximates the original system until the whole range of frequency spectrum of the original network.

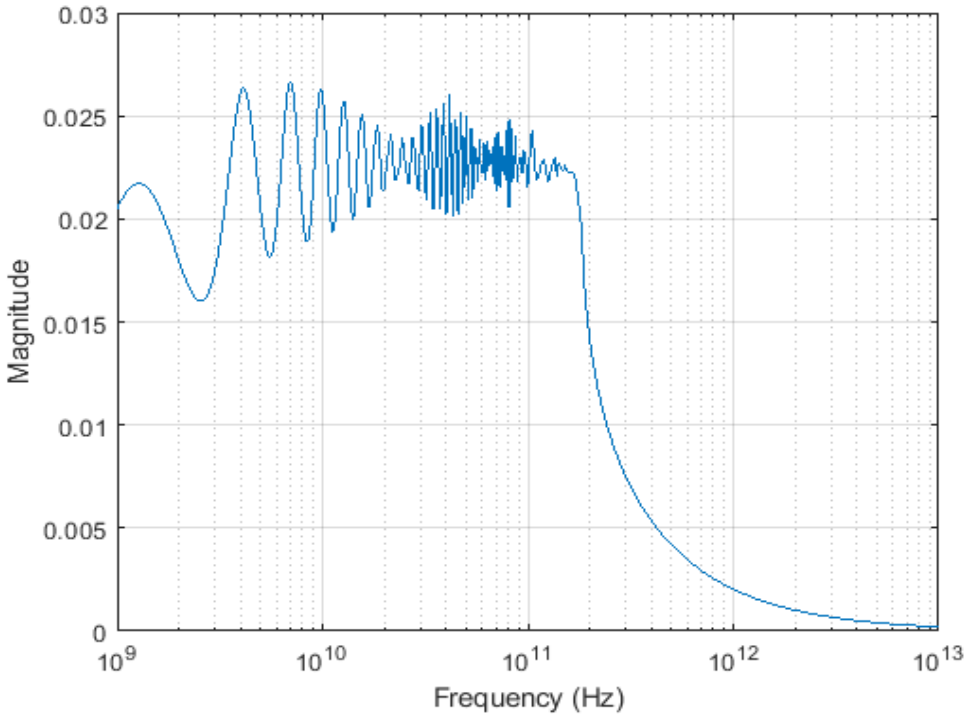

Figure 4.13: Frequency response of the network

Obtaining such an order is useless because we are simply reconstructing the original system back, as the optimum order given by the order selection algorithm is close to the order of the original network. Figure 4.3, shows the FNN graph for the time data directly obtained from the impulse response of the network. We can clearly see that the optimum order suggested by the algorithm is high and close to the original order of the network.

4.5 Applying Butterworth Filter on Impulse Response

The previous section, clearly illustrated the problem using the impulse response directly with the order selection algorithm. We now have the impulse response which is ideal and is surely better response than the response obtained using Gaussian pulse. But we need to restrict the frequency content in the impulse response in order to obtain a suitable order. In this regard, a Butterworth filter is applied on the impulse response of the network. The Butterworth filter is chosen for this purpose because it has no ripples in the passband which is desired because application of the filter should not alter the magnitude of the desired frequency components in this context. Moreover, using Butterworth filter allows a smooth and monotonically decreasing frequency response in the transition region. Figure 4.14, shows the frequency contents of the impulse response after applying the Butterworth filter in comparison with the frequency response of the network without applying Butterworth filter.

4.5.1 Using Matlab Toolbox

This section explains how the Butterworth filter is applied on the impulse response of the network. Owing to the fact that the response obtained in our problem is computed at regular time intervals (time step 'h') the digital filter implementation available in the MATLAB's Signal Processing Toolbox is used.

To filter any signal, firstly the transfer function co-efficients should be computed. In our case, we need to get co-efficients for a low pass filter because we clearly want to eliminate the higher frequency content of the signal. As low pass filter is commonly used, we need not specify any arguments to mention the low pass filtering mode. The command used to get the transfer function co-efficients is

[b,a] = butter(n,ω_n)

where, ω_n is the normalized cut-off frequency.

The normalised cut-off frequency is obtained from the desired cut-off frequency f_c and the sampling frequency f_s as

$$\omega_n = \frac{f_c}{(f_s/2)} \tag{4.65}$$

The sampling frequency in our problem is chosen as the inverse of the time step h.

After having the transfer function co-efficients, they are sent to the function

y = filter(b,a,x)

for filtering, where 'x' is the input data which is to be filtered.

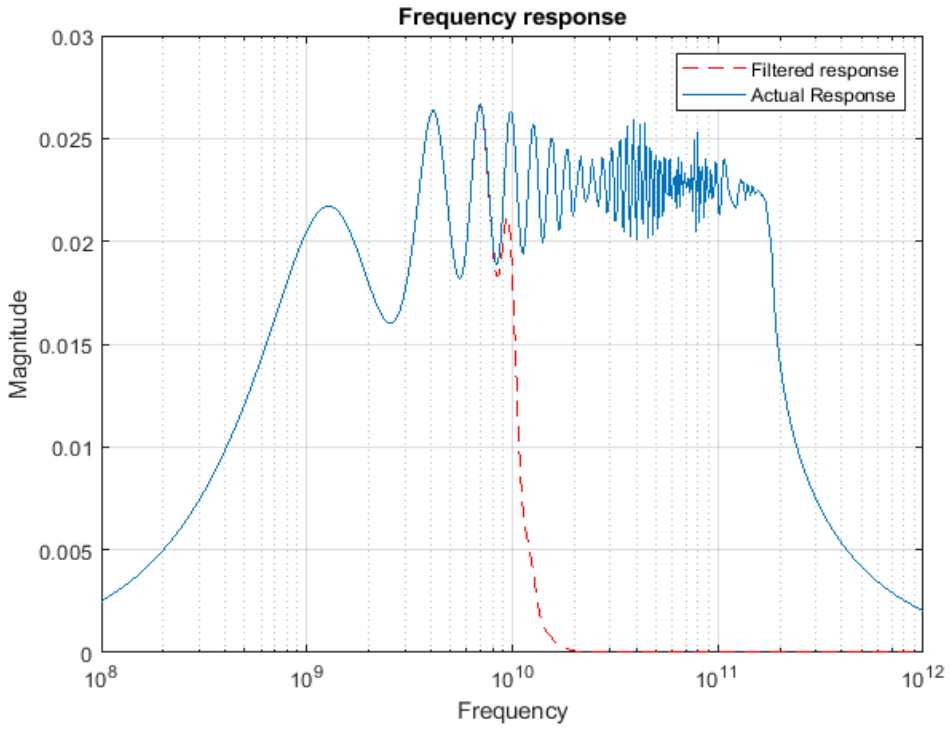

Figure 4.14: Comparison of the frequency response of the network with Butterworth filter response

Chapter 5

Numerical Examples

In this chapter, numerical results are presented to demonstrate the validity and accuracy of the proposed methodology.

5.1 Example - 1

In this example, the transmission line shown in Figure 5.1 is considered. The per-unit-length parameters are $C = 1.29pF/cm$, $L = 3.361nH/cm$ *and* $R = 5.421\Omega/cm$. The transmission line is discretized to the form of 1500 lumped RLGC π-sections in cascade using conventional lumped segmentation. The order of the subcircuit is 4500 excluding terminations. As discussed earlier, the input of the transmission line is set to impulse.

Applying the proposed method, Figure 5.2, shows the impulse response of the transmission line at the output port. The Figure 5.3, shows the filtered response of the response shown in Figure 5.2. The comparison of the frequency response of the impulse response and filtered response is shown in Figure 5.4, to validate the functioning of Butterworth filter. Figure 5.5, shows the count of the false nearest neighbors on the projected trajectory, while the dimension of the model is changed from m to $m + 1$. Figure 5.6, shows

the frequency response of original and reduced models, the order of the reduced model is chosen from Figure 5.5

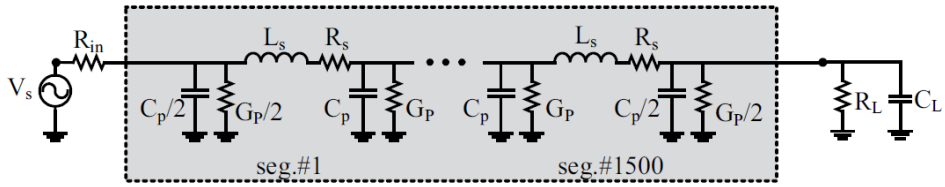

Figure 5.1: Transmission line modeled by 1500 lumped RLGC π-sections in cascade.

The Figure 5.2 shows the impulse response of the transmission line taken at the output port. The above figure clearly shows the behavior of a RLGC circuit with the damping oscillations and the reflections of the signal. By seeing the time scale, which is in 100's of pico seconds, we can clearly state that there are very high frequency components present in the response.

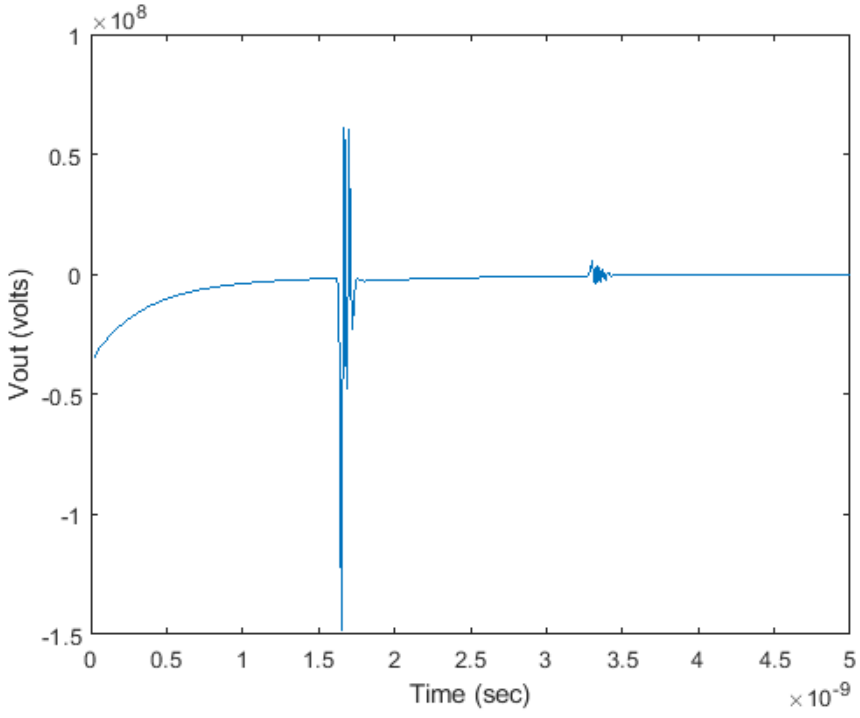

Figure 5.2: Impulse response of the transmission line

Figure 5.3, shows the filtered response of the transmission line which is excited with an

impulse. For the filtering purposes, a Butterworth filter of order 16 is used, and the cut-off frequency is chosen to be 5GHz. To validate that the Butterworth filter is functioning, the fast Fourier transform is applied to the impulse response of the transmission line and the filtered response of the transmission line.

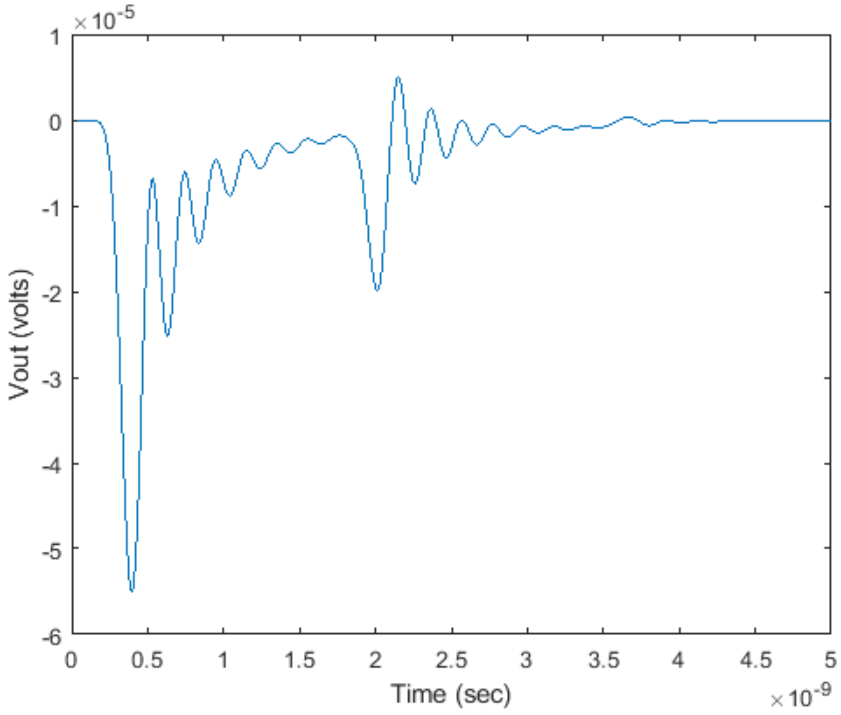

Figure 5.3: The response of the circuit, after applying the butterworth filter.

Figure 5.4, shows the comparison of those plots, which clearly shows that the magnitude of the filtered response is very negligible (almost zero), after the cut-off frequency. A higher order Butterworth filter is chosen because, it helps to get sharp transition into stop band which helps in eliminating the unwanted frequencies in the signal. Since higher frequencies in the signal will result in a higher order in reduced models, the sharp transition is required from the Butterworth filter. It is also important, not to the make the order of the butterworth filter too high which otherwise might lead to stability issues of the filter.

Once the filtered response is obtained, it is sent to order selection algorithm. Figure 5.5, shows the False Nearest Neighbors (%)(FNN) plot w.r.t the order of the reduced macro

model. We can clearly see that the FNN (%) is dropping to zero at the order 64.

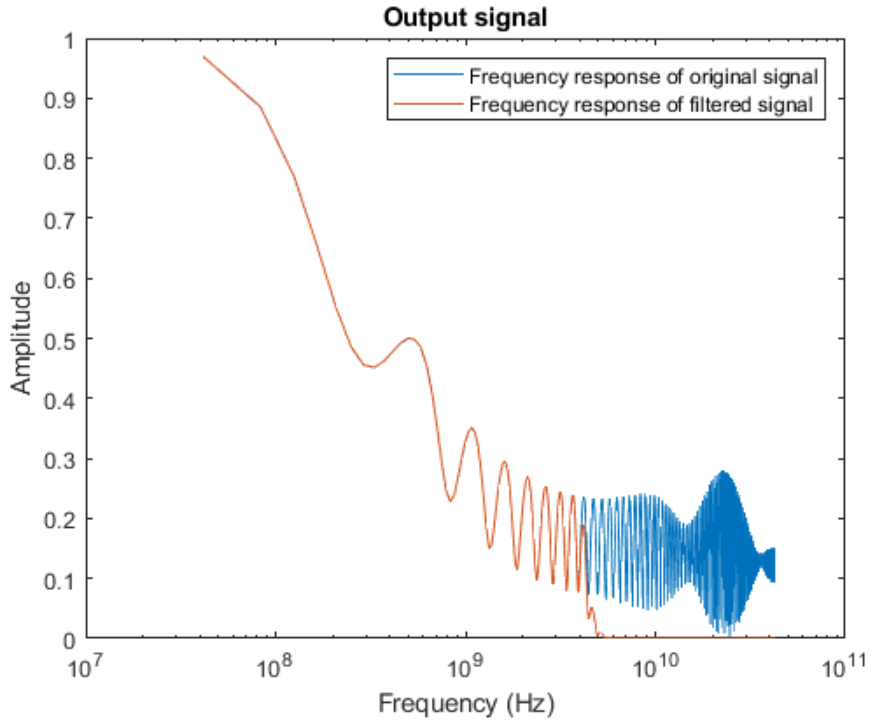

Figure 5.4: Comparison of frequency response of Original signal and filtered signal through FFT

As explained in the algorithm, the order selection algorithm terminates after a repeated **zero (%)** in FNN. By, using the order obtained from the order selection algorithm, the reduced macro model is formed by using an orthogonal matrix Q $\in$ $\mathbb{R}^{n\times m}$, where n is the number of states in the original model and m is the order of the reduced model. The frequency response of the original model and reduced model is compared to check if the order obtained shows the same behavior for both the models (original and reduced), till the desired frequency and Figure 5.6, shows the frequency response of the original and reduced model to validate the argument.

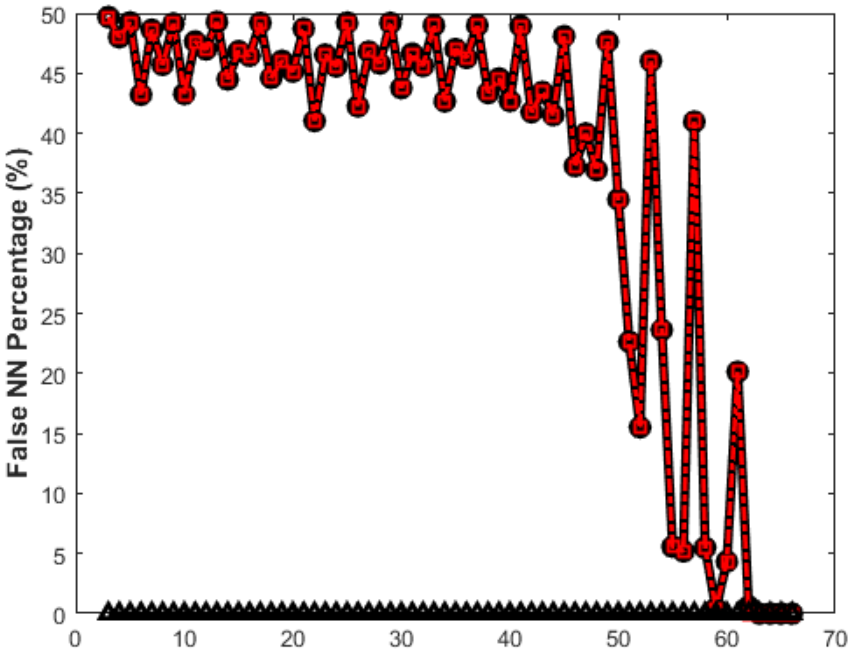

Figure 5.5: The percentage of the false nearest neighbors on the projected trajectory.

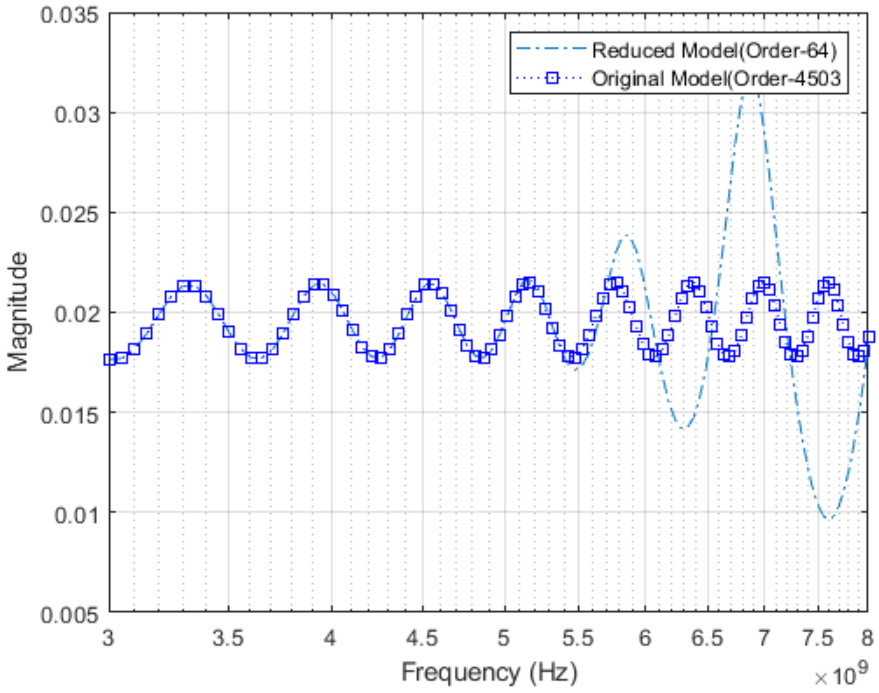

Figure 5.6: Frequency response of the Original model and the Reduced model (Order 64)

5.2 Example - 2

In this example, the transmission line of the form shown in Figure 5.7, is considered. The per-unit-length parameters are $C = 7.5372pF/cm$, $L = 2.893nH/cm$ *and* $R = 41.15\Omega/cm$. It is discretized into 200 lumped RLGC π-sections [40] resulting in a circuit of total order of 2412 (including the terminations). This is a 8-port network with 4-input ports and 4-output ports and the input of the transmission line is set to impulse.

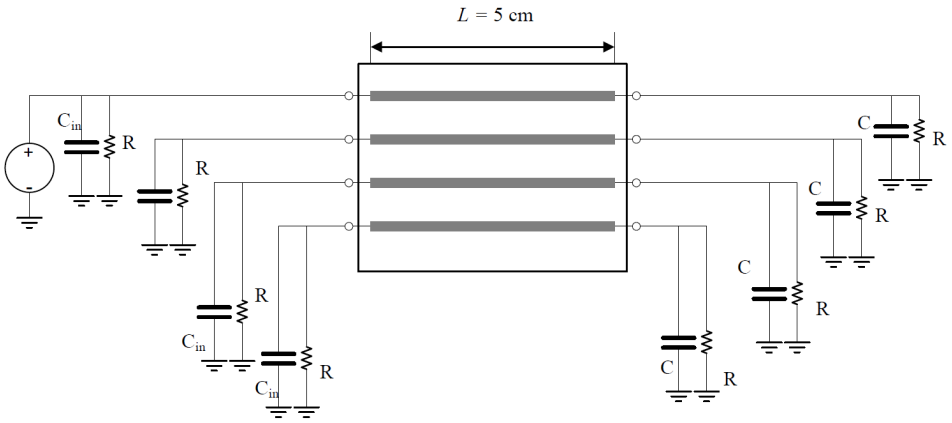

Figure 5.7: 8-port TL network with 200 segments

Applying the proposed method, Figure 5.8, shows the impulse response of the transmission line at the 4^{th} output port obtained using the NILT approach. Figure 5.9, shows the filtered response of the response shown in Figure 5.8 using the idea of Butterworth filter as illustrated in Section 4.4.

Next, the algorithm of order selection described in Chapter 3 has been applied to three different time-domain responses with the usual objective of reporting the suitable order to use in the Krylov subspace-based MOR algorithms. The three different time-domain points are listed as follows.

1. The unfiltered response obtained directly by applying the NILT approach to the original network (size 2412). This is the response shown in Figure 5.8.

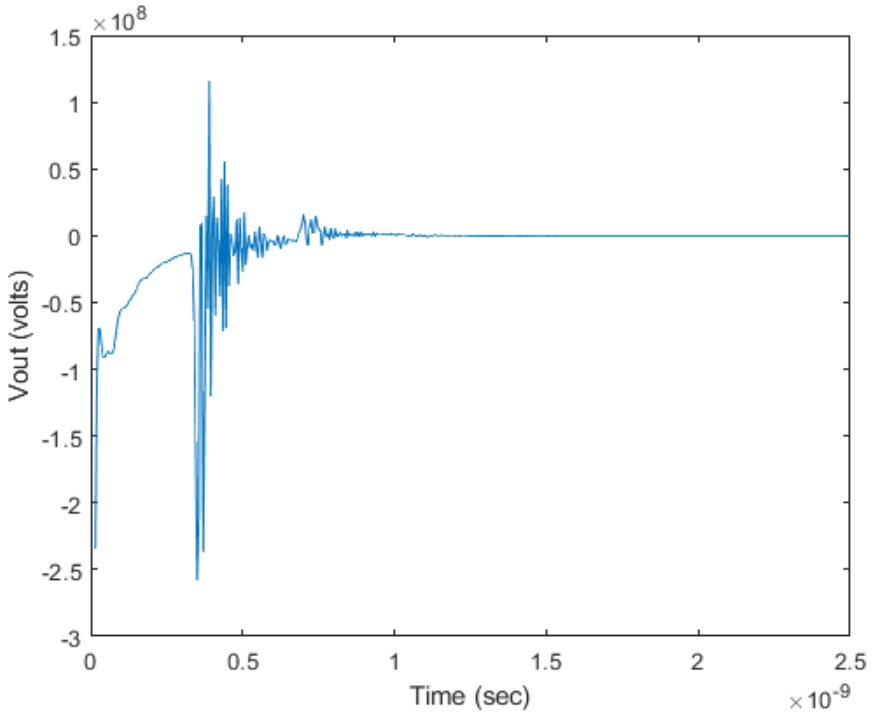

Figure 5.8: Impulse response of the network obtained using NILT without applying the filtering idea.

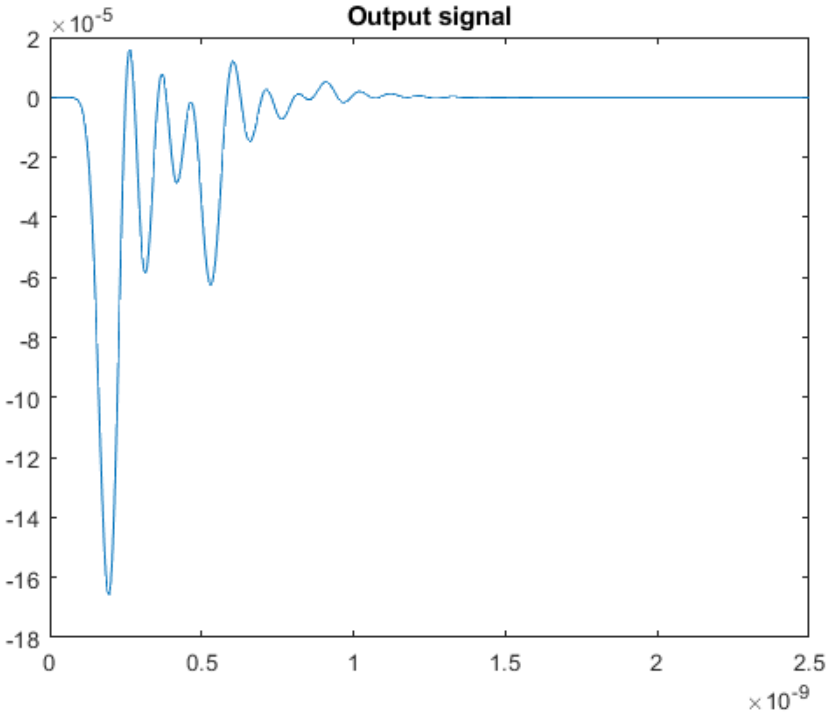

Figure 5.9: The response of the circuit, obtained from the NILT approach, after applying the butterworth filter.

2. The filtered response of NILT, which is obtained by taking the above response and applying the Butterworth filtering technique using the Matlab signal processing toolbox. This is the response shown in Figure 5.9. The cutoff frequency for the Butterworth filter was set at 10 GHz, and the order of the filter was set at order 16.

3. The response obtained by stimulating the original network with a Gaussian pulse described in Equation (3.16) with $\tau = 10$ps.

Figures 5.10, 5.11 and 5.12 show the count of the FNN on the projected trajectory obtained from applying the order selection algorithm to the above three time-domain responses, respectively, while the dimension of the model is incremented from m to $m+1$.

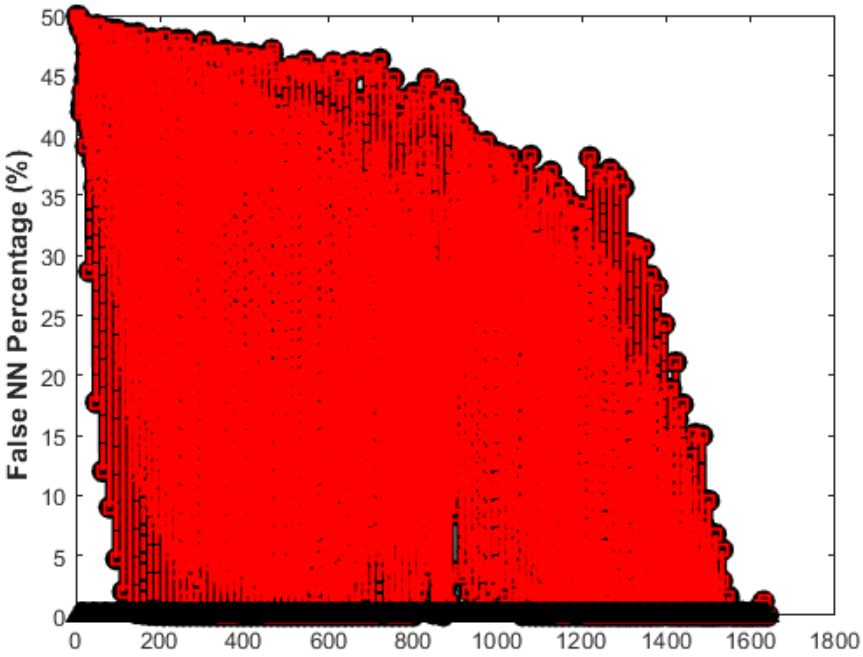

Figure 5.10: FNN of the impulse response obtained through NILT.

In order to provide a better picture of the termination condition of the order selection algorithm, i.e. the order of the reduced system where the number of FNN drops to zero, Figures 5.13 through 5.15 show a zoomed-in view of Figures 5.10 through 5.12 close to the termination of the algorithm.

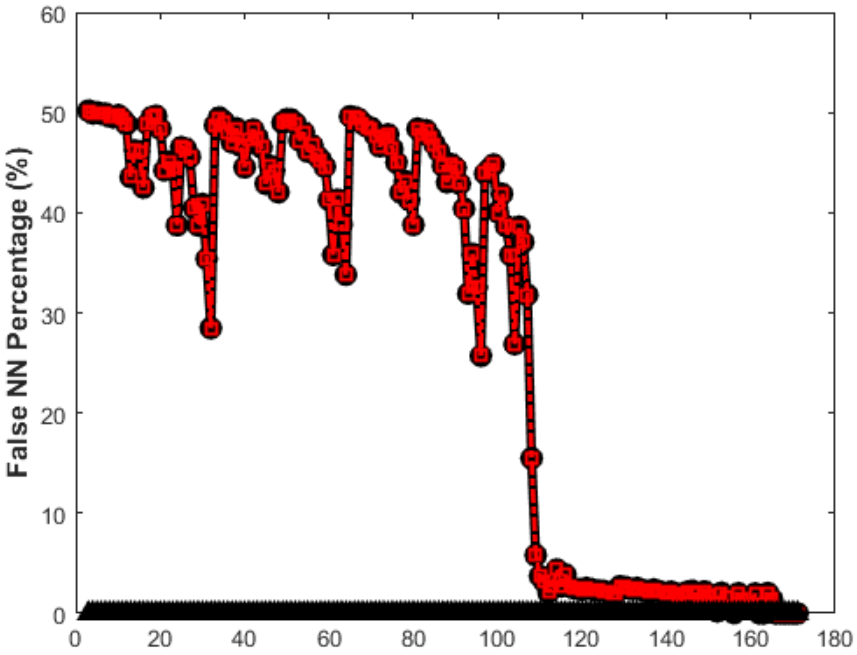

Figure 5.11: FNN of the impulse response (obtained through NILT) after applying Butterworth filter.

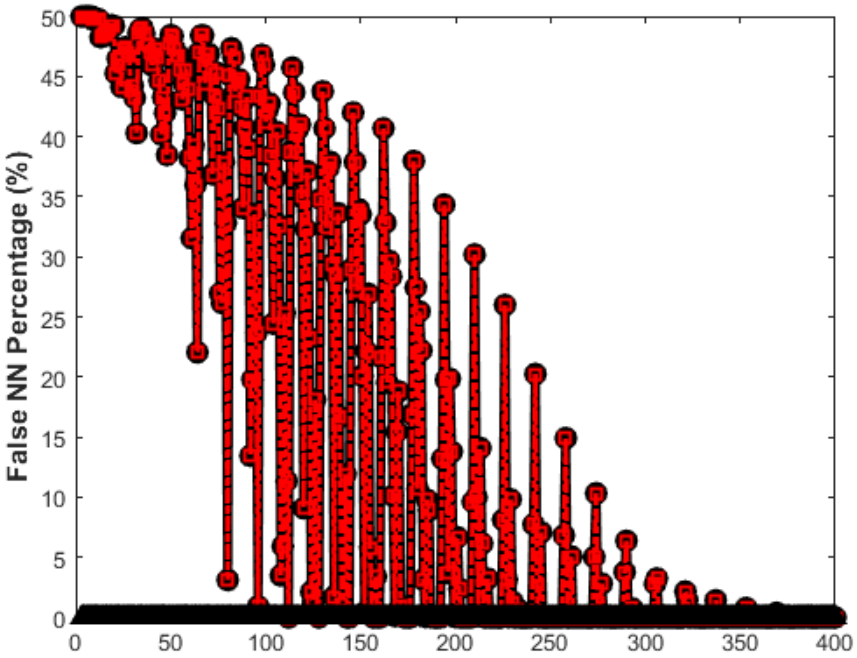

Figure 5.12: FNN based on using the time-domain data obtained from the network response to a Gaussian pulse.

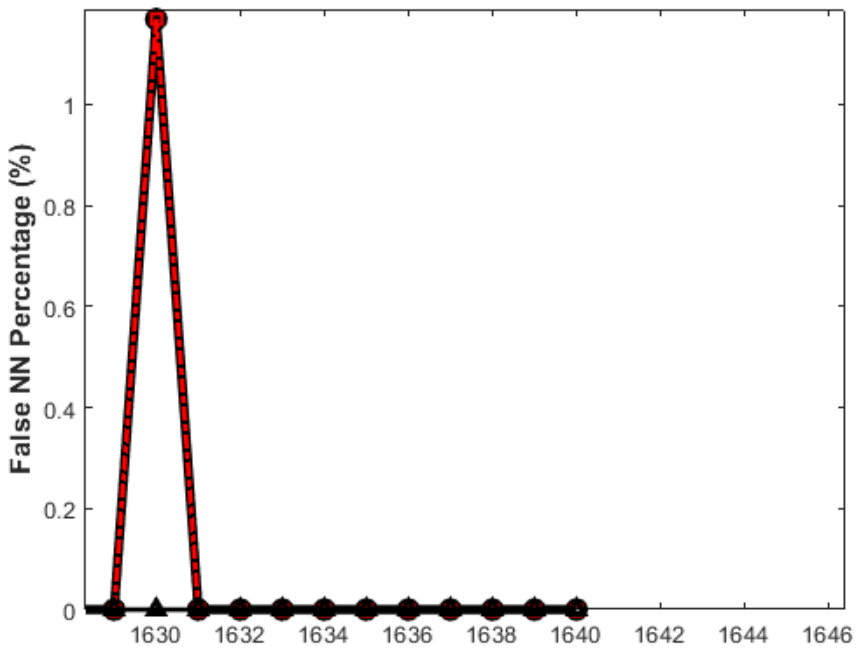

Figure 5.13: A zoomed in view of the graph in Figure 5.10 revealing the optimum order

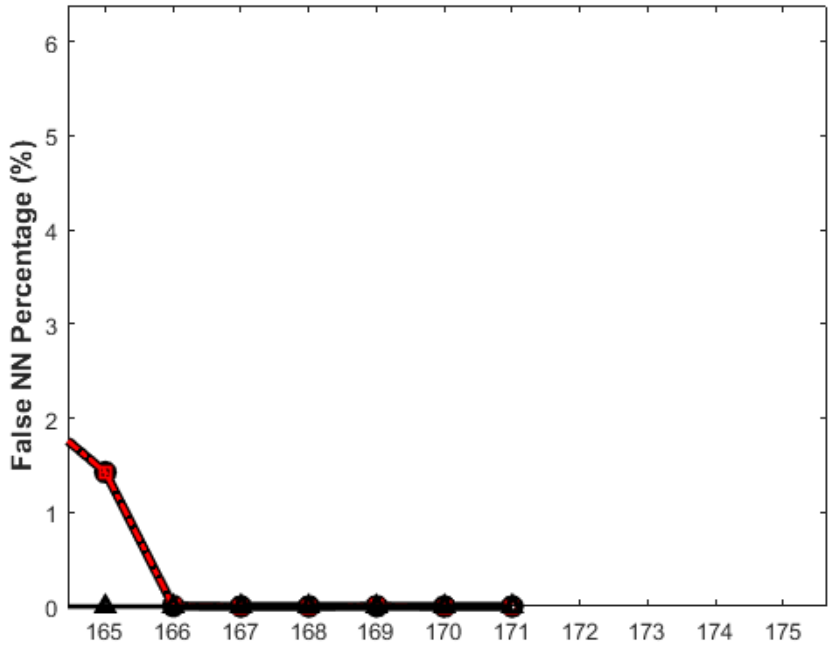

Figure 5.14: A zoomed in view of the graph in Figure 5.11 revealing the optimum order.

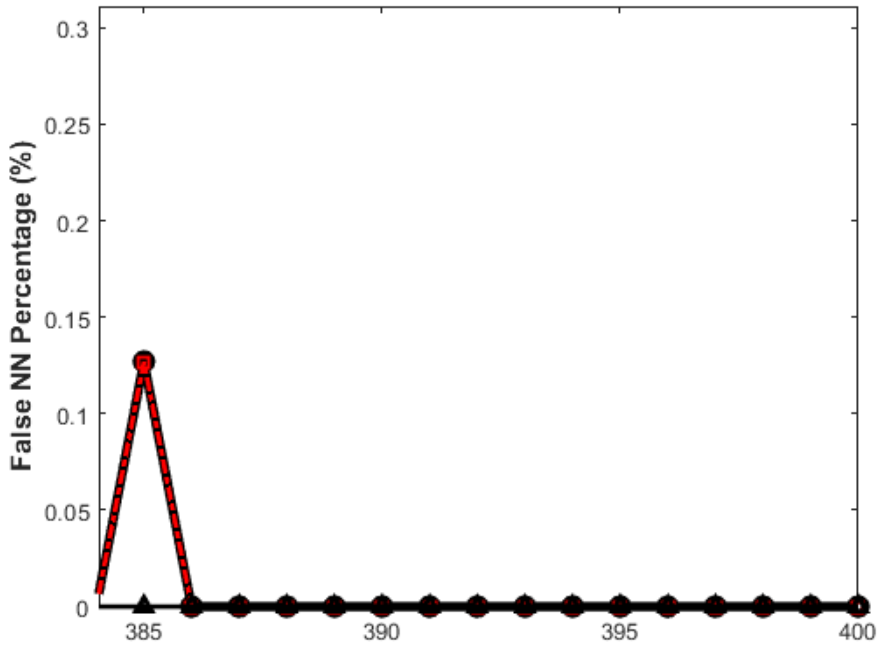

Figure 5.15: A zoomed in view of the graph in Figure 5.12 revealing the optimum order.

Table 5.1: The order selected based on the time-domain data generated from three different approaches.

Approach	NILT (no filtering)	(NILT with filtering)	TR with Gaussian input
Size	1631	167	386

The results obtained from those figures are summarized in Table 5.1, which shows the order selected for the size of the reduced system based on the three sets of time-domain data generated using the above-mentioned approaches. It clearly indicates that the proposed approach using NILT provides the smallest (and most optimal) size for the reduced system.

To shed more light on this result, the frequency content for each of the time-domain data generated through the above approaches was computed using Fast Fourier Transform (FFT). The results of this computations is shown in Figure 5.16. Those results clearly demonstrate that the one based on NILT with the filtering provides a perfect match of the frequency-domain response all the way up to the cutoff frequency of the Butterworth

filter, whereas the one based on the Gaussian response signficantly deviates from impulse response, and contains some high frequency components above the cutoff frequency.

To further validate the success of the proposed approach, in the sense that the frequency response of a reduced order system whose order is selected according to the results obtained from the second method (NILT+filtering) matches the frequency of the original system well up to the desired cutoff frequency, a frequency sweep for the two systems (original - size 2412 and reduced - size 167) was performed and the results are presented in Figure 5.17. It is obvious from this figure that a reduced system of size 167 is indeed sufficient to capture the frequency-domain behavior of the original system up to 10 GHz as desired.

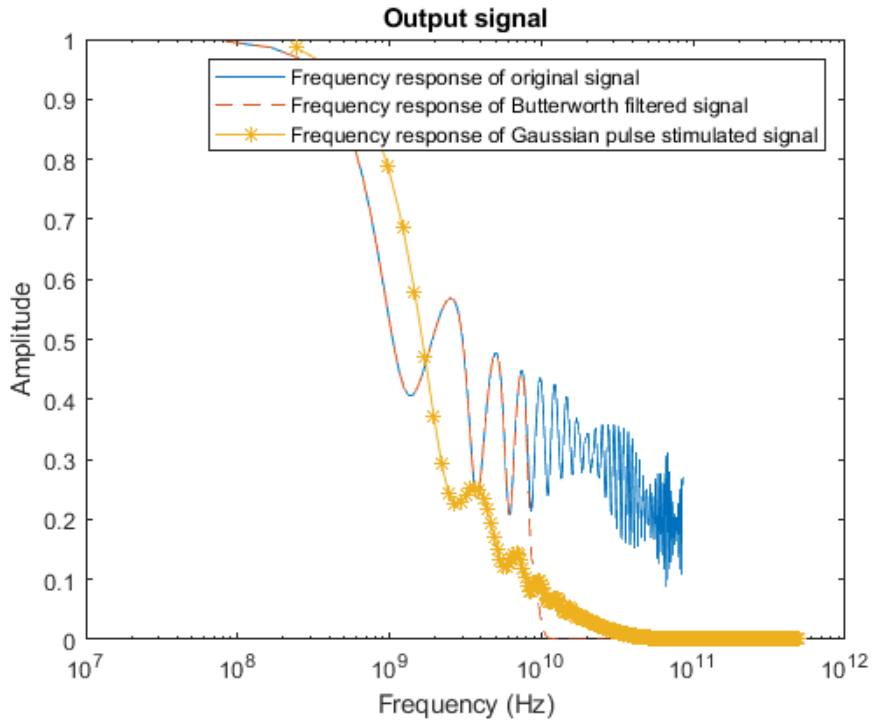

Figure 5.16: Comparison of frequency content of the original (NILT-based without filtering), butterworth-filtered impulse response and the Gaussian-based response, all obtained through applying FFT on the time-domain data.

To further show that the proposed approach (NILT+filtering) is indeed leading the order selection algorithm to choose a small order (compared to the order chosen based on the Gaussian-driven response), Figure 5.18 shows the results of performing a frequency

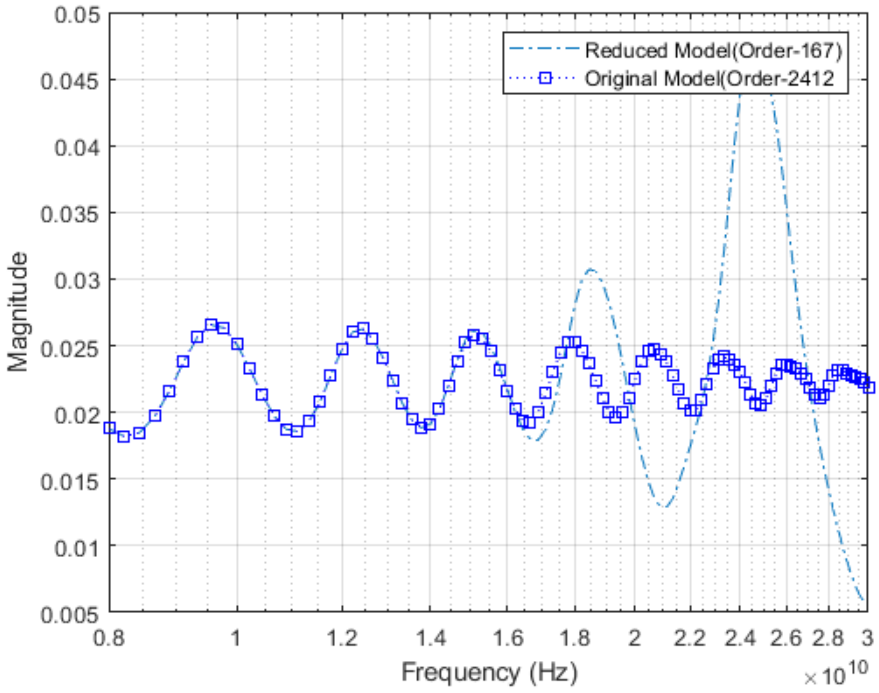

Figure 5.17: Frequency response of the Original model and the Reduced model (Order 167)

sweep of three different systems,

- Original system with order 2412,
- Reduced system with order 167 (chosen using the proposed approach - NILT+filtering),
- Reduced system with order 386 (chosen using the Gaussian-driven response).

The results in this figure clearly show that, while the one based on the proposed approach matched the original system perfectly well up to (and only up to) the desired frequency (10 GHz), the one selected from the Gaussian-driven matched (unnecessarily) more than the required range, thereby indicating an inflation in the reduced system.

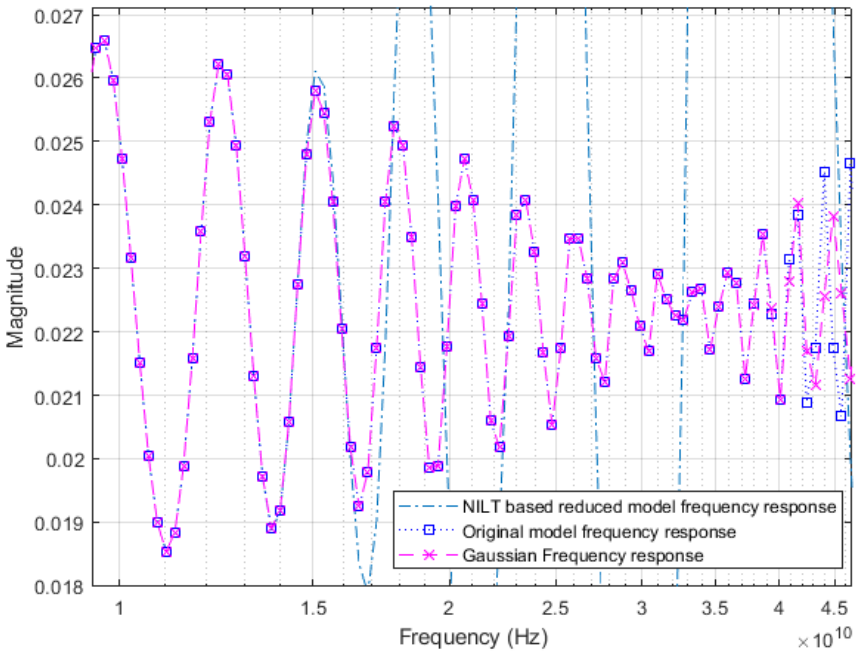

Figure 5.18: Comparison of Frequency responses of reduced models with original model

5.3 Example - 3

In this example, the transmission line of the form shown in Figure 5.7, is considered, but this time the length of the line was set to $L = 10$cm. As shown in the figure, this network can be seen as an 8-port network with 4-input ports and 4-output ports.

Here, the transmission line is descritized using 2000 lumped RLGC π-sections, as illustrtated in [40], resulting in a network with size 24012. The resulting network is then considered for reduction using a Krylov-subspace technique, where the order of the reduced system is selected according to the results obtained from the proposed approach.

The experiments run on this example are similar to the experiments run on the previous example, and the results obtained suggest the same conclusion as will be illustrated next.

First by applying the proposed method an approximation of the impulse response was obtained using the NILT approach. The time-domain impulse response at the 4^{th} output port thus obtained is shown in Figure 5.19.

Next, the Butterworth filtering approach presented in Section 4.4 was then applied on that response to truncate out frequency content (in the waveform in Figure 5.19) above a cutoff frequency which was taken at 50GHz, with the order of the Butterworth filter set at order 16. Figure 5.20 shows the time-domain waveform of the filtered response.

Following that, the full (all circuit variables) filtered time-domain waveforms (with a sample already given in Figure 5.20) is fed to the order selection algorithm described in Chapter 3 which monitors the count of FNN as the size of the reduced system is incremented from m to $m + 1$. Figure 5.23 shows the % of the FNN indicating that it drops to zero around $m = 651$ and suggesting that a reduced system of size 651 is sufficient to capture the behavior of the original system (size 24012) up to 10GHz. Figure 5.22 shows a magnified view near the termination of the order selection algorithm to clearly indicate the order selected based on those data.

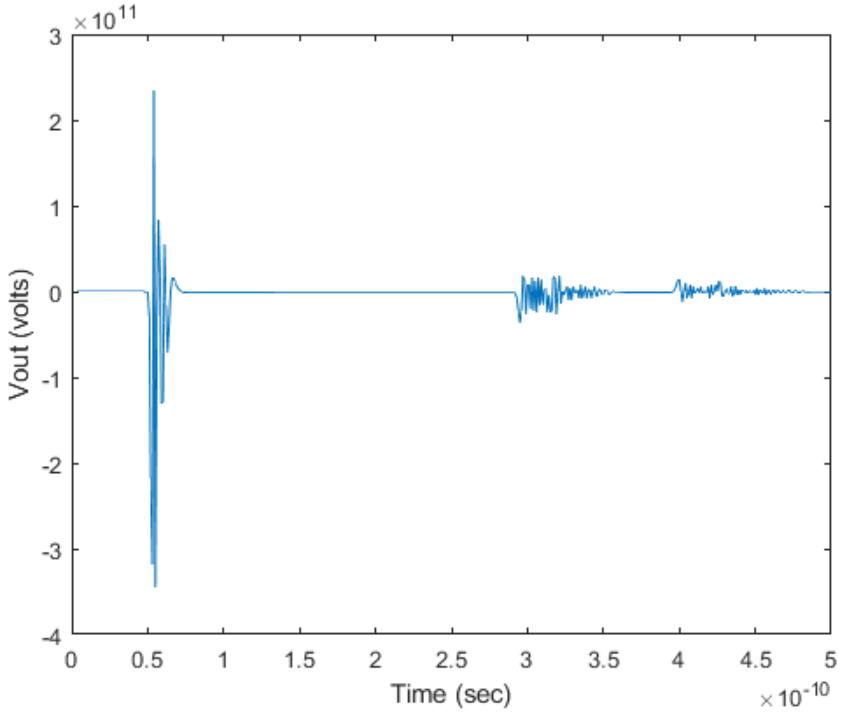

Figure 5.19: Impulse response of the network

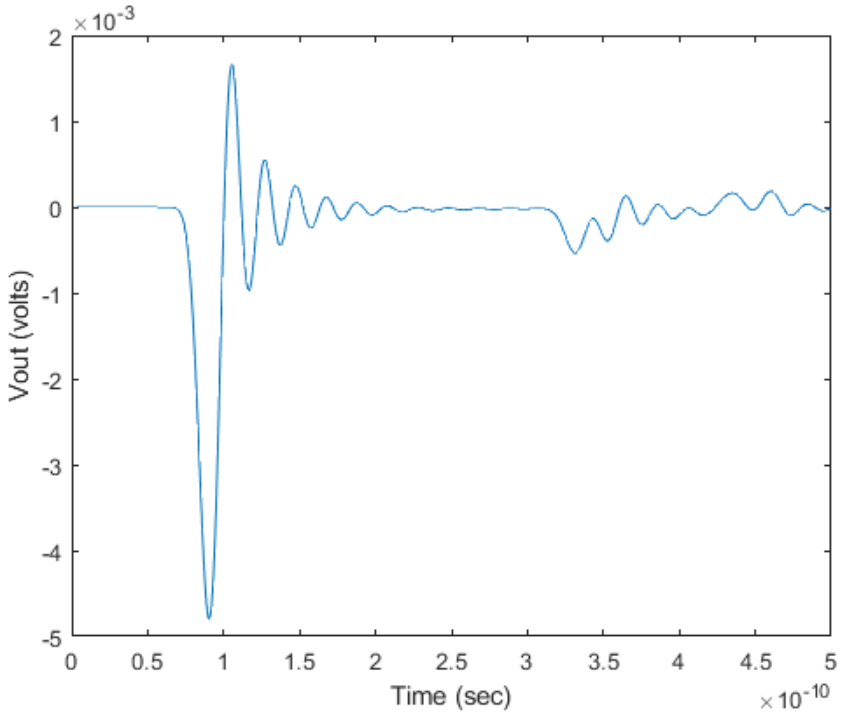

Figure 5.20: The response of the circuit, after applying the butterworth filter.

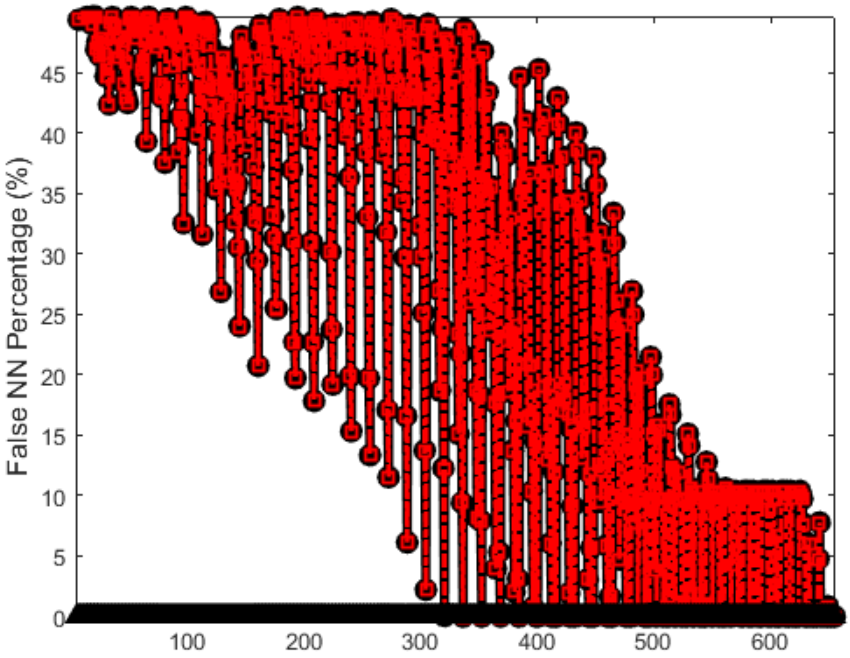

Figure 5.21: FNN of the impulse response after applying Butterworth filter.

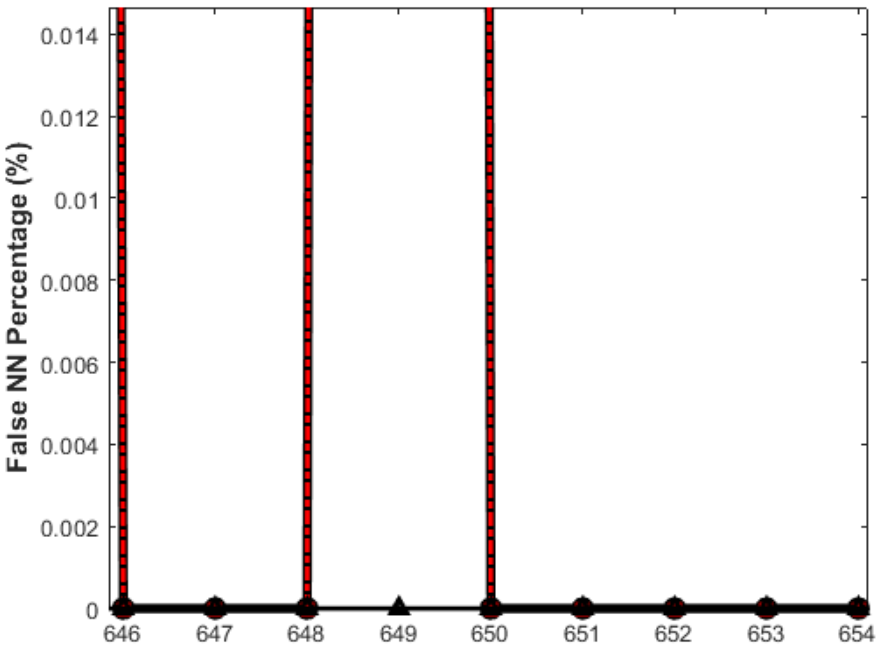

Figure 5.22: A zoomed in view of the graph in Figure 5.21 revealing the optimum order.

Table 5.2: The order selected based on the time-domain data generated from three different approaches for Example 3.

Approach	NILT (no filtering)	(NILT with filtering)	TR with Gaussian input
Size	8000	651	2995

In order to compare the performance of the order selection algorithm operating on the proposed approach (NILT + filtering) as opposed to it working directly on the unfiltered NILT or the conventional Gaussian-based data, Figures 5.23 through 5.26 present the FNN produced on the latter two approaches, along with a zoomed-in view.

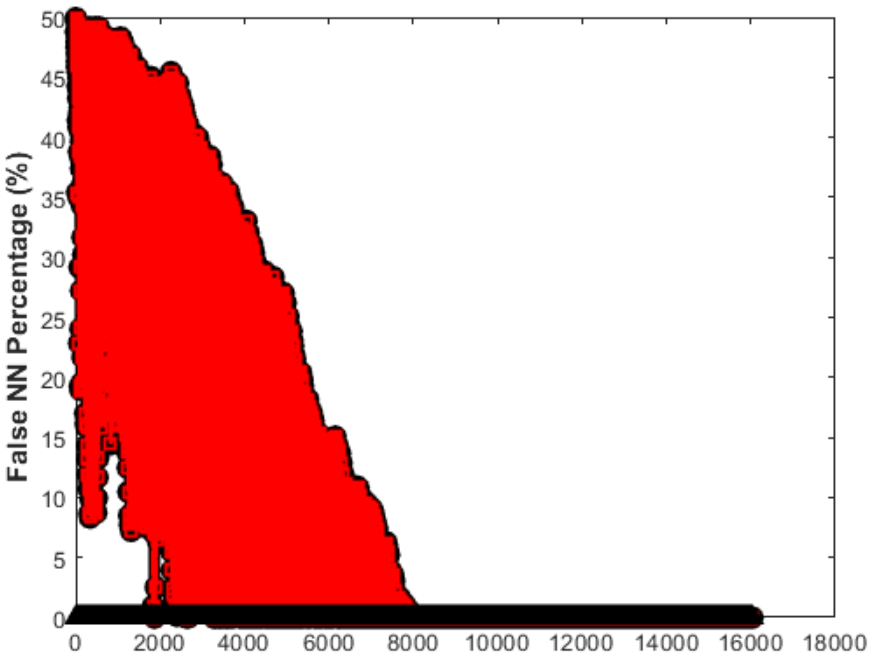

Figure 5.23: FNN of the impulse response approximated using the NILT approach without applying the filtering.

The sizes of the reduced-order systems, as indicated using the order selection algorithm when invoked on each of the above three time-domain results, is summarized in Table 5.2. Again those results clearly demonstrate that the proposed approach (NILT+filtering) conduces the order selection algorithm to present a more compact size.

To show the validity of this results, the Fast Fourier Transform is applied on the time-

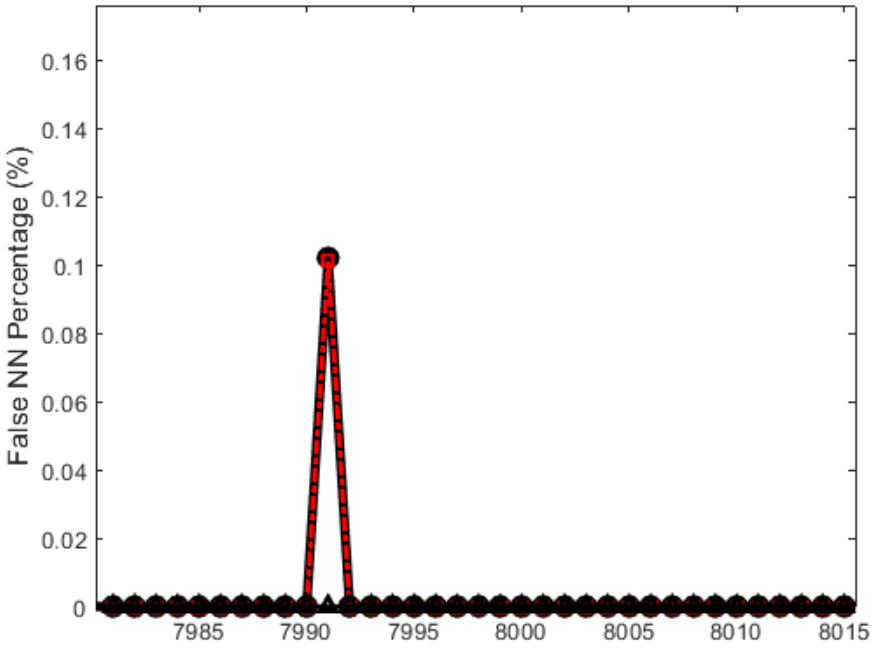

Figure 5.24: A zoomed in view of the graph in FIgure 5.23 revealing the optimum order

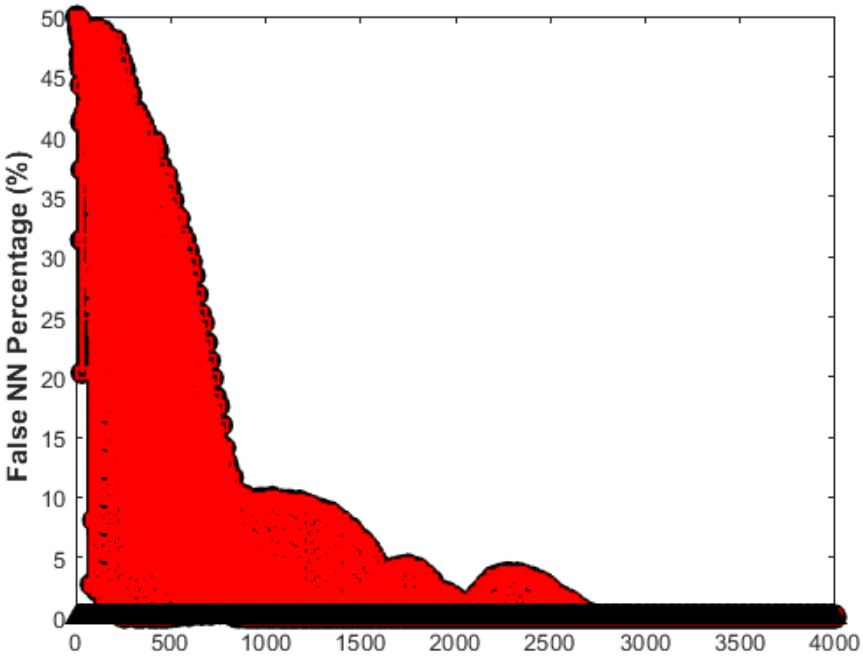

Figure 5.25: FNN obtained from applying the order selection algorithm on time-domain response of the network to a Gaussian pulse with $\tau = 3e - 12$.

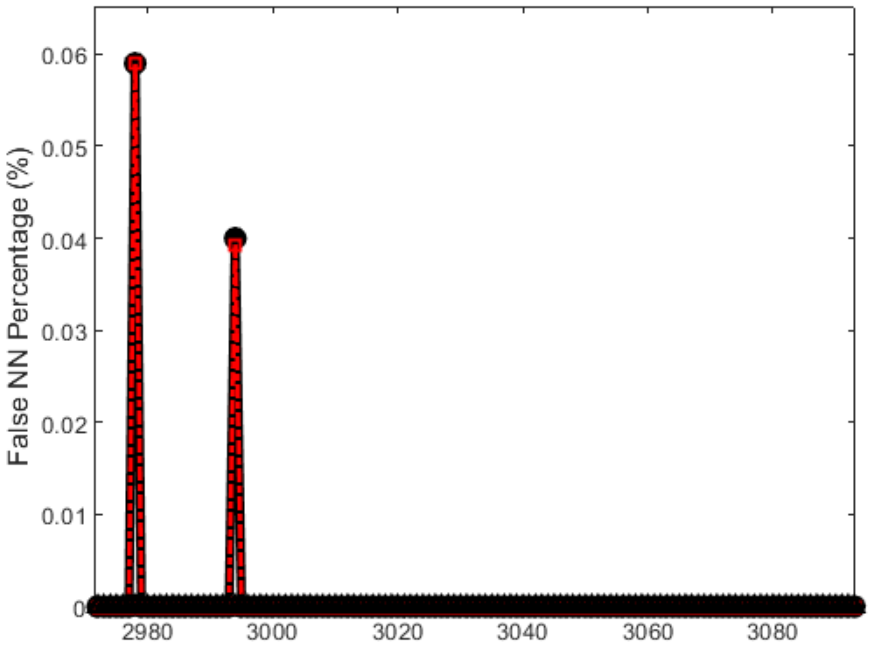

Figure 5.26: A zoomed in view of the graph in Figure 5.25 revealing the optimum order.

domain data of the Gaussian response, NILT+filtered response and impulse response from NILT. Figure 5.27 shows the comparison of the frequency contents obtained through FFT. Figure 5.28 clearly shows the higher frequency content of the Gaussian response.

To demonstrate that a reduced-order system of size 651 is indeed adequate in capturing the original system behaviour up to the cuttoff frequency of the filter (of 50 GHz), a frequency sweep for the original system (size 24012) and the reduced one is carried out and the comparison between the two system is depicted in Figure 5.29, confirming the adequacy of the order selected.

5.4 Discussion

The examples presented in this chapter demonstrate that the proposed approach of using NILT to approximate the impulse response followed by applying the Butterworth filtering

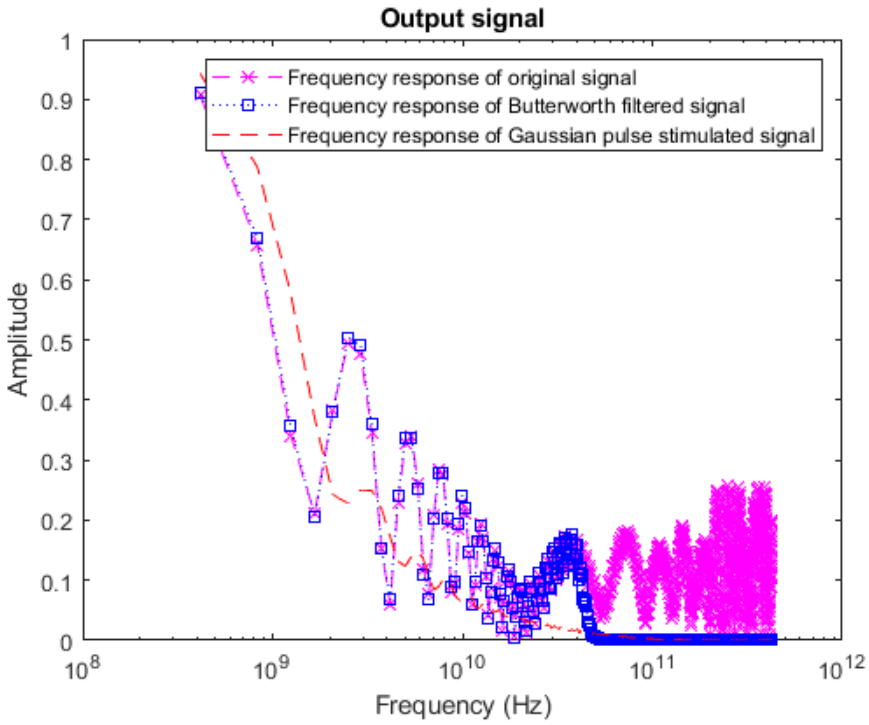

Figure 5.27: Comparison of frequency content of the original (NILT-based without filtering), Butterworth-filtered impulse response and the Gaussian-based response, all obtained through applying FFT on the time-domain data.

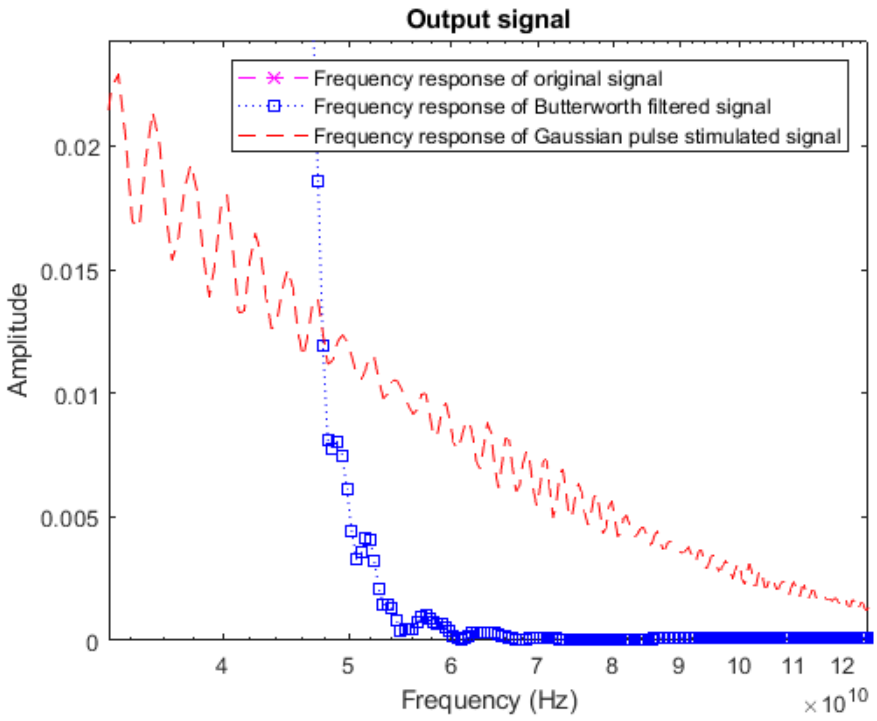

Figure 5.28: A zoomed in view of the graph in Figure 5.27 revealing the frequency contents of the time-domain data

provides an observable advantage. To summarize this advantage, one might say that the time-domain data thus obtained from that approach guides the order selection algorithm in converging on a concise reduced-order system. More specifically the order selected based on this approach is more concise than the order selected by a Gaussian-driven response.

The reasons for this advantage of using the NILT followed by the Butterworth may be summed up in the following remarks.

1. The time-domain data obtained from a Gaussian driven response contains higher frequency components above the desired maximum frequency. In contrast the Butterworth filtering more forcibly dampens the higher frequency components above the desired maximum frequency. This remark can be visualized by comparing the frequency-domain content of both the Gaussian pulse and that of the Butterworth filter of order 10, which is shown in Figure 5.30.

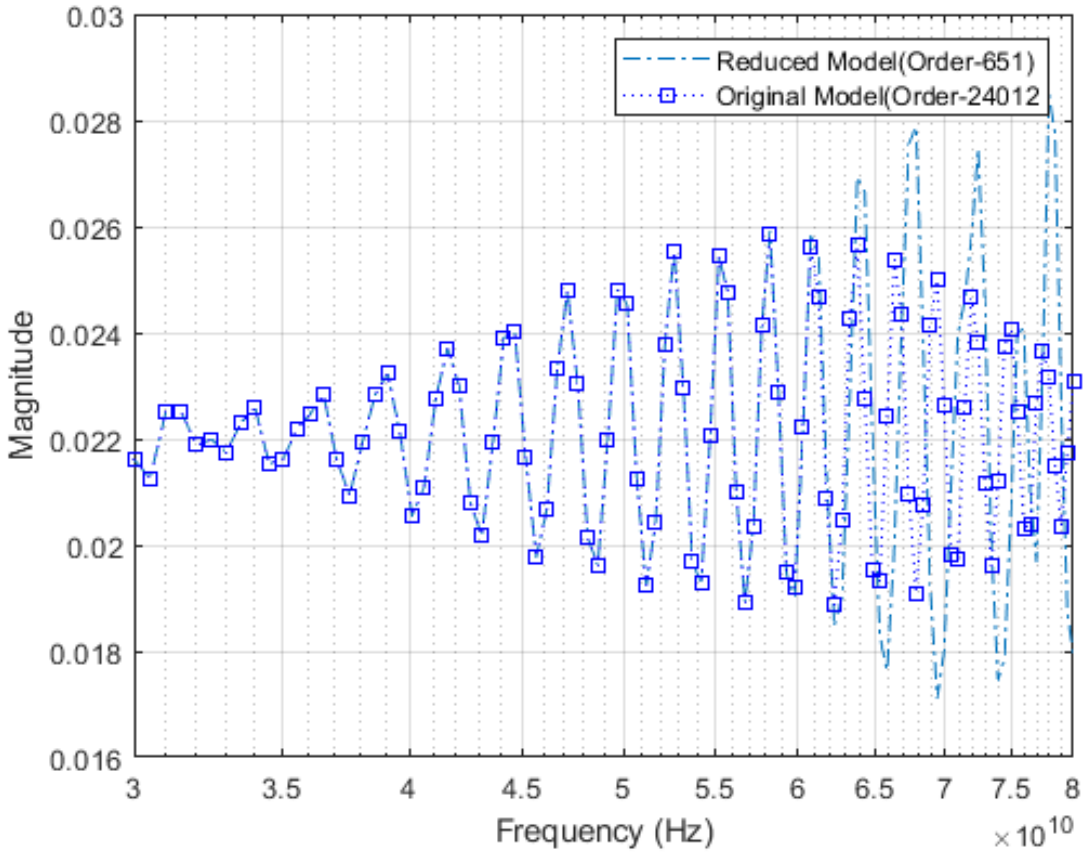

Figure 5.29: Frequency response of the Original model and the reduced model.

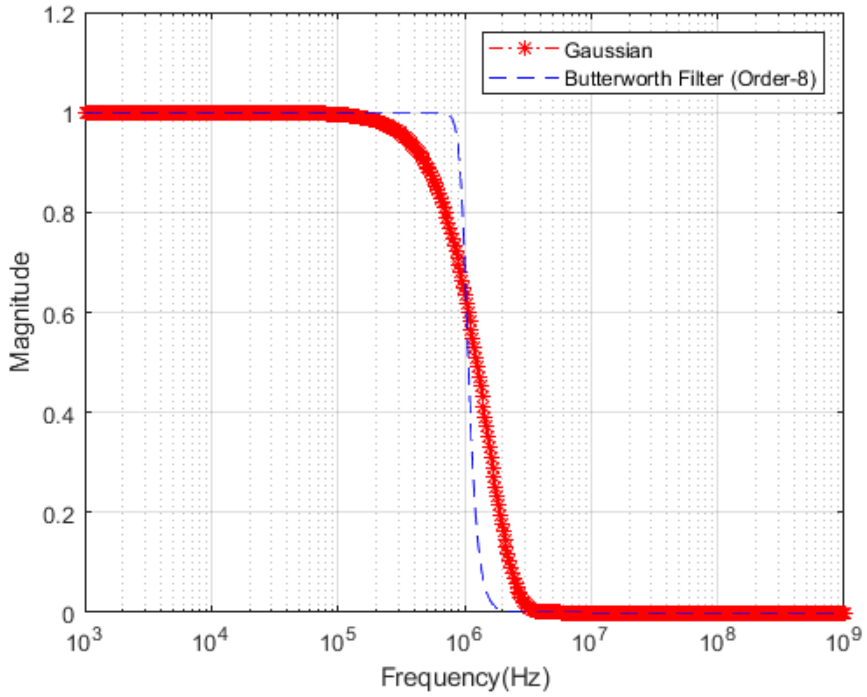

Figure 5.30: Comparison of filtering effect of Gaussian and Butterworth through frequency response

With higher frequency components present in the signal the order selection algorithm may be misled into producing a reduced model that capture higher frequency components that may not be needed.

2. It should also be noted that in generating the response to a Gaussian stimulus, the low-order method of the Trapezoidal Rule has been employed. Being of low-order, the response contains a certain amount of "numerical noise". Here, the notion of numerical noise refers to the numerical error incurred by the low-order, which acts as noise superimposed over the actual response. This numerical noise may be a misleading factor for the order selection algorithm, effectively steering the algorithm to increase the order to eliminate all the FNN. By contrast, the NILT approach is a high-order technique. This means that the error resulting from the approximation is much lower than the error resulting from the Trapezoidal Rule. This factor dampens the numerical noise strongly for NILT more than for the Trapezoidal Rule, again

leading the order selection algorithm to converge on a more compact reduced-order model.

Chapter 6

Conclusions and Future Work

This chapter contains a summary of the work presented in this thesis. In addition, the possible directions for the future work are discussed.

6.1 Conclusions

This thesis presented a natural extension to Model-Order Reduction techniques by providing and implementing an efficient method to obtain optimum order of the reduced model. The application of the NILT method paved way for obtaining the impulse response of the network which showed advantages over the Gaussian pulse as the input. The application of the Butterworth filter to the impulse response restricted the frequency content of the signal to the desired range, which led to get an efficient order for the reduced models.

The accuracy of the proposed method has been shown with a typical design problem of optimizing high speed interconnect parameters to meet strict design specifications. The high speed interconnects were discretized into a number of **RLGC** segments. The resulting large circuit was reduced in size using PRIMA algorithm although not limited to a specific order reduction algorithm. The resulting values from the optimization process accurately met the design specification for the original circuit.

6.2 Future Research

Unlike time marching methods which can only be computed by sequential/serial processing, NILT can be implemented by parallel processing. This can significantly increase the speed of the process.

APPENDICES

Appendix A

Special Complex Integration of a Rational Function

The problem discussed in this appendix is needed in Chapter 4 for the numerical inversion of the Laplace transform. It involves the solution of

$$\mathbf{I} \equiv \int_{c-j\infty}^{c+j\infty} \mathbf{V}(s)ds \tag{A.1}$$

with

$$\mathbf{V}(s) \;=\; \frac{\sum_{i=0}^{N} a_i s_i}{\sum_{i=0}^{M} b_i s_i} \tag{A.2}$$

We wish to establish the condition under which the integral can be found by means of residue calculus as follows:

$$\mathbf{I} \;=\; 2\pi j \times (\text{sum of the residues}). \tag{A.3}$$

To apply formula (A.3) we close the path by a semicircle in the left half plane and let the radius grow without bound. If the contribution to the integral along this infinite semicircle

is zero, then the whole integral will be given by the integration along the path indicated in (A.1) and the sum of residues will be the solution.

For the contribution I_1 along the semicircle introduce a new variable

$$s \;=\; R\exp\{j\phi\} \tag{A.4}$$

On the semicircle going from $+jR$ to $-jR$ in the left half plane R is constant, so that

$$ds \;=\; jR\exp\{j\phi\}\; d\phi \tag{A.5}$$

and

$$I_1 \;=\; \int_{+\pi/2}^{-\pi/2} V(R\exp\{j\phi\})jR\exp\{j\phi\}\; d\phi. \tag{A.6}$$

For very large R, the terms with the highest powers dominate and we can simplify:

$$\begin{aligned} I_1 \;&=\; \lim_{R\to\infty}\int_{+\pi/2}^{-\pi/2} \frac{a_N R^N \exp\{j\phi N\}}{b_M R_M \exp\{j\phi M\}} jR\exp\{j\phi\}\; d\phi \\ &=\; \lim_{R\to\infty}\frac{a_N}{b_M}\frac{j}{R^{M-N-1}}\int_{+\pi/2} -\pi/2\exp\{j\phi(N+1-M)\}\; d\phi. \end{aligned} \tag{A.7}$$

The last integral on the right is finite. In order to make the whole expression equal to zero, we must select M, N such that at least the first power of R remains in the denominator. This requires

$$M - N - 1 \geq 1 \tag{A.8}$$

or

$$M \geq N + 2 \tag{A.9}$$

and we can formulate a lemma:

The integral of a rational function $\mathbf{V}(s)$ along an infinite semicircle is zero

whenever $\mathbf{V}(s)$ has at least two more poles than zeros.

When integrating a rational function with $M \geq N+2$ along the straight line parallel to the imaginary axis, the integral is equal to $2\pi j \times$(sum of residues at poles to the left of the line) if the path is closed counterclockwise in the left half plane. If we close the integration path clockwise in the right half plane, the sum of the residues at poles appearing to the right of the line is taken but is multiplied by $-2\pi j$.

References

[1] Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499, 1994.

[2] T Aittokallio, M Gyllenberg, Jarmo Hietarinta, T Kuusela, and T Multamäki. Improving the false nearest neighbors method with graphical analysis. *Physical Review E*, 60(1):416, 1999.

[3] BD Anderson. o., and vongpanitlerd, s.:'network analysis and synthesis', 1973.

[4] Athanasios C Antoulas, Danny C Sorensen, and Serkan Gugercin. A survey of model reduction methods for large-scale systems. *Contemporary mathematics*, 280:193–220, 2001.

[5] Walter Edwin Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of applied mathematics*, 9(1):17–29, 1951.

[6] Zhaojun Bai. Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. *Applied numerical mathematics*, 43(1-2):9–44, 2002.

[7] E. Beltrami and E.J. Beltrami. *Mathematics for Dynamic Modeling*. Academic Press, 1998.

[8] Gal Berkooz, Philip Holmes, and John L Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual review of fluid mechanics*, 25(1):539–575, 1993.

[9] Peter Caines. Review of 'linear systems' (t. kailath, 1980). 27:385– 386, 06 1981.

[10] Mustafa Celik, Lawrence Pileggi, Larry Pileggi, and Altan Odabasioglu. *IC interconnect analysis*. Springer Science & Business Media, 2002.

[11] Chi-Tsong Chen. *Linear System Theory and Design*. Oxford University Press, Inc., New York, NY, USA, 3rd edition, 1998.

[12] Eli Chiprout and Michel S Nakhla. Asymptotic waveform evaluation. In *Asymptotic Waveform Evaluation*, pages 15–39. Springer, 1994.

[13] Dale F Enns. Model reduction with balanced realizations: An error bound and a frequency weighted generalization. In *Decision and Control, 1984. The 23rd IEEE Conference on*, volume 23, pages 127–132. IEEE, 1984.

[14] Peter Feldmann and Roland W Freund. Efficient linear circuit analysis by padé approximation via the lanczos process. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 14(5):639–649, 1995.

[15] Roland W. Freund. *Reduced-Order Modeling Techniques Based on Krylov Subspaces and Their Use in Circuit Simulation*, pages 435–498. Birkhäuser Boston, Boston, MA, 1999.

[16] Roland W Freund. Krylov-subspace methods for reduced-order modeling in circuit simulation. *Journal of Computational and Applied Mathematics*, 123(1-2):395–421, 2000.

[17] Roland W Freund. Sprim: structure-preserving reduced-order interconnect macromodeling. In *Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design*, pages 80–87. IEEE Computer Society, 2004.

[18] John R Gilbert and E Ng. Predicting structure in nonsymmetric sparse matrix factorizations, graph theory and sparse matrix computation (edited by alan george and john r. gilbert and joseph wh liu), 1993.

[19] Keith Glover. All optimal hankel-norm approximations of linear multivariable systems and their l,∞-error bounds. *International journal of control*, 39(6):1115–1193, 1984.

[20] Eric James Grimme. *Krylov projection methods for model reduction*. PhD thesis, Citeseer, 1997.

[21] Läuter H. Loéve, m.: Probability theory i and ii, 4th ed. (graduate texts in mathematics 45 u. 46) springer-verlag, berlin-heidelberg-new york 1977/78, 441, 429 pp. *Biometrical Journal*, 24(5):462–462.

[22] E. Hairer, S. P. Nøsett, and G. Wanner. *Solving Ordinary Differential Equations I Nonstiff Problems*. Springer, Berlin Heidelberg, 3 edition, 2008.

[23] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems*, volume II. Springer, 1996.

[24] Payam Heydari and Massoud Pedram. Model-order reduction using variational balanced truncation with spectral shaping. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 53(4):879–891, 2006.

[25] Chung-Wen Ho, A Ruehli, and Pierce Brennan. The modified nodal approach to network analysis. *IEEE Transactions on circuits and systems*, 22(6):504–509, 1975.

[26] Harold Hotelling. Simplified calculation of principal components. *Psychometrika*, 1(1):27–35, 1936.

[27] H. Kantz and T. Schreiber. *Nonlinear Time Series Analysis*. Cambridge nonlinear science series. Cambridge University Press, 2004.

[28] Matthew B Kennel, Reggie Brown, and Henry DI Abarbanel. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical review A*, 45(6):3403, 1992.

[29] Ernest S Kuh and Ronald A Rohrer. The state-variable approach to network analysis. *Proceedings of the IEEE*, 53(7):672–686, 1965.

[30] Cornelius Lanczos. *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*. United States Governm. Press Office Los Angeles, CA, 1950.

[31] Alanj Laub, MICHAELT Heath, C Paige, and R Ward. Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms. *IEEE Transactions on Automatic Control*, 32(2):115–122, 1987.

[32] Jing-Rebecca Li and Jacob White. Efficient model reduction of interconnect via approximate system gramians. In *Proceedings of the 1999 IEEE/ACM international conference on Computer-aided design*, pages 380–384. IEEE Press, 1999.

[33] Wolfgang Liebert, K Pawelzik, and HG Schuster. Optimal embeddings of chaotic attractors from topological considerations. *EPL (Europhysics Letters)*, 14(6):521, 1991.

[34] Ching-An Lin, Tai-Yih Chiu, et al. Model-reduction via frequency weighted balanced realization. *Control-Theory and Advanced Technology*, 8(2):341–351, 1992.

[35] Bruce Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE transactions on automatic control*, 26(1):17–32, 1981.

[36] F.N. Najm. *Circuit Simulation*. Wiley - IEEE. Wiley, 2010.

[37] Behzad Nouri, Michel S Nakhla, and Ramachandra Achar. A novel algorithm for optimum order estimation of reduced order macromodels. In *Signal Propagation on Interconnects (SPI), 2011 15th IEEE Workshop on*, pages 33–36. IEEE, 2011.

[38] Behzad Nouri, Michel S Nakhla, and Ramachandra Achar. A novel algorithm for optimum order estimation of nonlinear reduced macromodels. In *Electrical Performance of Electronic Packaging and Systems (EPEPS), 2013 IEEE 22nd Conference on*, pages 137–140. IEEE, 2013.

[39] Altan Odabasioglu, Mustafa Celik, and Lawrence T Pileggi. Prima: Passive reduced-order interconnect macromodeling algorithm. In *Proceedings of the 1997 IEEE/ACM international conference on Computer-aided design*, pages 58–65. IEEE Computer Society, 1997.

[40] C. Paul. *Analysis of Multiconductor Transmission Lines*. Wiley, New York, 1994.

[41] Joel R Phillips, Luca Daniel, and Luís Miguel Silveira. Guaranteed passive balancing transformations for model order reduction. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22(8):1027–1041, 2003.

[42] Joel R Phillips and Luís Miguel Silveira. Poor man's tbr: A simple model reduction scheme. *IEEE transactions on computer-aided design of integrated circuits and systems*, 24(1):43–55, 2005.

[43] L. T. Pillage, R. A. Rohrer, and C. Visweswariah. Electronic circuit and system simulation methods [book reviews]. *IEEE Circuits and Devices Magazine*, 14(6):45–45, Nov 1998.

[44] Lawrence T Pillage and Ronald A Rohrer. Asymptotic waveform evaluation for timing analysis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 9(4):352–366, 1990.

[45] Lawrence T Pillage, Ronald A Rohrer, and Chandramouli Visweswariah. *Electronic circuit and system simulation methods*. McGraw-Hill New York, 1995.

[46] Muruhan Rathinam and Linda R Petzold. A new look at proper orthogonal decomposition. *SIAM Journal on Numerical Analysis*, 41(5):1893–1925, 2003.

[47] Carl Rhodes and Manfred Morari. The false nearest neighbors algorithm: An overview. *Computers & Chemical Engineering*, 21:S1149–S1154, 1997.

[48] Carl Rhodes and Manfred Morari. False-nearest-neighbors algorithm and noise-corrupted time series. *Physical Review E*, 55(5):6162, 1997.

[49] Ronald A Rohrer. *Circuit theory: an introduction to the state variable approach*. McGraw-Hill, 1969.

[50] Yousef Saad. *Iterative methods for sparse linear systems*, volume 82. siam, 2003.

[51] Michael G Safonov and RY Chiang. A schur method for balanced-truncation model reduction. *IEEE Transactions on Automatic Control*, 34(7):729–733, 1989.

[52] Amol Jagannath Sasane. *Hankel norm approximation for infinite-dimensional systems*. PhD thesis, 2001. Relation: https://www.rug.nl/ date_submitted:2001 Rights: University of Groningen.

[53] Thomas Schreiber. Efficient neighbor searching in nonlinear time series analysis. *International Journal of Bifurcation and Chaos*, 5(02):349–358, 1995.

[54] L Miguel Silveira, Mattan Kamon, and Jacob White. Efficient reduced-order modeling of frequency-dependent coupling inductances associated with 3-d interconnect structures. *IEEE Transactions on Components, Packaging, and Manufacturing Technology: Part B*, 19(2):283–288, 1996.

[55] S. Smale. On the mathematical foundations of electrical circuit theory. *J. Differential Geom.*, 7(1-2):193–210, 1972.

[56] V Sreeram. On the properties of frequency weighted balanced truncation techniques. In *American Control Conference, 2002. Proceedings of the 2002*, volume 3, pages 1753–1754. IEEE, 2002.

[57] V Sreeram and BDO Anderson. Frequency weighted balanced reduction technique: A generalization and an error bound. In *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, volume 4, pages 3576–3581. IEEE, 1995.

[58] Sheldon Tan and Lei He. *Advanced model order reduction techniques in VLSI design*. Cambridge University Press, 2007.

[59] Dmitry Missiuro Vasilyev. *Theoretical and practical aspects of linear and nonlinear model order reduction techniques*. PhD thesis, Massachusetts Institute of Technology, 2008.

[60] Christian de Villemagne and Robert E Skelton. Model reductions using a projection formulation. *International Journal of Control*, 46(6):2141–2169, 1987.

[61] Jiri Vlach and Kishore Singhal. *Computer Methods for Circuit Analysis and Design*. John Wiley & Sons, Inc., New York, NY, USA, 1983.

[62] Jiri Vlach and Kishore Singhal. *Computer Methods for Circuit Analysis and Design*, chapter 10. John Wiley & Sons, Inc., New York, NY, USA, 1983.

[63] G Wang, Victor Sreeram, and WQ Liu. A new frequency-weighted balanced truncation method and an error bound. *IEEE Transactions on Automatic Control*, 44(9):1734–1737, 1999.

[64] M Ronald Wohlers. *Lumped and distributed passive networks: a generalized and advanced viewpoint*. Academic press, 2017.

[65] Ngai Wong and Venkataramanan Balakrishnan. Fast balanced stochastic truncation via a quadratic extension of the alternating direction implicit iteration. In *Proceedings of the 2005 IEEE/ACM International conference on Computer-aided design*, pages 801–805. IEEE Computer Society, 2005.