NONLINEAR TRANSIENT ANALYSIS BASED ON WAVE DIGITAL FILTERS

by

Hui Xu

A thesis submitted to the faculty of graduate studies
Lakehead University
in partial fulfillment of the requirements for the degree of
Masters of Science in Engineering

Electrical and Computer Engineering

Lakehead University

December 2010

Lakehead

UNIVERSITY

OFFICE OF GRADUATE STUDIES

NAME OF STUDENT: Hui Xu

DEGREE AWARDED: Masters of Science in Engineering

ACADEMIC UNIT: Electrical and Computer Engineering

TITLE OF THESIS: **NONLINEAR TRANSIENT ANALYSIS BASED ON WAVE DIGITAL FILTERS**

This thesis have been prepared
under my supervision
and the candidate has complied
with the Master's regulations.

____

Signature of Supervisor

____

Date

Abstract

Simulation of large circuit networks is a CPU-time and memory consuming task. Computer clusters and multi-core processors have become common but algorithms must be parallelizable to fully take advantage of these architectures. Therefore, it is of great interest to find efficient parallel algorithms for circuit simulation. In this thesis an algorithm based on a Wave digital Filter (WDF) model of the circuit combined with waveform relaxation is presented for the first time. The new approach is tested with the simulation of a nonlinear transmission line. Several variations of the algorithm and the potential for parallelization are analyzed.

Topics relevant for the work in this thesis are presented first followed by transient simulation results using a Jacobi-like algorithm applied to two different WDF models of a nonlinear transmission line. The simulation results are shown to be in agreement with the results obtained using a traditional method. The Waveform Relaxation implementation is presented next followed by a set of simulation results and a discussion. Simulation results indicate that the approach requires more CPU time than traditional methods. Some suggestions for future improvements are given in the last chapter of this thesis.

Acknowledgments

First, I would like to thank Dr. Carlos E. Christoffersen, for his continuous support in my Master program. It is an honor for me to work in his research group. Without his generous support and patient guidance throughout my graduate studies, this thesis would not have been possible.

I also want to thank Dr. Abdelhamid Tayebi, Dr. Krishnamoorthy Natarajan and Dr. A. Manzak for their instruction during course study; thank Dr. and Dr. for reviewing this thesis and providing instructive comments.

I am indebted to my many of my graduate students colleagues and my friends, for all the help and encourage.

Finally, I would like to say thank to my parents. I know I can always have your encouragement and support.

Hui Xu
hxu2@lakeheadu.ca

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation and Objectives of This Study

Simulation of a large circuit network is a challenging task. New technologies make it possible solve complex systems using numerical methods. But without an efficient simulation method, a simulation of nonlinear circuit network still consume lots of computer time and memory. Since computer cluster and multi-core processors are widely used now, a parallel computing method can utilize the power of modern technology. Thus there is great interest to find a efficient parallel simulation method. A Jacobi-like iteration and a waveform relaxation method for simulating a nonlinear transmission line are presented in this thesis. These approaches can be parallelized to improve the efficiency of the computation.

Every numerical solution will bring unavoidable errors during the procedure of time discretization and solving nonlinear equations. Wave digital filter (WDF) can preserve important properties such as passivity or losslessness of an analog circuit in a digital system, and make WDF a good candidate for circuit simulation.

One objective of this thesis is to study transient simulation of circuits based on WDF principles. A Jacobi-like parallel algorithm based on WDF theory has been studied, which is implemented using basic WDF adaptors. An improved approach using custom WDF adaptors is presented next. This approach tries to combine several basic WDF adaptors to achieve a simplified and efficient WDF implementation for improving circuit simulation speed.

The main contribution of this thesis is the combination of Waveform Relaxation (WR) and WDF techniques. An approach which is a combination of Jacobi-like iteration method and WR provides advantages for parallel computing. WR is a relaxation method used in the time domain circuit simulation [21]. In parallel computing, there are several factors that affect program execution time. Communication overhead is one of the main ones. The Jacobi-like algorithm studied in this thesis is easy to implement in a parallel computing program. This algorithm demands frequent data exchange among processors for each sample time step, which is a bottleneck for parallel program efficiency. Instead of computing each time step separately the WR calculate a whole time interval at once, which significantly reduces communication frequency.

1.2 Thesis Overview

This thesis is composed of five chapters. Chapter 2 presents the review of topics relevant for the work in this thesis. Chapter 3 include the study of Jacobi-like algorithm and its simulation results based on WDF. An improvement with combined WDF adaptor is also included in this chapter. A WR approach based on WDF is studied in Chapter 4. A set of simulations based on WR and a "window" technical are discussed. The possibility of increasing parallel computing efficiency is also presented. The final chapter presents the conclusions of this study and suggestions for future work.

Chapter 2

Literature Review

2.1 Introduction

This chapter summarizes topics relevant to this thesis. The basic concept of Wave Digital filter Theory (WDF) is introduced first, which includes the derivation of the basic WDF elements, the concept of WDF interconnections and modeling of non-linear elements. A parallel algorithm based on the WDF is presented subsequently. Message-Passing Interface (MPI) and Waveform relaxation are summarized at the end of this chapter.

2.2 WDF Theory

2.2.1 Introduction

WDF(Wave Digital Filter) theory [1] was developed in 1960's by Alfred Fettweis and is extensively used to digitize analog circuit network. WDF attempt to translate analog filters into the digital model and preserve as much of the underlying physics of analog filters as possible. Particularly, through Fettweis's procedure the digital counterpart of a analog filter keeps the same precise network topology and energetic properties of analog filter.

Basic concept of WDF theory can be fairly simple [2]; In time domain, the characteristics of analog circuit components are commonly represented by a voltage-current relationship, while in WDF domain, the characteristics of circuit component are represented with a relationship between incident and reflected wave. Thus a circuit element definition in WDF domain is based on a equivalent characterization of voltage-current relationship. Although it can be regarded as a transformation of variable, it has the advantage to describe the dynamic behavior of a circuit network. Consider an analog circuit element (N-port device or basic element), energy coming from the network (connected with the element through a port) incident on the circuit element, the circuit element may store energy within itself, transmit energy through other port or ports to the network, or only reflect energy back by the same port. Incident wave and reflected wave are called wave variables and energy is carried by these waves. An arbitrary constant called port resistance is assigned to each individual port, and port resistance determine the reflectances and transmittances of ports. Reflectance and transmittance of wave port construct coefficient, and the coefficient parametrizes the entire network. For a passive network, these coefficient are bounded independently of inductances, capacitances, and resistances etc.

Wave Variables

Wave variables, also known as wave quantities from classical circuits [1, 3], characterize the one-ports elements or generally N-ports in WDF. In the work of Fettweis [1], for realizability, he mentioned that the voltage wave quantities from scattering parameter theory [3] are more suitable than power-wave quantities. In this thesis, voltage wave quantities will be adopted as follows in Figure 2.1 (N denotes circuit element or circuit network) and Equation (2.1) :

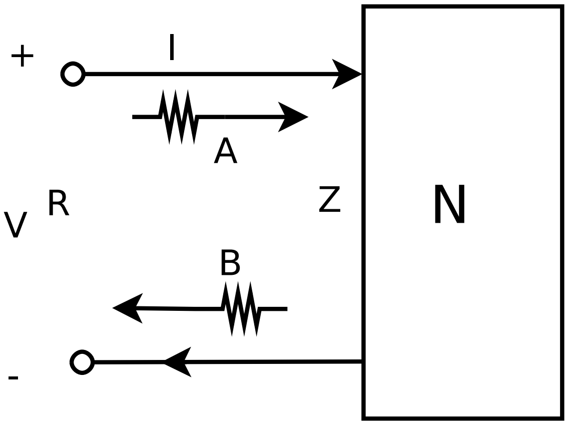


Figure 2.1: Voltage wave variables

$$
\begin{aligned}
A &= V + IR \\
B &= V - IR
\end{aligned}
\tag{2.1}
$$

In z domain A is defined as incident wave, B is defined as reflected wave which reflect back from the same port; or more precisely, A and B represent the wave transmitted in the forward and backward directions, respectively, for a port. R, which is an arbitrary positive constant, is called a port resistance for certain port. As wave variables are linear combinations of voltages and currents, the Voltage-current relationship can be described in the form of wave variables.

$$
\begin{cases}
V = \dfrac{A+B}{2} \\
I = \dfrac{A-B}{2R}
\end{cases}
\tag{2.2}
$$

Considering the impedance, $Z = V/I$ the reflection coefficient can be obtained

$$
\left.\begin{aligned}
A &= IZ + IR \\
B &= IZ - IR
\end{aligned}\right\} \Rightarrow \frac{A}{B} = \frac{Z+R}{Z-R}
$$

$$
Z \neq R \tag{2.3}
$$

Using S to denote the transfer function which only depend on the port resistance R and Impedance Z. Rewrite the function (2.3) in the form shown below.

$$B = SA \qquad S = \frac{Z - R}{Z + R} \tag{2.4}$$

2.2.2 WDF Elements

WDFs can be considered to be digital models of their continuous-time reference network. Mathematically, a digital filter which has a continuous-time reference filter can be described using difference equations [1]. These difference equations should be ordered sequentially, thus digital filter can be implemented by the arithmetic operations which are fully described by difference equations.

Bilinear Transformation

The simplest and most appropriate method for implementing a digital filter is bilinear transformation [1].To achieve discretization [2], a kind of spectral mapping should be carried out between the analog complex frequency variable s and an appropriate discrete frequency variable z, and $z = e^{sT}$.

$$z = e^{sT} = \frac{e^{\frac{sT}{2}}}{e^{-\frac{sT}{2}}} \approx \frac{1 + \frac{sT}{2}}{1 - \frac{sT}{2}} \tag{2.5}$$

$$s \to \frac{2}{T}\frac{1 - e^{-sT}}{1 + e^{-sT}} = \frac{2}{T}\frac{1 - z^{-1}}{1 + z^{-1}} \tag{2.6}$$

Bilinear transformation has an important property [19]: If a transfer function of a LTI (linear, time-invariant) system is stable and causal in s domain, it could be mapped to the z domain with the discrete variable z by the bilinear transformation, and the transfer function in z domain is still stable and causal. This is shown in Figure 2.2 [2, 19].

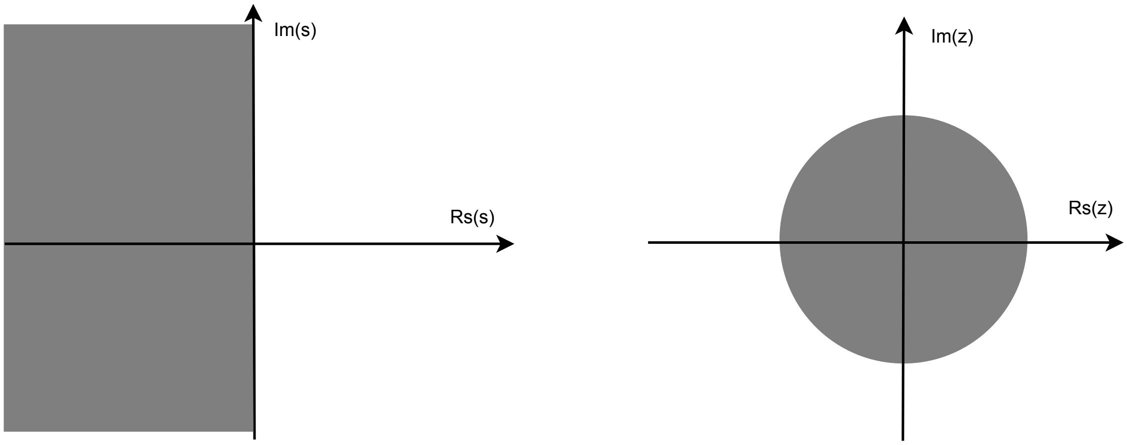


Figure 2.2: Spectral mapping corresponding to the trapezoidal rule

The trapezoidal rule for numerical integration is used for time domain interpretation of bilinear mapping [2]. z^{-1} is regarded as the unit delay, then the righthand side of (2.6) serves as an approximation to the derivative in discrete-time setting. Thus the derivative in discrete-time can be writen in the form $T/(1-\delta_t)/(1+\delta_t)$ where δ_t is a unit shift defined by $\delta_t x(t) = x(t-T)$. In this form some basic linear element could be easily written in difference equation with voltage-current relationship.

Example for Element Derivation

Using an inductor as an example, we can represent circuit elements using wave digital equivalents. The inductor as an one-port element can be described using a voltage-current relationship. In time domain it is:

$$v = L\frac{di}{dt} \tag{2.7}$$

Or for the WDF reference domain

$$V = sLI \tag{2.8}$$

$Z = sL$ is used as the impedance. Applying bilinear transformation

$$V = \frac{2L}{T}\frac{1-z^{-1}}{1+z^{-1}}I \tag{2.9}$$

Using wave variables, the voltage-current relationship is changed to the wave relationship. Substituting Equation (2.1) into (2.9) , the inductor can be described in term of wave quantities as follows:

$$\frac{B}{A} = \frac{\frac{2L}{T}\frac{1-z^{-1}}{1+z^{-1}} - R}{\frac{2L}{T}\frac{1-z^{-1}}{1+z^{-1}} + R}\,, \tag{2.10}$$

if the port resistance is chosen to be $R = 2L/T$, then

$$B = -z^{-1}A\,. \tag{2.11}$$

In time domain, applying the trapezoidal rule could get an equivalent result; according to the trapezoidal rule, voltage-current relationship of inductor can be expressed as

$$i(t) = i(t-T) + \frac{1}{L}\int_{t-T}^{t} v(\tau)d\tau \tag{2.12}$$

$$i(t) \approx i(t-T) + \frac{T}{2L}\left(v(t) + v(t-T)\right) \tag{2.13}$$

As mentioned in equation (2.2), wave variables can be described in voltage-current relationship in time domain like this

$$\begin{aligned} v(t) &= \frac{a(t)+b(t)}{2} \\ i(t) &= \frac{a(t)-b(t)}{2R} \end{aligned} \tag{2.14}$$

Here $a(t)$ and $b(t)$ denote the wave variables in time domain. Substituting Equation (2.14) into (2.13), results in the following

$$\frac{a(t)-b(t)}{2R} = \frac{a(t-T)-b(t-T)}{2R} + \frac{T}{2L}\left(\frac{a(t)+b(t)}{2} + \frac{a(t-T)+b(t-T)}{2}\right)$$

Using a port resistance equal to $R = 2L/T$, the difference equation in time domain can be obtained as follows:

$$b(t) = -a(t-T) \tag{2.15}$$

Comparing Equation (2.11) and Equation (2.15), the incident wave on the inductor will be observed and the product of a minus unit delay and incident wave is the reflected wave. These two equations are the same but in different domain. It is easy to tell reflected wave B or $b(t)$ (in time domain) value from the trapezoidal derivation which shows clearly reflected wave is the negative value of incident wave of previous time step.

WDF Model of Major Circuit Elements

A similar derivation can be made for other circuit elements. Major one port element realizations in WDF domain are summarized in the Table 2.1

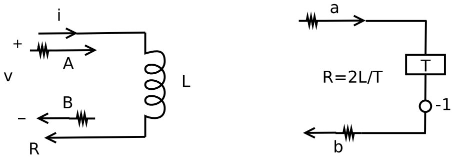
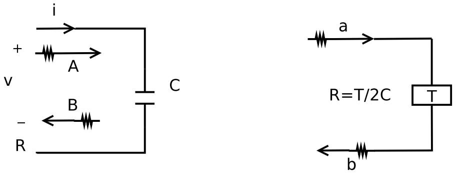
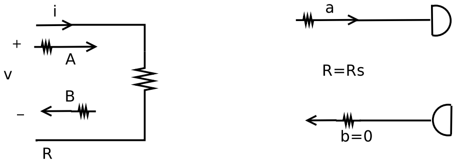
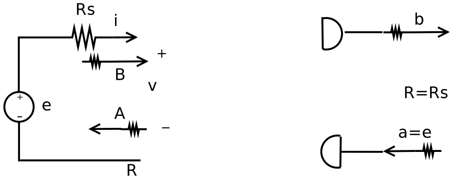
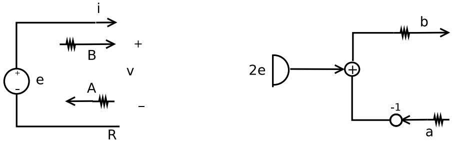
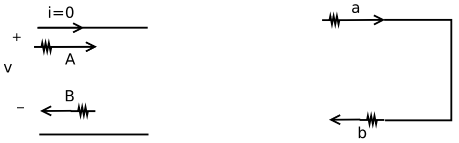
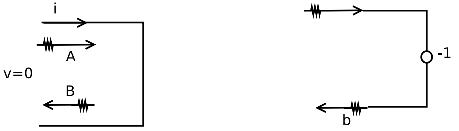
Elements Chart	
Difference Equation	Circuit Elements Analog and WDF Model
$B = -z^{-1}A \qquad b(t) = -a(t-T)$	 i + A v B – R L a R=2L/T T -1 b
$B = z^{-1}A \qquad b(t) = a(t-T)$	 i + A v C B – R a R=T/2C T b
$B = 0 \qquad b(t) = 0$	
$B = E \qquad b(t) = e$	 Rs i + B v e A – R b R=Rs a=e
$B = 2E - A \qquad b(t) = 2e - a(t)$	
$B = A \qquad b(t) = a(t)$	
$B = -A \qquad b(t) = -a(t)$	

Table 2.1: Major Circuit Elements WDF Realization

Circulators

A circulator has n ports ($n \geq 3$) [1]. For an ideal n-port circulator, the waves of its ports must satisfy the following:

$$B_1 = A_n, \quad B_2 = A_1, \quad \ldots \quad B_n = A_{n-1}$$

For a realistic circulator, wave variables do not exactly follow the equations as $B_1 = A_n, \quad B_2 = A_1, etc.$. For the ideal three ports circulator, realization is shown in Figure 2.3.

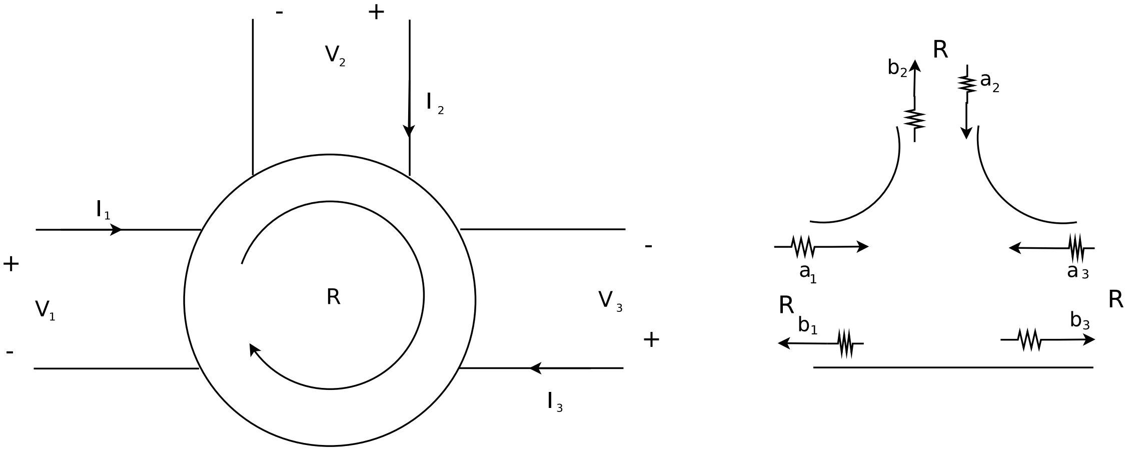


Figure 2.3: WDF Realization of Circulator

Delay-Free Directed Loop

In a signal-flow diagram, if each of the branches of a loop has the same orientation with respect to a given loop orientation, the loop is called directed. In a loop, the sum of delays in branches which are oriented in the same way minus the sum delays in oriented oppositely branches is the total delay of a given loop. If there is no delay elements in a branch, path, or loop, then it is called delay-free [1]. Thus a Delay-Free Directed Loop (DFDL) is a directed loop without delay in it (Figure 2.4). In other words, if a reflected wave depends simultaneously on its incident wave, a DFDL occurs in WDF realization. Basic WDF model like ideal source, short circuit and open circuit are often considered as elements without delay. Nonlinear capacitor and inductor, whose reflected wave is depends on their incident wave simultaneously, are often treated as elements without delay.

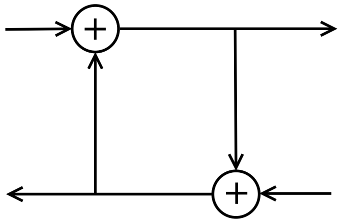


Figure 2.4: Delay free directed loop

In Fettweis's work [1], he gave the theorem for full-synchronic circuit realization: "the signal-flow diagram of a proper digital filter is realizable at a rate $F = 1/T$ if and only if it satisfies the following conditions:

- It does not contain delay-free directed loops.
- The total delay in any loop (directed or not) is equal to a multiple (zero, positive, or negative) of T."

Therefore, in the procedure of WDF realization a circuit it is important to avoid or cut DFDLs

2.2.3 Interconnections and Adaptors

From the last section, basic circuit elements and sources could be simulated in the digital wave domain. But the circuit topology and topological rules (Kirchoff's laws) should be present in WDF domain too, hence interconnections between different parts of the circuit network should also be simulated to establish the whole reference circuit in digital wave domain [1]. The digital representations of interconnections between circuit network parts are called adaptors, memory-less devices which pass energy carried by waves among different parts of circuit.

Series and Parallel Adaptors Derivation

The most common adaptors are series and parallel adaptors, which describe the most important connections in Kirchhoff's laws, series and parallel connections.

Here we consider a series connection which has n ports and each port resistance $R_j > 0, j = 1, \ldots, n$ is shown in Figure 2.5.

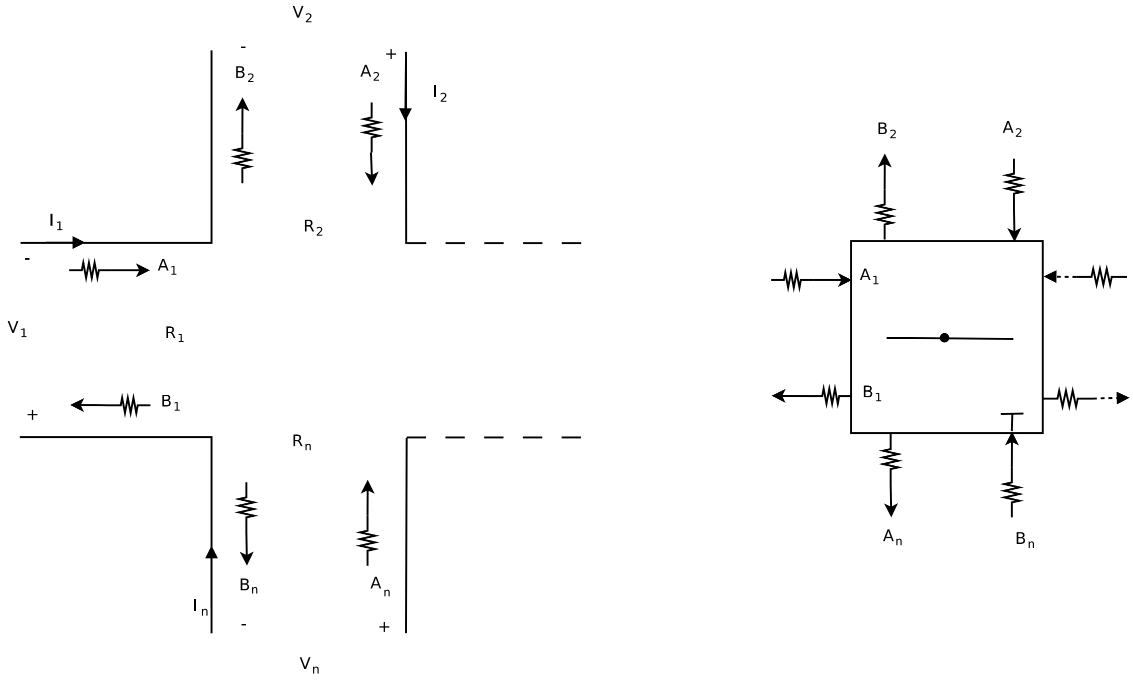


Figure 2.5: Series connection and series adaptor

According to Kirchoff's laws, we have:

$$\sum_{j=1}^{n} v_j = 0$$
$$i_1 = i_2 \ldots = i_n$$

In terms of wave variables, the equation could be written as follows

$$\begin{aligned}\sum_{j=1}^{n}\frac{a_j+b_j}{2} &= 0\\ \frac{a_1-b_1}{2R_1} &= \frac{a_2-b_2}{2R_2}\ldots=\frac{a_n-b_n}{2R_n}=i\end{aligned}$$

As at all ports, the currents are all equal to i, obviously reflected wave at port j can be written in the form of:

$$b_j = a_j - 2R_j i$$

So the KVL equation can be rewritten as shown below:

$$\sum_{j=1}^{n}\frac{2a_j-2R_j i}{2}=0$$

$$i=\frac{1}{\sum_{j=1}^{n}R_j}\sum_{j=1}^{n}2a_j$$

Thus the input wave and output wave variables relation for series adaptor at k port can be expressed in the form of

$$b_k = a_k - \frac{2R_k}{\sum_{j=1}^{n}R_j}\sum_{j=1}^{n}a_j \qquad k=1,\ldots,N \tag{2.16}$$

Equation (2.16) is shown in matrix form:

$$\begin{aligned}\begin{pmatrix}B_1\\B_2\\\vdots\\B_n\end{pmatrix} &= \begin{pmatrix}(1-\beta_1) & -\beta_1 & \ldots & -\beta_1\\ -\beta_2 & (1-\beta_2) & \ldots & -\beta_2\\ \vdots & \vdots & \ddots & \vdots\\ -\beta_n & -\beta_n & \ldots & (1-\beta_n)\end{pmatrix}\begin{pmatrix}A_1\\A_2\\\vdots\\A_n\end{pmatrix}\\ \beta_k &= \frac{2R_k}{(R_1+R_2+\ldots+R_n)} \quad k=1,2,\ldots n \end{aligned}\tag{2.17}$$

The A and B denote vectors of incident and reflected waves, respectively. Define a matrix S as shown below,

$$S = \begin{pmatrix} (1-\beta_1) & -\beta_1 & \dots & -\beta_1 \\ -\beta_2 & (1-\beta_2) & \dots & -\beta_2 \\ \vdots & \vdots & \ddots & \vdots \\ -\beta_n & -\beta_n & \dots & (1-\beta_n) \end{pmatrix},$$

S is called scattering matrix [2]. The series adaptor equations can be described in the same form as (2.4)

$$B = SA.$$

There are two properties to be mentioned here, first is dependent port, second is the reflection-free port. Since $\sum_{k=1}^{n} \beta_k = 2$, the coefficient of one of ports can be eliminated. Choose port n as an example, the coefficient of port n can be determined from $\beta_n = 1 - \sum_{k=1}^{n-1} \alpha_k$, so port n is called dependent port. It will help in calculating WDF parameter of the adaptor.

Reflection-free port is any one port of a series adaptor in which the reflected wave is independent from its incident wave. As the port resistance can be chosen arbitrarily, one port resistant of the series adaptor ports can be expressed as follows

$$R_n = \sum_{k=1}^{n-1} R_k$$

So the coefficient of the corresponding port n is equal to 1, and port n is called reflection-free port. From the scattering matrix shown below we can clearly observe the behavior of the reflection-free port.

$$\begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{pmatrix} = \begin{pmatrix} (1-\beta_1) & -\beta_1 & \dots & -\beta_1 \\ -\beta_2 & (1-\beta_2) & \dots & -\beta_2 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \dots & 0 \end{pmatrix} \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{pmatrix}$$

The schematic of a parallel adaptor is show in Figure 2.6, and adaptor equations are shown below:

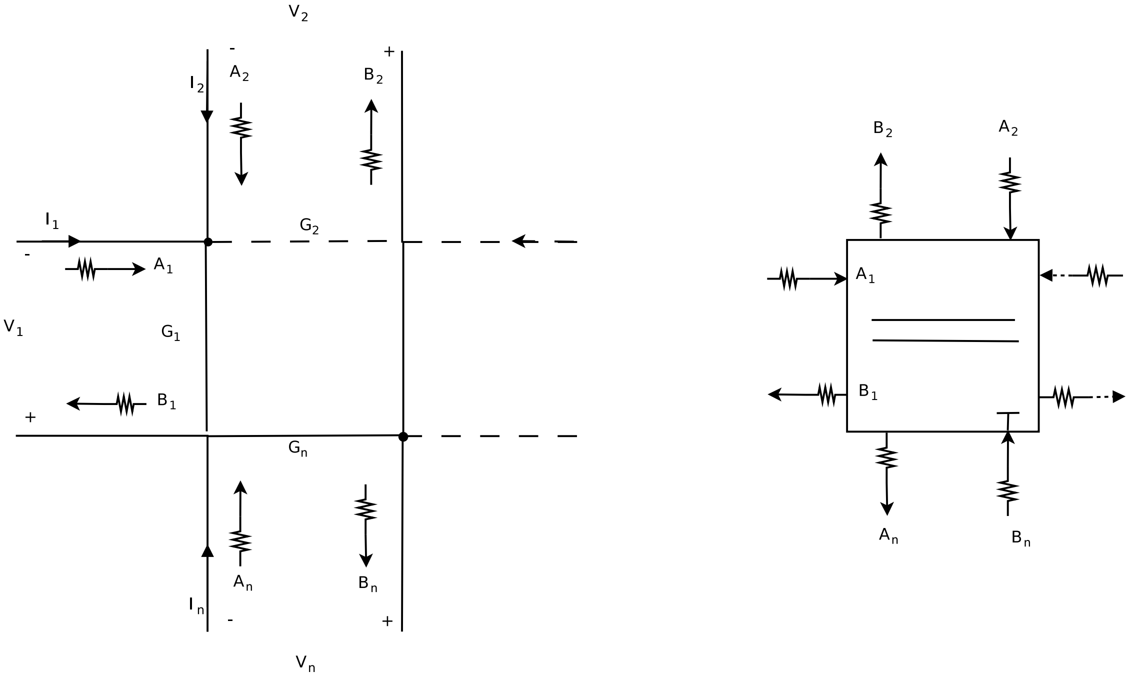


Figure 2.6: Parallel connection and parallel adaptor

$$
\begin{aligned}
b_k &= -a_k + \frac{2}{\sum_{j=1}^{n} G_j} \sum_{j=1}^{n} G_j a_j \qquad k = 1, \ldots, N \\
\begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{pmatrix} &= \begin{pmatrix} (\alpha_1 - 1) & \alpha_2 & \ldots & \alpha_n \\ \alpha_1 & (\alpha_2 - 1) & \ldots & \alpha_n \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1 & \alpha & \ldots & (\alpha_n - 1) \end{pmatrix} \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{pmatrix} \\
\alpha_k &= \frac{2G_k}{(G_1 + G_2 + \ldots + G_n)} \quad k = 1, 2, \ldots n. \quad G_k = 1/G_k \qquad (2.18)
\end{aligned}
$$

This is the same as series adaptor with α instead of β and G instead of R, any one port of a parallel adaptor can be chosen as a reflection-free port or dependent port.

Circuit interconnection Analysis Based on Graph Theory

Network topology theory is a very important concept for circuit simulation. Using WDF requires a way to handle the arbitrary network topologies. For topologies different from series or parallel connections, graph theory is used to derive the adaptor equations.

Some Graph Theory concepts are reviewed first [16, 17, 18].

Graph: A graph of an electrical network is a set of *branches* and a set of *nodes*. Branches are two-terminal elements, and the ends of elements are represented by nodes.

Path: A path in a graph is a sequence of nodes such that from each of its nodes there is an branch to the next node in the sequence.

Connected Graph: If there is at least one path between each pair of nodes, the graph is called a connected graph.

Loop: If a pair of nodes have more than one path, there is a loop.

Tree: In a connected graph, remove loops until there is only one path between any two nodes which don't have a loop, the result connected graph is called tree. Any branch in the tree is called one *twig* of a tree. The rest of branches form a *cotree* and are called *chords*.

Cut set: A cut divides the graph into two separate parts and only contains one twig of tree and as many chords as necessary. a set of branches included in a cut is called cut set.

Fundamental theorem of graphs: a connected graph with n nodes and b branches called Γ and T_r is a tree of Γ [16]:

- There is a unique path along the tree between any pair of nodes;

- There are $n-1$ tree branches and $l = b-(n-1)$ links;

- Every tree branch of T_r together with some links defines a unique cut set;

- Every link of T_r and the unique path on the tree between its two nodes defines a unique loop.

Kirchhoff's Current law and Kirchhoff's Voltage law are based on cut sets and loops, respectively. Using graph theory to make a set of hybrid (KVL and KCL) network equations for circuit network is described below [17] :

- For a particular tree, Kirchhoff's voltage law is applied to all of the basic loops, constructing a set of network equations by the form of KVL.
- For a particular tree, Kirchhoff's current law is applied to all of the basic cut sets, constructing a set of network equations by the form of KCL.

For a circuit network, a non-directed graph could express the topological information, but when KCL and KVL are applied, a directed graph is introduced to include the reference directions which are required by Kirchhoff's law.

A bridge structure showed in Figure 2.7 [4] is analyzed next. From graph theory and Kirchhoff's laws a scattering matrix can be derived. Nodes are named with letters A, B , C , D

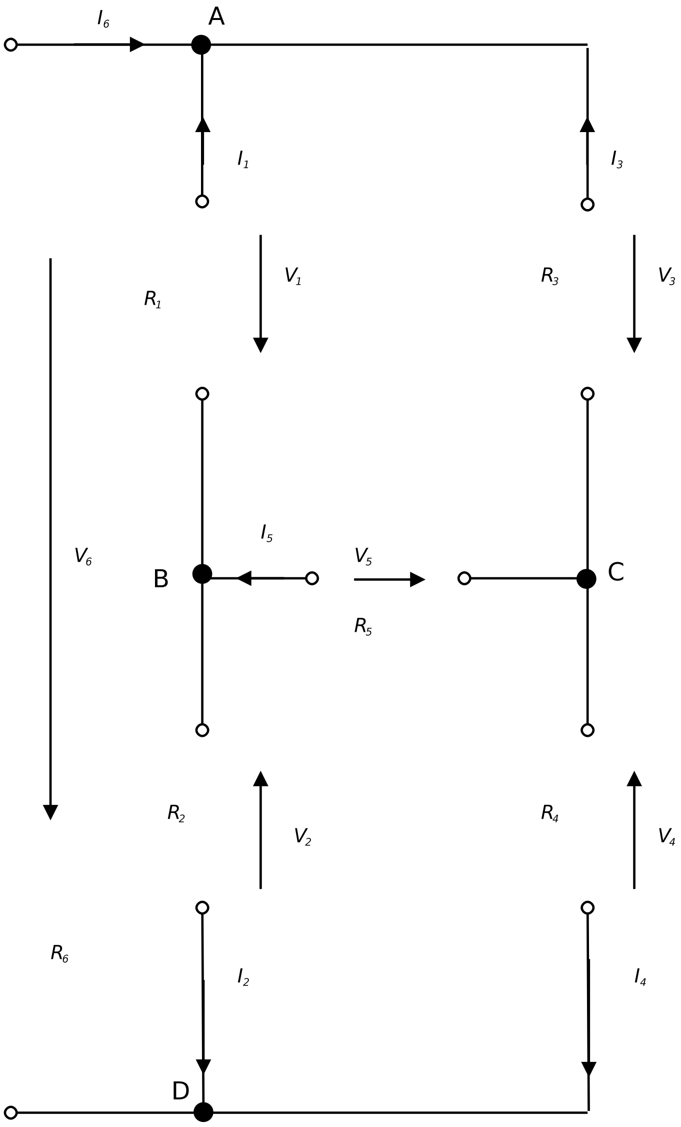


Figure 2.7: Diagram of a bridge structure

The basic loops and cut sets for the bridge structure are showed in Figure 2.8. Figure 2.8 illustrates the cut sets and loop sets where L_1, L_2 and L_3 are loop sets, and C_1, $C_2$and C_3 are cut sets. The thicker lines indicate the tree twigs. From the illustration tree branches and links can be calculated by equations $b = n - 1 = 3$ and $l = b - (n - 1) = 3$. Equations of KCL and KVL can be expressed as such:

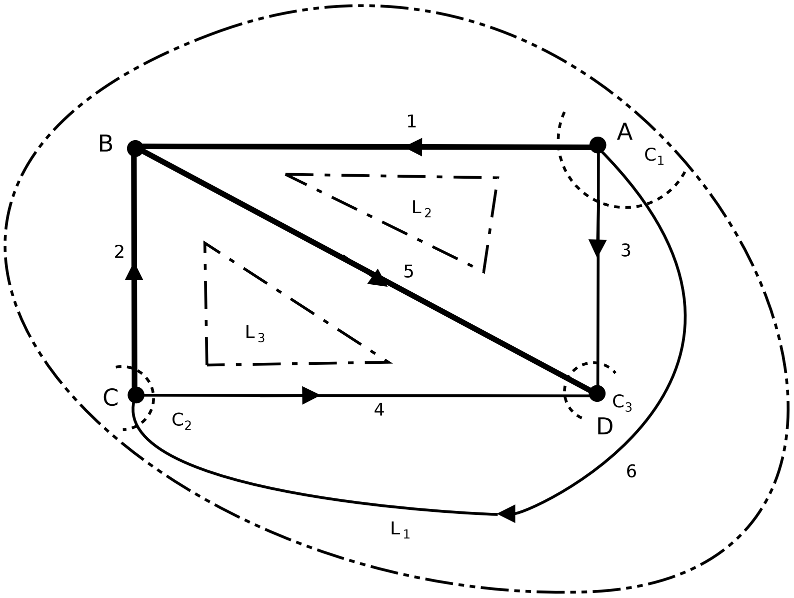


Figure 2.8: Basic loops and cut sets of the bridge structure

$$\begin{cases} V_1 - V_2 - V_6 = 0 \\ V_1 - V_3 + V_5 = 0 \\ V_2 - V_4 + V_5 = 0 \end{cases}$$

$$\begin{cases} I_1 + I_3 + I_6 = 0 \\ I_2 + I_4 - I_6 = 0 \\ I_3 + I_4 + I_5 = 0 \end{cases}$$

Defined voltage vector and current vector

$$\overrightarrow{V} = [V_1, \quad \dots \quad , V_6]^T$$

$$\overrightarrow{I} = [I_1, \quad \dots \quad , I_6]^T$$

KCl and KVL equations can be expressed in the matrix form:

$$\begin{bmatrix} 1 & -1 & 0 & 0 & 0 & -1 \\ 1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{bmatrix} = \overrightarrow{0}$$

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \end{bmatrix} = \overrightarrow{0}$$

Define M_V, M_I to be matrices for voltage and current equations, respectively.

$$M_V = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & -1 \\ 1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 & 1 & 0 \end{bmatrix} \Longleftrightarrow M_V \overrightarrow{V} = \overrightarrow{0}$$

$$M_I = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \Longleftrightarrow M_I \overrightarrow{I} = \overrightarrow{0}$$

Define wave variables below

$$\overrightarrow{A} = [A_1, \quad \dots \quad, A_6]^T$$

$$\overrightarrow{B} = [A_1, \quad \dots \quad, B_6]^T$$

$$\underline{G} = diag\left(\frac{1}{R_1}, \quad \dots \quad, \frac{1}{R_6}\right)$$

KVL and KCL equation can be written as

$$\overrightarrow{V} = \frac{\overrightarrow{A} + \overrightarrow{B}}{2} \Rightarrow M_V(\overrightarrow{A} + \overrightarrow{B}) = \overrightarrow{0}$$

$$\overrightarrow{I} = \underline{G}\frac{\overrightarrow{A} - \overrightarrow{B}}{2} \Rightarrow M_I(\overrightarrow{A} - \overrightarrow{B}) = \overrightarrow{0}$$

From the preceding, we can obtain the bridge structure adaptor, which like series and parallel adaptors could use a scattering matrix S to describe the relationship between incident and reflected waves:

$$\begin{bmatrix} M_V \\ M_I\underline{G} \end{bmatrix} \overrightarrow{A} + \begin{bmatrix} M_V \\ -M_I\underline{G} \end{bmatrix} \overrightarrow{B} = \overrightarrow{0}$$

$$\overrightarrow{B} = S\overrightarrow{A} \Rightarrow S = -\begin{bmatrix} M_V \\ -M_I\underline{G} \end{bmatrix}^{-1} \begin{bmatrix} M_V \\ M_I\underline{G} \end{bmatrix} \tag{2.19}$$

From the above derivation, circuit network topological information in voltage-current domain can be transformed to the WDF domain, and network equations can be expressed with a scattering matrix S. The derivation of scattering matrix S of the bridge structure is one example.

2.2.4 Wave description for Nonlinear Element

In “Wave Digital Filters: Theory and Practice” [1], Alfred Fettweis gave essential WDF elements blocks, which represent linear circuit elements and sources. Considering nonlinear elements, a nonlinear circuit element characteristic can be described by WDF theory under some conditions. In many works [6, 7, 8], nonlinear elements have been discussed, and several methods have been proved. Hence not only resistances having a continuous piecewise-linear voltage-current characteristic [6] can be described by WDF theory, but also an extended method of WDF principles can be used for modeling nonlinear dynamic elements [7, 8].

Nonlinear Resistor

A Nonlinear resistor voltage-current relationship can be defined by the form $F(v,i) = 0$. This constitutive relation function can be converted into a wave variables form function $f(a,b) = 0$ (Figure 2.9).

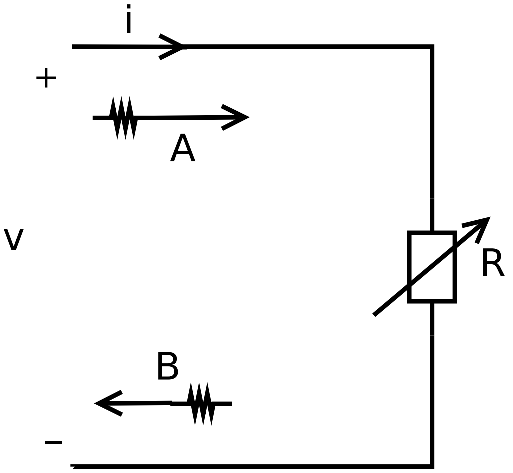


Figure 2.9: Model of nonlinear capacitance

$$f(a,b) = F(\frac{a+b}{2}, \frac{a-b}{2R}) = 0$$

Function $f(a,b) = 0$ is a implicit function. Thus the reflected wave (b) can be described by an explicit function $b = \tilde{f}(a)$. If the nonlinear resistor wave function, $b = \tilde{f}(a)$, and its derivatives are continuous, for a point (a_0, b_0) belonging to the range of the characteristic of resistor, the condition $\frac{\partial f}{\partial b}|_{a_0,b_0} \neq 0$ guarantees that in a neighborhood of a_0 there is a function $\tilde{f}(\bullet)$ such that $f(a, \tilde{f}(a)) = 0$ [6, 7].

Using the current-controlled resistor $F(v,i) = v - v(i) = 0$ as an example, the voltage-current relationship can be mapped onto wave domain.

$$f(a,b) = \frac{a+b}{2} - v(\frac{a-b}{2R}) = 0$$

Checking the condition for existence of $\tilde{f}(\bullet)$.

$$\frac{\partial f}{\partial b} = \frac{\partial}{\partial b}\left\{\frac{a+b}{2} - v(\frac{a-b}{2R})\right\} \tag{2.20}$$

$$= \frac{1}{2} - \frac{\partial v}{\partial i}\frac{\partial i}{\partial b} = \frac{1}{2} + \frac{1}{2R}v'\left(\frac{a-b}{2R}\right) \tag{2.21}$$

$$v'(i) = \frac{dv}{di} \tag{2.22}$$

Clearly under the condition $v'(i) \neq -R$, it is guaranteed that a function for reflected wave can be written in $b = \tilde{f}(a)$. The voltage-controlled nonlinear resistor can be treated in the same way. When invertibility is satisfied, the form of $b = \tilde{f}(a)$ can be used for nonlinear resistor and be implemented into the wave domain.

Nonlinear Elements with Memory

In several works [7, 8], an extension of WDF principle is used to model a class of nonlinear elements with memory in wave domain. This extension of the WDF principle introduces a new class of wave variables, and use the concept of mutator to treat nonlinear elements with memory as a nonlinear resistor [7].

Consider a scattering junction shown in Figure 2.10, which shows the transformation between two different classes of wave variables. Consider $R(z)$ and $R'(z)$ are *reference transfer function* (RTF) [7] in z domain, $A(z)$ $A'(z)$ are incident waves of junction, $B(z)$ $B'(z)$ are reflected waves. And let

$$\begin{aligned} A(z) &= V(z) + R(z)I(z) \\ B(z) &= V(z) - R(z)I(z) \\ A'(z) &= V'(z) + R'(z)I'(z) \\ B'(z) &= V'(z) - R'(z)I'(z) \end{aligned}$$

As $V(z) = V'(z)$ and $I(z) + I'(z) = 0$, reflected waves can be described like

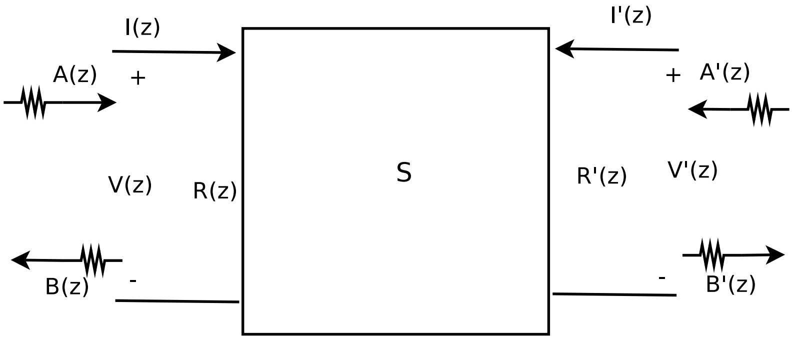


Figure 2.10: Scattering junction

$$
\begin{aligned}
B(z) &= A'(z) + K(z)(A(z) - A'(z)) \\
B'(z) &= A(z) + K(z)(A(z) - A'(z)) \\
K(z) &= \frac{R'(z) - R(z)}{R'(z) + R(z)}
\end{aligned}
$$

Signal flow chart shown in Figure 2.11 .

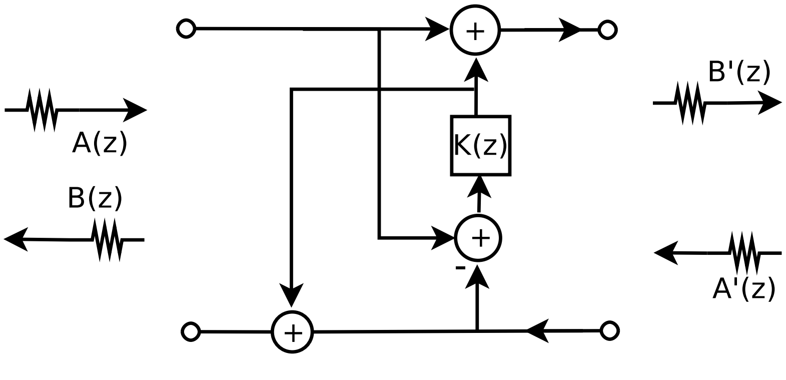


Figure 2.11: Signal flow of scattering junction

When considering the realization of the scattering junction, a key point is to avoid delay free loops when it connects with other circuits. In other words, incident waves entering the two port scattering junction do not depend on reflected waves of it. Thus $K(z)$ does not have instantaneous input/output connection is a necessary and sufficient condition for realization, for example, $K(z) = z^{-1}\hat{k}(z)$ when $\hat{K}(z)$ is causal and stable.

Figure 2.11 shows a structure that can transform a nonlinear capacitor into a nonlinear resistor, meanwhile still preserve the capacitor's nonlinear characteristic. A class of two port element called mutator has been discussed in L.O.Chua's work [10]. This class of element can be used to transform nonlinear capacitor into a nonlinear resistor in Kirchhoff domain. In wave domain similar counterpart of mutator is present by the wave digital mutator [11]. This mutator reduces the difficulty of realization of nonlinear element with memory.

Structure of wave digital mutator, for nonlinear capacitor could be called $R-C$ mutator, is a scattering junction connected with a capacitive RTF and a resistive RTF. For the structure shown in Figure 2.10 , using $R(s) = R$ and $R'(s) = 1/(sC)$, $C > 0$, the scattering junction can be rewritten by bilinear transformation

$$\begin{aligned} K(z) &= \frac{R'(z) - R}{R'(z) + R} \\ R'(z) &= \frac{T}{2C}\frac{1 + z^{-1}}{1 - z^{-1}} \end{aligned}$$

When choosing $R = T/2C$, we got $K(z) = z^{-1}$, which means for both ports of mutator instantaneous reflections have been eliminated. The signal flow chart for mutator is shown in Figure 2.12, where $K(z)$ is substituted by a unit delay z^{-1}.

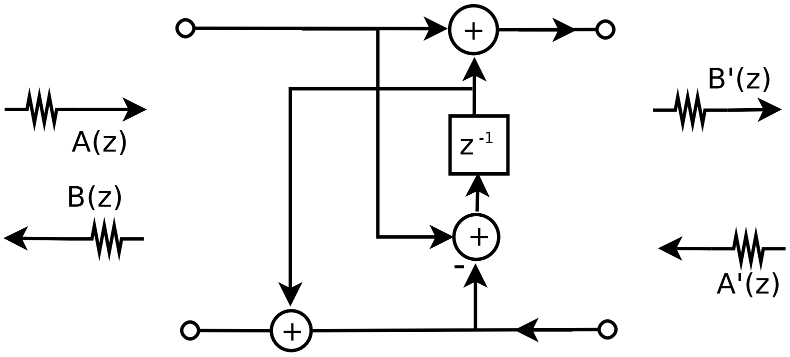


Figure 2.12: RC mutator

With the $R - C$ mutator, a nonlinear capacitor can be treated as a nonlinear resistor, and A' B' could be written in the form below

$$A'(z) = V'(z) + \frac{T}{2C}\frac{1+z^{-1}}{1-z^{-1}}I(z) = V'(z) + \frac{1}{C}Q(z)$$
$$B'(z) = V'(z) - \frac{T}{2C}\frac{1+z^{-1}}{1-z^{-1}}I(z) = V'(z) - \frac{1}{C}Q(z)$$

Where $Q(z)$ is associated with the electrical charge $q(t)$, $\dot{q}(t) = i(t)$. A nonlinear capacitor can be described by an algebraic relationship in form of $P(v,q) = 0$. C is the reference capacitance, and it plays a role for mapping nonlinear capacitor in Kirchhoff domain onto the wave domain. In Wave domain, $R-C$ mutator connects with the nonlinear map with a form $b = \tilde{f}(a)$.

For the nonlinear inductor, a $R-L$ mutator could be derived using the same procedure, and a similar signal flow chart can be found, Figure 2.13 shows its signal flow in z domain.

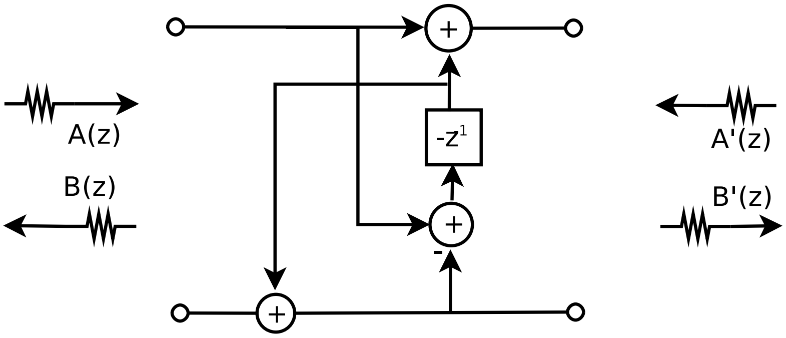


Figure 2.13: RLmutator

2.3 WDF in Circuit Simulation

Simulation is a well-known technique to study complex circuit systems. For many years WDF has been used for substituting classical analog filter by digital circuits [1, 2, 4, 6, 7, 8, 9, 11, 13, 14]. For the linear and passive systems, the WDF is stable and shows advantage compared with other simulation methods. Therefore the WDF is a good candidate for transient simulation. In many works [6, 7, 8], several approaches made it possible for describing nonlinear elements by WDF structure. Simulation based on WDF is possible for a circuit combined with linear and nonlinear elements [5, 9, 13].

2.3.1 Simulation of a Nonlinear Transmission line

A Jacobi-like method that can be applied to a nonlinear transmission line is presented in this section. The well-known Jacobi method has a strong local property, in other words, data for computing can be divided in many blocks which can be calculated independently [20](Appendix B).

Using a nonlinear transmission line which has been discussed by M. Toda [12] as an example, we can find out how DFDLs are introduced in WDF realization. A nonlinear transmission line model is shown in Figure 2.14 where linear inductor value is $1.38\mu H$. The transmission line has nonlinear capacitors with the characteristic shown below:

$$q = c_0 v_0 ln(1 + \frac{v}{v_0}) \qquad c_0 = 224.9pF, \quad v_0 = 3.73V$$

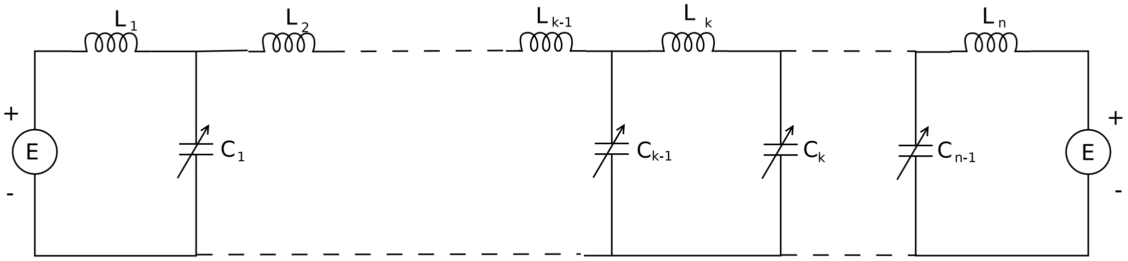


Figure 2.14: Nonlinear transmission line

Solution of voltage for the each nonlinear capacitor is given by M. Toda [12]. Voltage on kth nonlinear capacitor C_k shown below:

$$v_k(t) = \hat{v}\text{sech}^2[\omega(t \mp k/v)]$$

Where $\hat{v}$ denotes an arbitrary positive constant.

$$\omega = \sqrt{\frac{\hat{v}/v_0}{Lc_0}} \qquad v = \frac{\omega}{\operatorname{arcsinh}\sqrt{\hat{v}/v_0}}$$

Usually when nonlinear elements are introduced, DFDLs occurs, that means incident waves instantaneously depend on reflected waves. If there is only one nonlinear element in a circuit, DFDL can be eliminated using the reflection-free port of the adaptor (Figure 2.15). But for a complex circuit with more than one nonlinear element there is no easy way for the full-synchronic circuit realization. As an implicit equation can be used to describe DFDL, we either have to find a solution to solve implicit equations, or introduce an iterative formulation to cut DFDLs [13].

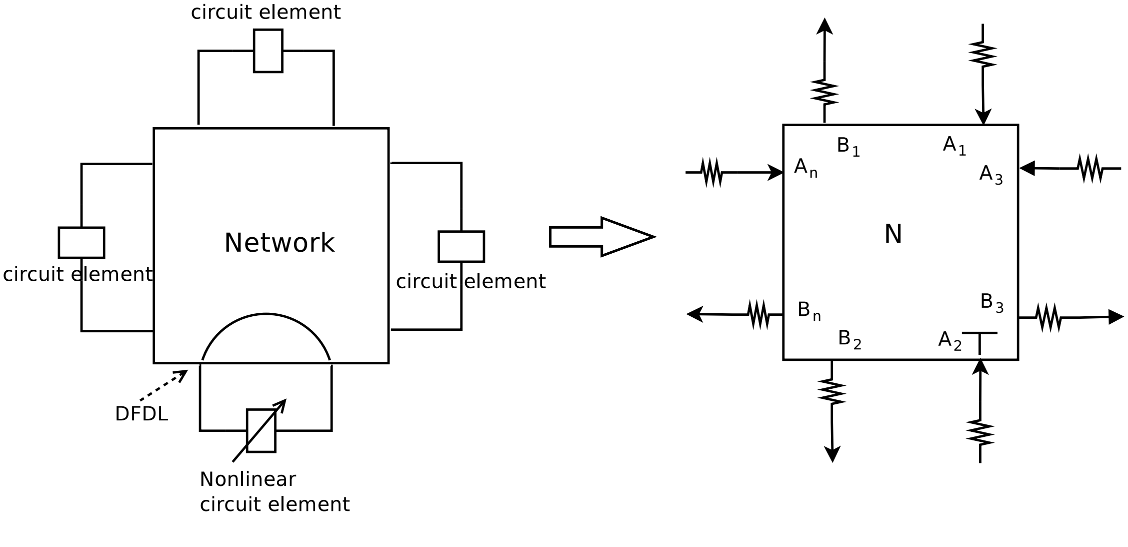


Figure 2.15: DFDL cut by reflection-free port of adaptor

One approach is use an iterative method at each time step to solve implicit equations which represent DFDLs. This iterative method uses an impedance insert into the path of the DFDL to remove the DFDL. Figure 2.16 illustrates the model used in this method.

For each time step $\overline{a}$ vector is the vector containing all known incident wave, reflected waves of $\overline{a}$ is $\overline{b}$. $\overline{a_0}$ vector contain waves incident in DFDLs path, reflected

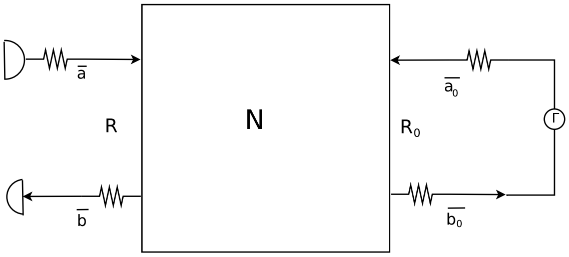


Figure 2.16: WD model for iteration

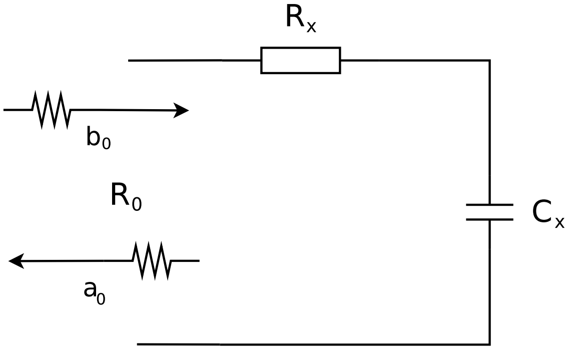


Figure 2.17: Impedance model

wave is $\overline{b_0}$ vector. Γ is the inserted reflectance. Network could be described as:

$$\begin{pmatrix} \overline{b} \\ \overline{b_0} \end{pmatrix} = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix} \begin{pmatrix} \overline{a} \\ \overline{a_0} \end{pmatrix}$$

Matrix S where $S_{ij} \quad i,j \in [1,2]$ denote a scattering matrix describes topology of known wave and DFDLs.

For the iteration, we demand reflectance $\Gamma|_{\psi=1} = 0$ at the beginning of iterations, where $\psi = (1 - z^{-1})/(1 + z^{-1})$ is the frequency . Finally inserted reflectance Γ achieves steady state, in other words, $\overline{a_0} = \overline{b_0}$, reflectance needs to satisfy $\Gamma|_{\psi=0} = 1$. A network that produces that kind type reflectance is shown in Figure 2.17. We consider $\Gamma = a_0/b_0$, and let $Z_c = (R_0 - R_x)/\psi$, where $\psi = (1 - z^{-1})/(1 + z^{-1})$.

$$\frac{\overline{a_0}}{\overline{b_0}} = \Gamma = \frac{R_x + \dfrac{(R_0 - R_x)}{\psi} - R_0}{R_x + \dfrac{R_0 - R_x}{\psi} R_0}$$

Substituting $\psi = (1 - z^{-1})/(1 + z^{-1})$ we get:

$$\frac{\overline{a_0}}{\overline{b_0}} = \Gamma = \frac{z^{-1}(R_0 - R_x)}{R_0 - z^{-1} R_x}$$

We can write $\overline{b_0}$ as a function of $\overline{a}$, $\overline{b_0} = S_{21} \times \overline{a} + S_{22} \times \overline{a_0}$ thus we get a Jacobi-like algorithm shown below

$$\overline{a_0(n)} = \gamma\overline{a_0(n-1)} + (1-\gamma)[S_{22}\overline{a_0(n-1)} + S_{21}\overline{a}] \qquad \gamma = \frac{R_x}{R_0} \tag{2.23}$$

This Jacobi-like iteration can be described like this: at each time step a linear R-C circuit and a circulator are inserted into DFDL. The iteration can be treated as an imaginary time dimension which does not affect reality. In this imaginary time dimension, linear R-C circuit makes circulator realizable. When R-C circuit achieves steady state in the imaginary time dimension, DFDL is cut and incident wave and reflected wave of DFDL achieve real value at this time step.

Thus modified nonlinear transmission line WD realization could be described as shown in (Figure 2.18)

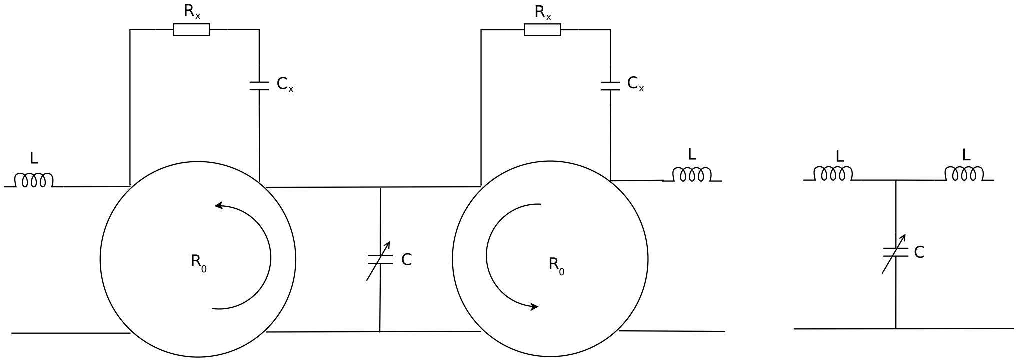


Figure 2.18: Jacobi-like WD realization model for nonlinear capacitor

2.4 MPI Introduction

MPI (Message-Passing Interface) is a message-passing library interface specification [22, 23], which addresses primarily the message-passing parallel programming model. Data is transferred from the address space of one process to another process through cooperative operations on each process. The MPI standard has been defined through an open process by a community of parallel computing vendors, computer scientists, and application developers.

The goal of the MPI is to establish a practical, portable, efficient, and flexible standard for message passing. A detail list of goals is shown as follows:

- Provide programming interface for application.
- Improve communication efficiency. Avoid memory-to-memory copying, allow overlap of computation and communication, and offload to communication co-processor, where available
- Allow for implementations that can be used in a heterogeneous environment.
- Assume a reliable communication interface: communication failures are dealt with by the underlying communication subsystem.
- Define an interface that can be implemented on many vendors platforms, with no significant changes in the underlying communication and system software.
- Semantics of the interface should be language independent.
- The interface should be designed to allow for thread safety.

2.4.1 Basic Schema for MPI Programming

Two basic schema for MPI programming are peer to peer communications schema and master slave communications schema. Most of parallel programs are based on either one of two schemas or a combination of two schema.

Using peer to peer communications schema, every processor sends data to its adjacent processors or collects data from them. Jacobi iteration for the nonlinear transmission line is a good example for peer to peer communications schema, as it calculate a new value based on the values which are adjacent from the last iterative step [20](Appendix B). Therefore Jacobi iteration can be parallelized, and for each processor, data is only needed to be exchanged with the adjacent processor. No dedicated processor is needed to collect and distribute data for all other processors. Figure 2.19 shows the basic concept.

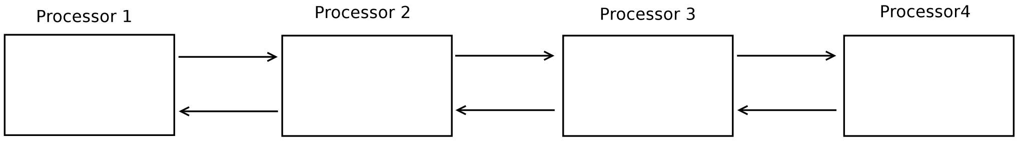


Figure 2.19: Peer to peer communication

There is a master processor in the master slave schema, which collects data from other processors and distributes data to them, as shown in Figure 2.20.

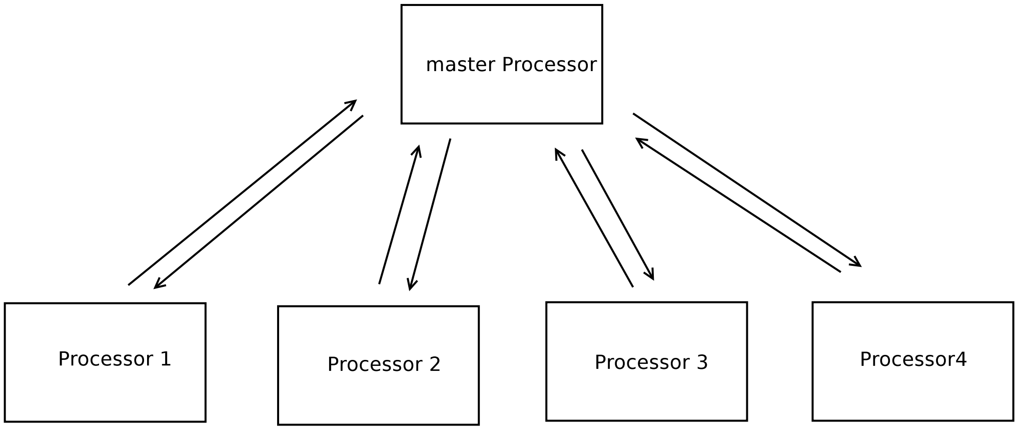


Figure 2.20: Master and slave communication

2.5 Waveform Relaxation

In A.R.Newton and A.L.Sangiovanni-Vincentelli's work [21], Waveform Relaxation (WR) is an important part of relaxation techniques for electrical simulation. Compared with timing analysis and iterated timing analysis ,which are at the nonlinear algebraic equation level, waveform relaxation is applied to differential equation level. The WR techniques can be explained as an analogue of the Gauss-Seidel algorithm [20] (Appendix B), where, unknowns are elements of a function space, rather than real variables.

Consider the first-order two-dimensional differential equation in $x(t) \in \mathbb{R}^2$ on $t \in [0, T]$,

$$\dot{x_1} = f_1(x_1, x_2, t), \qquad x_1(0) = x_{10} \tag{2.24}$$

$$\dot{x_2} = f_1(x_1, x_2, t), \qquad x_2(0) = x_{20} \tag{2.25}$$

The basic idea of the waveform-relaxation algorithm is to fix the waveform $x_2 : [0, T] \to R$ and solve as a one dimensional differential equation in $x_1(\cdot)$. The solution thus obtained for x_1 can be substituted into Equation 2.25 which will then be reduced to another first-order differential equation in one variable, x_2 . Equation 2.24 is then resolved using the new solution for $x_2(t)$, and the procedure is repeated.

Chapter 3

Nonlinear Transmission line Simulation

3.1 Introduction

In this chapter, a nonlinear transmission line is simulated by WDF principles. In simulations, all the incident and reflected waves are represented by lower case.

First, a simple nonlinear RLC circuit simulation result is presented. Simulation results are compared with differential equation solution solved by using the ODE45 Matlab function. Second, nonlinear transmission line is simulated by WDF principles. WDF analysis based on a parallel algorithm is discussed. Finally a different set of adaptors are applied on the WDF analysis for the transmission line and parallel algorithm.

The simulation results are compared with an circuit simulator named Carrot [25]. Carrot is a circuit simulator based on C++ developed by Dr Carlos E. Christoffersen.

3.2 Simulation of Nonlinear RLC Circuit

When a circuit has one nonlinear elements, delay-free direct loops occurs. By the theorem for full-synchronic circuit realization [1], these kind of circuits can not be realized. To simulate circuit with nonlinear elements, DFDL must be cut. There are several methods that can be used for cutting DFDL.

3.2.1 Cutting DFDL with Reflection-free Port

For a simple RLC circuit with one nonlinear element shown in (Figure 3.1), the common method is to use an important property of WDF interconnection adaptor to cut DFDL, which is the reflection-free port. By matching port resistance of the port which is connected with nonlinear element, the incident wave to the adaptor port is not affected by reflected wave at the same port. This method can be applied not only on series or parallel adaptor, but can also be applied on arbitrary interconnection adaptors which can be described by a scattering matrix [5].

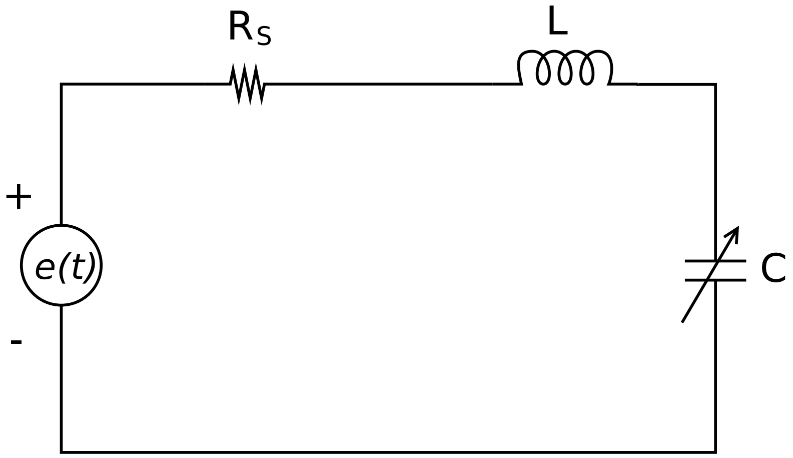


Figure 3.1: Nonlinear RLC circuit

The nonlinear RLC circuit condition are given by:

$$e(t) = \begin{cases} E & t \leqslant 200ns \\ 0 & t > 200ns \end{cases}$$

$$L = 1.38\mu H, \qquad R_s = 10\Omega$$

Nonlinear capacitor equation is as follows:

$$q = c_0 v_0 ln(1 + \frac{v}{v_0}), \qquad v > -v_0 \qquad c_0 = 224.9pF, \qquad v_0 = 3.73V$$

Figure 3.2 shows the voltage-charge controled nonlinear capacitor characteristic in WDF domain. The x-axis represent voltage on capacitor, y-axis represent the wave variable value corresponded with voltage. Implementation of the RLC circuit in WDF domain can be described in two parts: linear and nonlinear element.

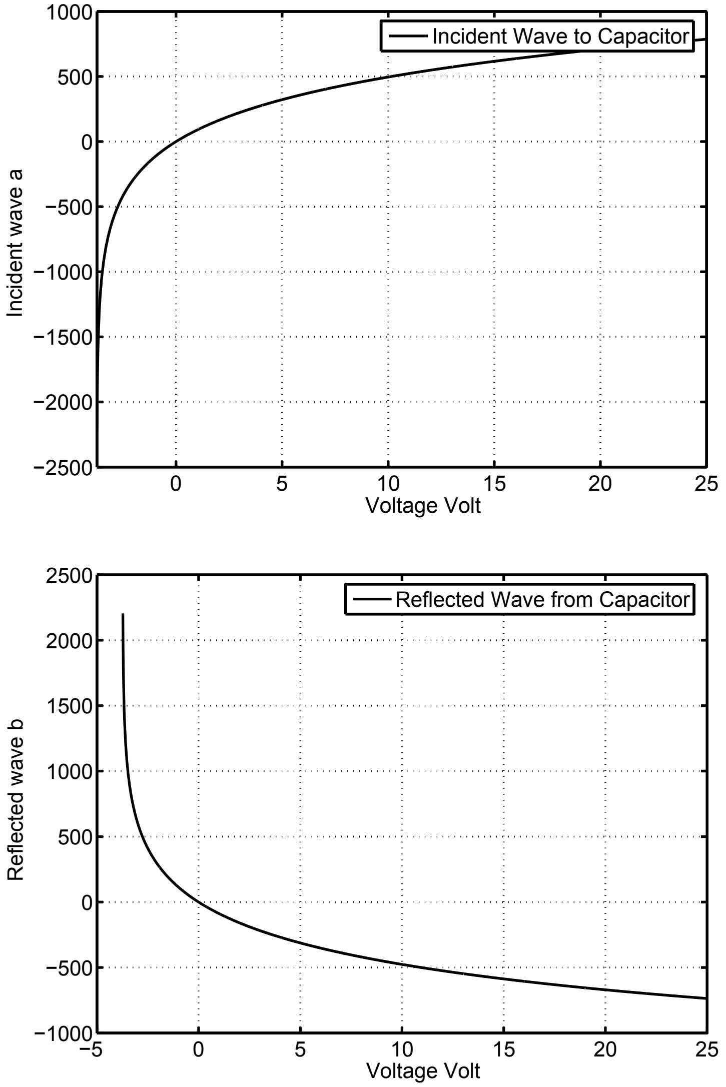


Figure 3.2: Nonlinear capacitor WDF characteristic correpsoned with voltage

The linear inductor and voltage source with resistor can be connected with a series adaptor directly. With wave digital mutator, the nonlinear capacitor connects with the series adaptor. WDF implementation is shown in (Figure 3.3)

Port three is chosen to be a reflection-free port, the reflected wave of it does not depend on incident wave coming from nonlinear capacitor. T is the unit delay, and $R_1 = R_s$ is the internal resistance of voltage source, $R_2 = 2L/T$. Reference capacitance of the mutator is C, reflection-free port resistance and reference capacitance

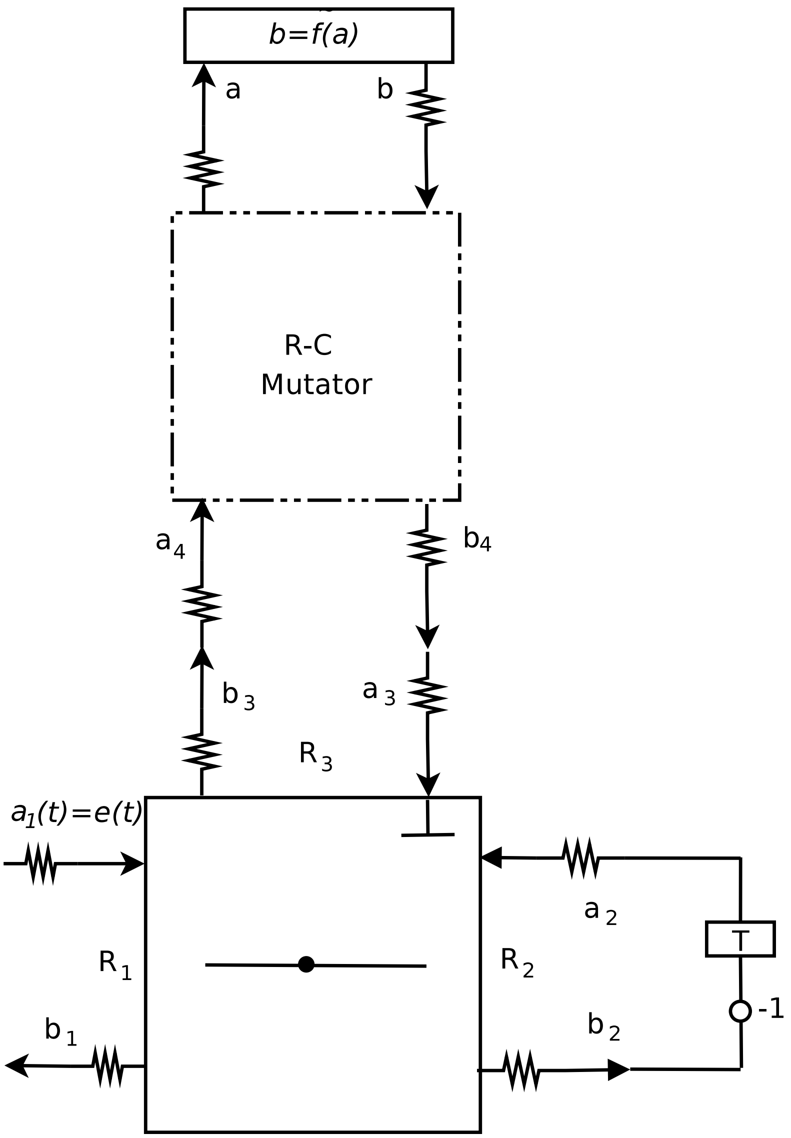


Figure 3.3: WDF implement of nonlinear RLC circuit

can be found by

$$R_1 + R_2 = R_3 = \frac{T}{2C} \Rightarrow C = \frac{T/2}{R_s + 2L/T}$$

Scattering matrix of series adaptor can be described as follows:

$$
\begin{aligned}
\begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} &= \begin{pmatrix} (1-\beta_1) & -\beta_1 & -\beta_1 \\ \beta_1 - 1 & \beta_1 & \beta_1 - 1 \\ -1 & -1 & 0 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \\
\beta_1 &= \frac{2R_s}{2(R_s + 2L/T)}
\end{aligned}
$$

To implement the nonlinear capacitor and mutator, wave variables can be described as follows:

$$b(t) = \tilde{f}(a(t))$$

$$\begin{cases} a(t) = a_4(t) + a_4(t-T) - b(t-T) \\ b_4(t) = b(t) + a_4(t-T) - b(t-T) \end{cases}$$

$$\begin{cases} a_4 = b_3 \\ b_4 = a_3 \end{cases}$$

Simulations are implemented in Matlab. For WDF simulation, $T = 1ns$, and use linear interpolation method (table lookup) for finding solution of function $b = \tilde{f}(a)$. RLC circuit is also simulated with Kirchhoff's voltage-current equations, while the Runge-Kutta method (ODE45) is used for solving differential equation. ODE45 result compares with the result of WDF shown as (Figure 3.4). From result we only observe little difference between these two methods. A possible reason for the difference between ODE45 and WDF simulation is that error of linear interpolation accumulates during the WDF calculation. In next section, Newton method will be applied on nonlinear function to achieve higher efficiency and accurate error result.

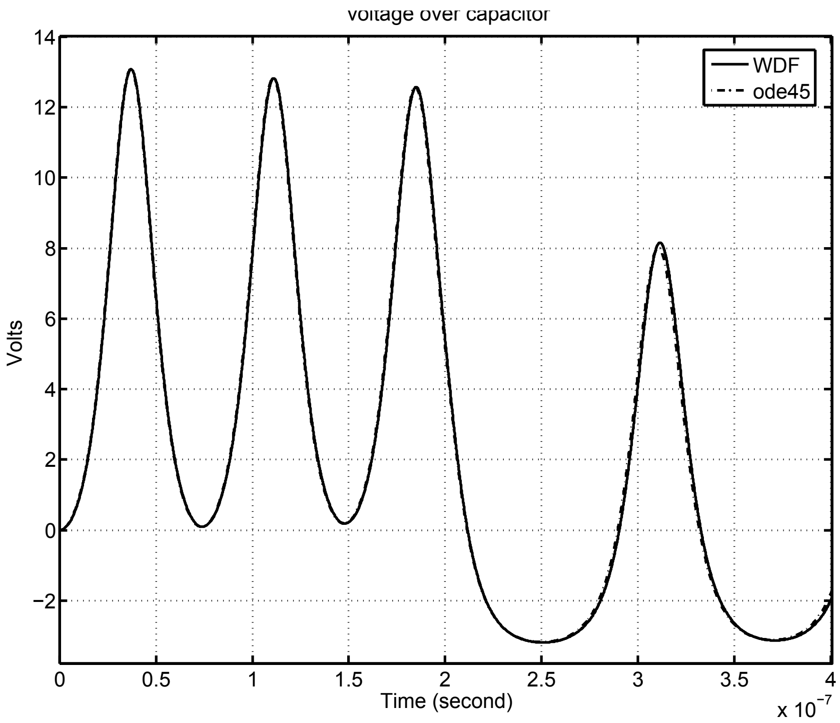


Figure 3.4: Voltage of Nonlinear capacitor by WDF and ODE45

3.2.2 Cutting DFDL with Jacobi Method

DFDL can be described as an implicit function, it can be cut by an iterative formulation [13]. Using the same nonlinear RLC circuit in (Figure 3.1) as an example, WDF analysis using Jacobi-like method to cut the DFDL is shown in Figure 3.5.

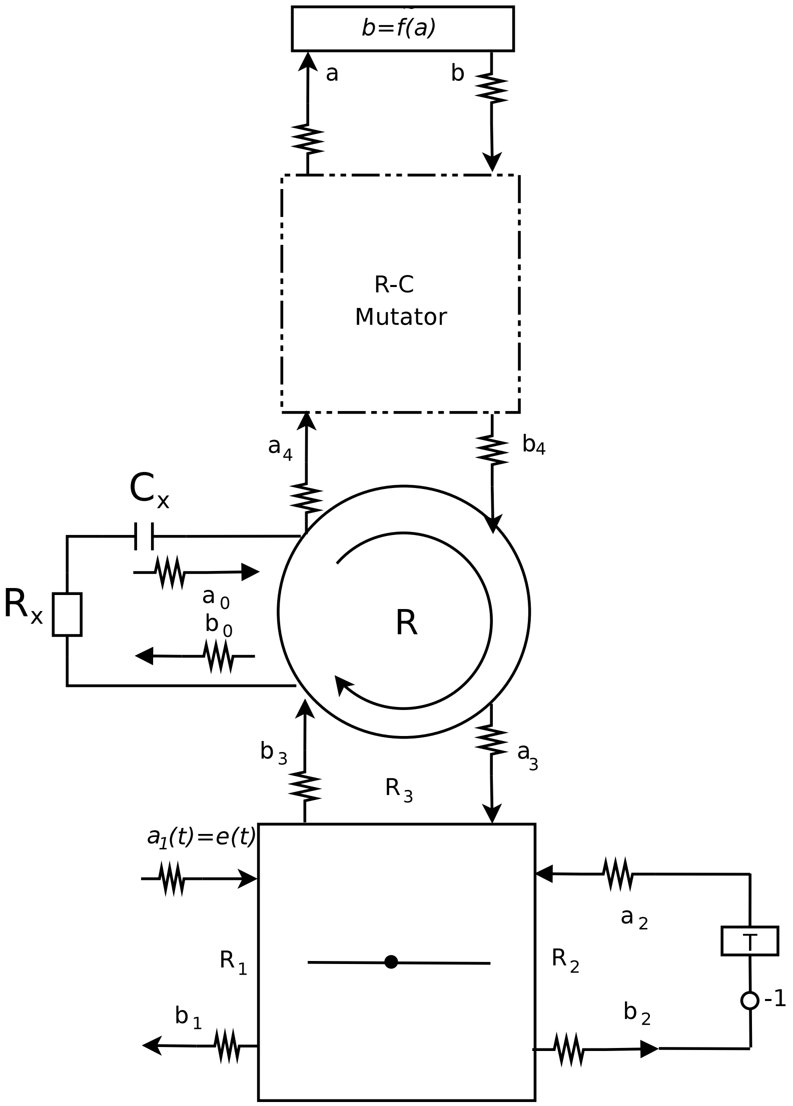


Figure 3.5: WDF implement of RLC circuit by Jacobi-like iteration

Let Port 3 of the series adaptor to be a normal port, R_3 as port 3 resistance can be assigned any arbitrary resistance. Mutator reference capacitance is chosen to be $C_0/10$. Circulator port resistance is chosen to be R. Condition of WDF analysis shown as follows:

$$C = C_0/10 \qquad R_c = T/(2 \times C)$$

$$R = \sqrt{L/C_0} \qquad R_x = R/5 \qquad C_x = T/(2 \times (R_0 - R_x))$$

Scattering matrix of series adaptor shown below:

$$\begin{aligned}\begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} &= \begin{pmatrix} (1-\beta_1) & -\beta_1 & -\beta_1 \\ -\beta_2 & 1-\beta_2 & -\beta_2 \\ -\beta_3 & -\beta_3 & 1-\beta_3 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \\ \beta_1 &= \frac{2R_s}{R+R_s+R_l} \\ \beta_2 &= \frac{2R_l}{R+R_s+R_l} \\ \beta_3 &= \frac{2R}{R+R_s+R_l}\end{aligned}$$

A Jacobi-like iterative formulation is applied on the circulator at each time step, in other words, b_3 is calculated by iteration to achieve an accurate value. So at each time step, a linear R-C circuit and a circulator are inserted between the series adaptor and the mutator. The iteration can be treated as an imaginary time dimension which does not effect reality. In this imaginary time dimension, linear R-C circuit makes circulator realizable. When R-C circuit achieves steady state in the imaginary time dimension, b_3 and a_3 achieve real values at this time step. The Jacobi-like equation for circulator shown below:

$$a_0 = (1-\gamma)b_0 + \gamma a_0 \tag{3.1}$$

$$b_3 = b_0; \qquad a_4 = a_0$$

Simulation results shown in Figure 3.6. Both of them are implemented in Matlab, two results are compared with each other. One result is based on reflection-free port (matched port resistance), while another one is based on unmatched port resistance. From these results we can observe that two methods are perfect matched, in other words, Jacobi-like iteration can be used in WDF implementation for cutting DFDL.

Table 3.1 shows simulation time between two method. From the table we can observe that Simulation time based on Jacobi-like iteration is affected by value of

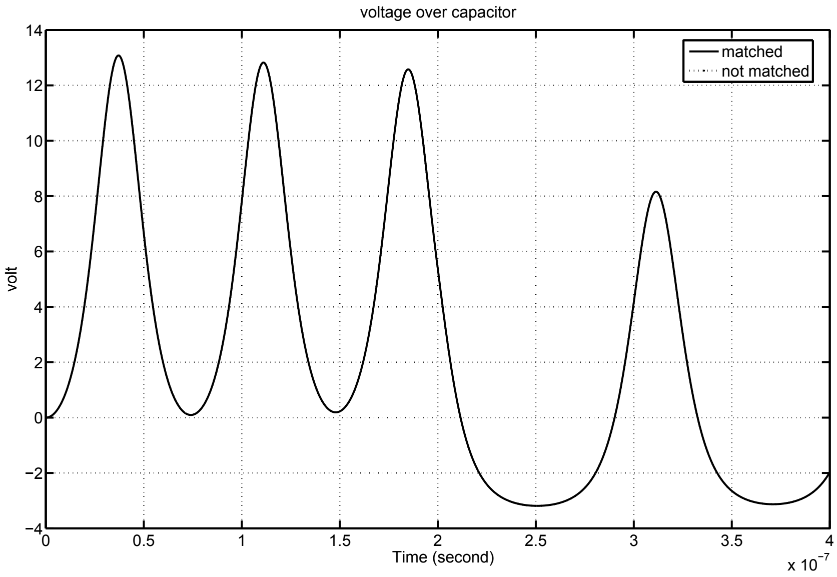


Figure 3.6: Voltage on Nonlinear Capacitor

γ. When $\gamma = 0.5$ is chosen, simulation time difference between matched port and Jacobi-like iteration is small. When $\gamma = 0.2$ and $\gamma = 0.8$ are chosen, the difference of simulation time are quite large.

Simulation Time and Jacobi Iteration Step Comparison of Matched and Unmatched port

Methods		Matched Port	Unmatched Port		
			$\gamma = 0.2$	$\gamma = 0.5$	$\gamma = 0.8$
Simulation Time (second)		10.525	72.245	12.878	72.864
Jacobi Iteration Steps	Min	—	27	4	32
	Max	—	46	9	54

Table 3.1: Simulation time and Jacobi-Iteration step comparison

3.3 WDF Analysis and Simulation for Transmission Line

3.3.1 Simulation of A Circuit Containing Two Ideal Voltage Sources

The Jacobi-like iteration can be used for a circuit which has more than one DFDL. If a circuit contains an ideal voltage source, DFDL is introduced into the circuit's WDF implementation. From the WDF implementation of ideal voltage source $b(t) = 2e - a(t)$, it clearly shows that reflected wave is simultaneously dependent on the incident wave.

A WDF implementation of a circuit with ideal voltage sources and nonlinear capacitor proves that Jacobi-like iteration can cut more than one DFDL. The circuit shown in (Figure 3.7).

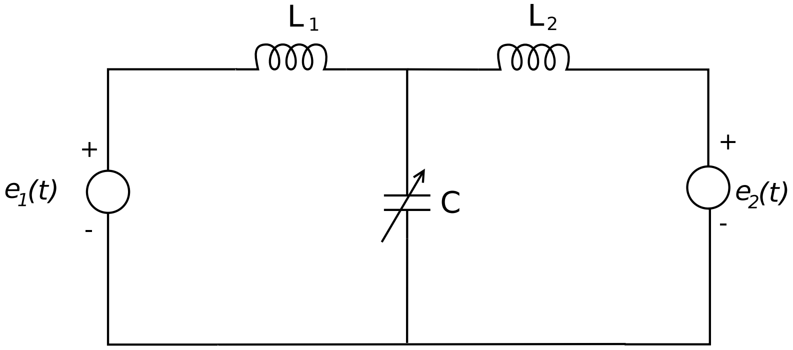


Figure 3.7: Circuit with ideal voltage source

This circuit has two ideal voltage sources and one nonlinear capacitor. Circuit condition shown as follow:

$$e_1(t) = e_2(t) = \begin{cases} E & t \leqslant 200ns \\ 0 & t > 200ns \end{cases}$$

$$L_1 = L_2 = 1.38\mu H$$

Nonlinear capacitor equation is as follow:

$$q = c_0 v_0 ln(1 + \frac{v}{v_0}), \qquad v > -v_0 \qquad c_0 = 224.9pF, \qquad c_0 = 3.73V$$

(Figure 3.8) shows that more than one DFDLs exist in WDF implementation.

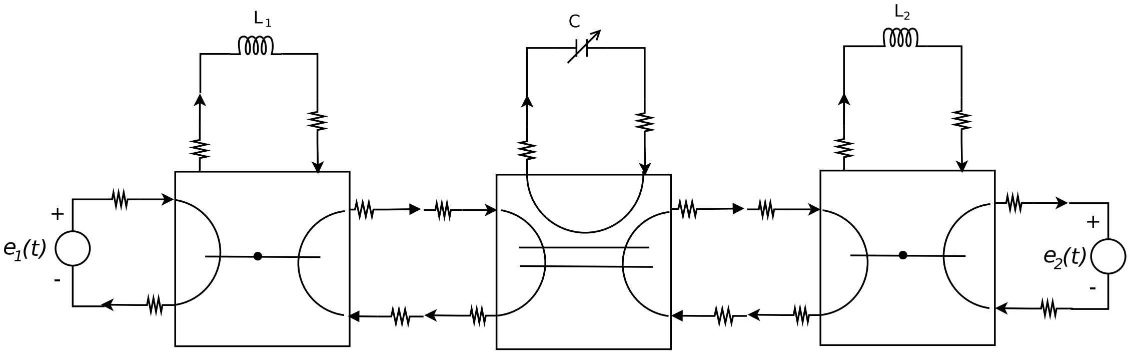


Figure 3.8: DFDLs in WDF implement of circuit

WDF Implementation

From Figure 3.8 we find DFDL occurs not only on the nonlinear port, but also it occurs on the interconnection ports between the two adaptors as well. In "Wave digital filters: Theory and practice" [1], to implement a fifth-order Cauer structure in WDF, every interconnection between two adaptors has to match the port resistance to achieve a reflection-free port. As the fifth-order Cauer structure only has linear elements, using reflection-free port is an easy method for realization. As for the circuit Figure 3.7, it is not easy to find reflection-free port.

With help from Jacobi-like iteration, it is possible to realize this nonlinear circuit. Combined with Jacobi-like iteration method and reflection-free port, there are several WDF implement configurations to cut DFDLs. as shown in Figure 3.9, Figure 3.10 and Figure 3.11.

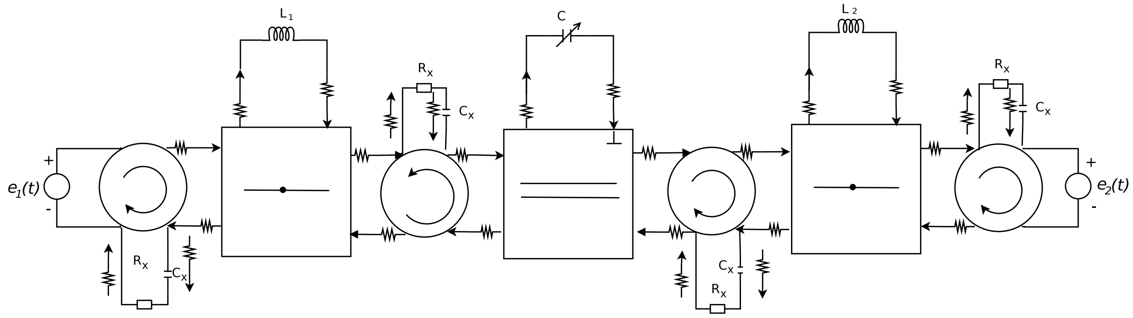


Figure 3.9: Configuration 1

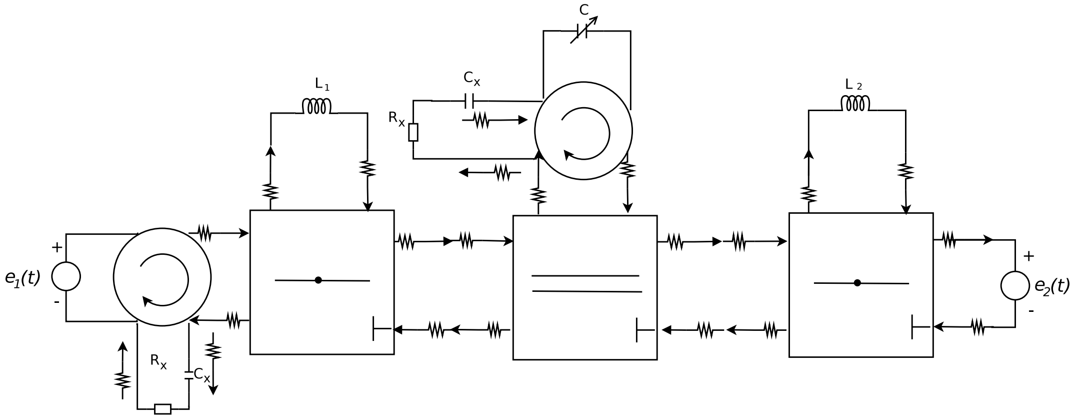


Figure 3.10: Configuration 2

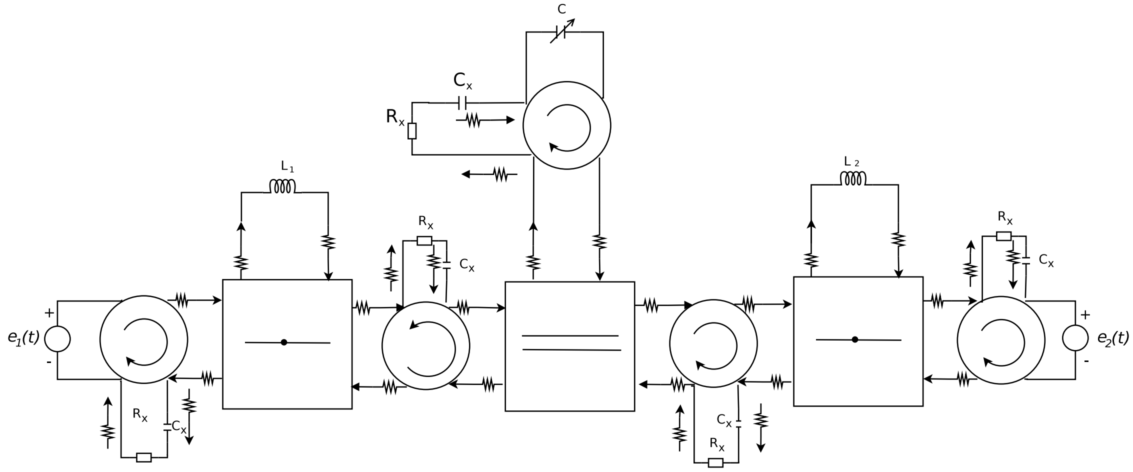


Figure 3.11: Configuration 3

In Configuration 1 (Figrue 3.9), the parallel adaptor port connected with nonlinear capacitor is chosen as a reflection-free port. Circulators are inserted between adaptors to cut DFDLs caused by adaptor interconnection. To achieve a symmetric WDF structure, circulators are inserted between ideal voltage sources and series adaptor port.

Configuration 2 is shown in Figure 3.10. Between the nonlinear capacitor and the parallel adaptor, a circulator is inserted to cut DFDL, which is cut by inserting a circulator between the left voltage source and the series adaptor. Every adaptor port which connects the next adaptor is chosen to be a reflection-free port, where port resistance is matched. The last series adaptor port connected with voltage source is a reflection-free port to cut DFDL.

The Configuration 3 is shown in Figure 3.11, where each DFDL is cut by inserting circulator.

Comparing these three configurations, the first configuration appears to be the best choice for simulation. In the second configuration, each adaptor port connect with another adaptor has to be a reflection-free port, in other words, scattering matrix of each adaptor are different and have to be calculated separately. In third configuration, there are five circulators. Actually, each circulator is a loop in simulation program, thus the efficiency of simulation is weaker than the first configuration

WDF analysis of configuration 1 is shown here, where the inserted network condition is:

$$R = \sqrt{L/C_0} \qquad G = 1/R \qquad R_x = R/2 \qquad C_x = T/(2 \times (R - R_x))$$

Mutator reference capacitance matched the parallel adaptor port resistance to get a reflection-free port:

$$C = \frac{T}{\sqrt{L/C0}} \qquad R_c = \frac{T}{2 \times C} = \frac{R}{2} \qquad G_c = \frac{1}{R_c}$$

Scattering matrix of series adaptor shown below, where $R_{l1} = R_{l2} = 2L/T$:

$$\begin{aligned}
\begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} &= \begin{pmatrix} 1-\beta_1 & -\beta_1 & -\beta_1 \\ -\beta_2 & 1-\beta_2 & -\beta_2 \\ -\beta_3 & -\beta_3 & 1-\beta_3 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \\
\beta_1 &= \frac{2R}{R+R+R_l} \\
\beta_2 &= \frac{2R_l}{R+R+R_l} \\
\beta_3 &= \frac{2R}{R+R+R_l}
\end{aligned} \tag{3.2}$$

Parallel adaptor scattering matrix is:

$$\begin{aligned}
\begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} &= \begin{pmatrix} \alpha_1-1 & 1 & 1-\alpha_1 \\ \alpha_1 & 0 & 1-\alpha_1 \\ \alpha_1 & 1 & -\alpha_1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \\
\alpha_1 &= \frac{2G}{2G+G_c}
\end{aligned}$$

Simulation Result

According to the definition of series adaptors, the voltage reference direction is not the same on every port. When a series adaptor connects to a parallel adaptor as seen in Figure 3.9, scattering matrix of series and parallel adaptor has to be calibrated for suiting the correct voltage reference direction.

For the example circuit, the first series adaptor and parallel adaptor need to be revised. After changing the scattering matrix of the parallel adaptor, the voltage reference direction change to the same direction as the next series adaptor, so that the last series adaptor does not need to be revised. Revised scattering matrices are shown below:

Scattering matrix of first series adaptor

$$\begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 1-\beta_1 & \beta_1 & \beta_1 \\ \beta_2 & 1-\beta_2 & -\beta_2 \\ \beta_3 & -\beta_3 & 1-\beta_3 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

Scattering matrix of parallel adaptor:

$$\begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} \alpha_1-1 & -1 & -\alpha_1 \\ -\alpha_1 & 0 & 1-\alpha_1 \\ -\alpha_1 & 1 & -\alpha_1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

WDF implementation of nonlinear capacitor is a function of the incident wave $b = \tilde{f}(a)$. The solution of this nonlinear function is found by linear interpolation in Section 3.2. This method is not a best choice for numeric simulation, as it is slow to get an accurate result. Newton method is a fast convergence method for numeric simulation, which can reduces simulation time and error for simulation.

From the nonlinear capacitor characteristic function $q = c_0v_0ln(1+\frac{v}{v_0})$, a voltage-charge function can be found:

$$q(v) = c_0v_0ln(1+\frac{v}{v_0}) \Rightarrow v(q) = v_0(e^{\frac{q}{c_0v_o}} - 1)$$

A function of $F(v, v(q)) = 0$ can be found. To find the root of this function, an error function $v - v(q) = 0$ has to be constructed. Using the R-C mutator, which is described in Chapter two, voltage and charge can be described in wave variables as follows:

$$a = v + \frac{1}{C}q \Rightarrow v = \frac{a+b}{2}$$
$$b = v - \frac{1}{C}q \Rightarrow q = \frac{a-b}{2}$$

Error function in WDF domain shown below:

$$err = a + b - 2v_0(e^{\frac{(a-b)\times C}{2c_0v_0}} - 1) = 0$$

To solve the error function, by Newton method [20] the iterative equation is as follows:

$$b = b_0 - \frac{a + b_0 - 2v_0 e^{C(a-b_0)/2C_0 v_0} + 2v_0}{1 - \dfrac{C}{C_0} \times e^{C(a-b_0)/2C_0 v_0}}$$

Where b_0 is the initial value for iteration, C is the mutator reference capacitance.

Simulation results are shown in (Figure 3.12, 3.13, 3.14). Voltage on capacitor is shown in Figure 3.12 which compares with circuit simulator Carrot [25]. Voltage on nonlinear capacitor shows behavior of a loss less circuit, an oscillation happens and no energy is lost. Simulation result shows perfect matching at a $200ns$ wide voltage impulse. At the edge of voltage impulse, current on capacitor clearly shows the response of linear inductor in the circuit.

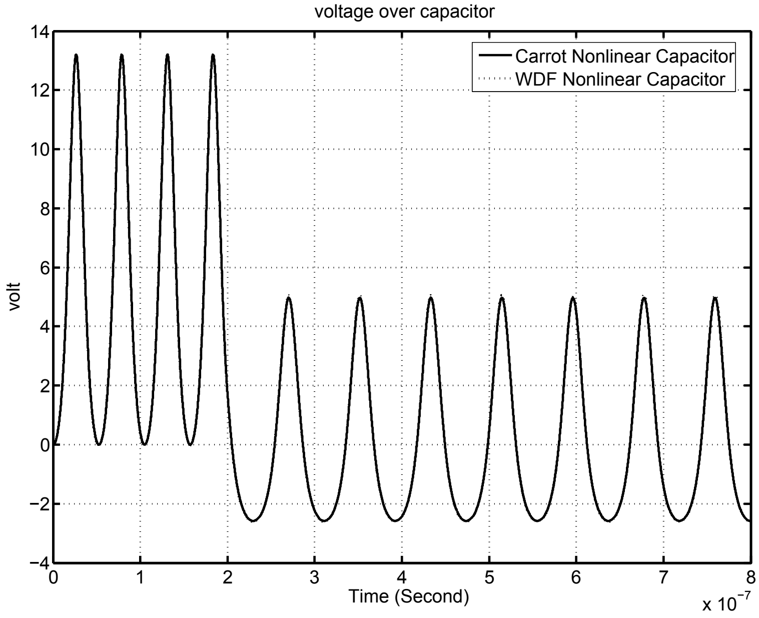


Figure 3.12: Voltage on Nonlinear capacitor

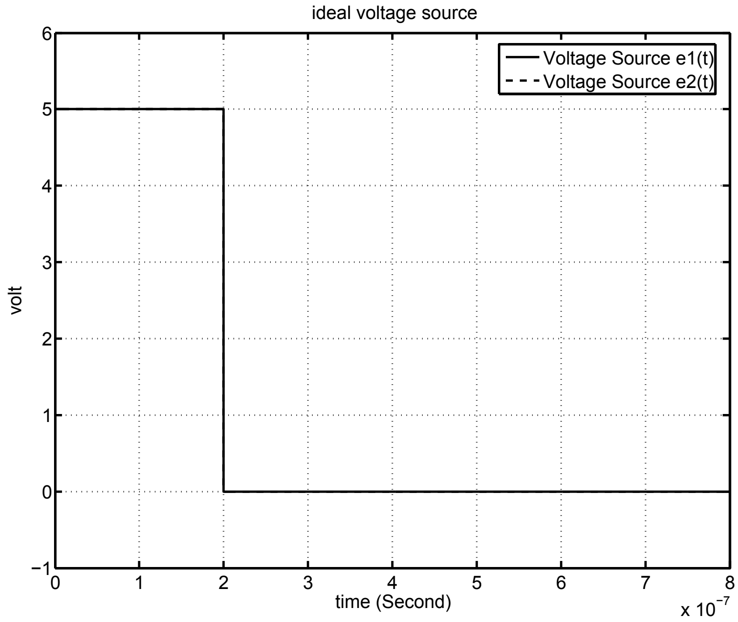


Figure 3.13: Voltage on ideal Voltage sources

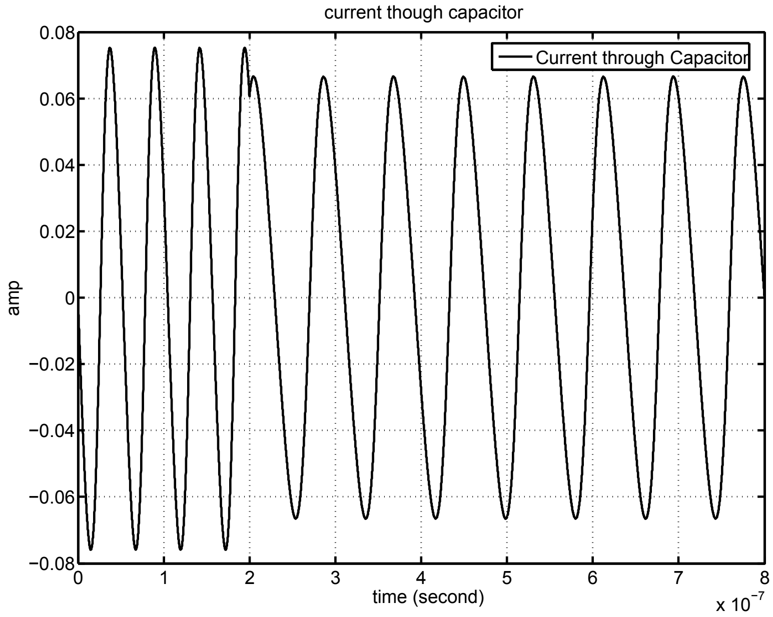


Figure 3.14: Current though Nonlinear capacitor

3.3.2 Simulation of Nonlinear Transmission Line

The circuit shown in (Figure 3.7) can be regarded as a part of a nonlinear transmission line (Figure 3.15) [12, 13]. The WDF implementation of the circuit (Figure 3.9) can be used for implementation of a nonlinear transmission line and it is suitable for parallel computing. The transmission line parameters are the following:

$$e_1(t) = e_2(t) = \begin{cases} E & t \leqslant 200ns \\ 0 & t > 200ns \end{cases}$$

$$L_1 = L_2 = L_n = 1.38\mu H$$

$$q = c_0 v_0 ln(1 + \frac{v}{v_0}), \quad v > -v_0,\, c_0 = 224.9pf,\, v_0 = 3.73v,\, for\, all\, capacitors$$

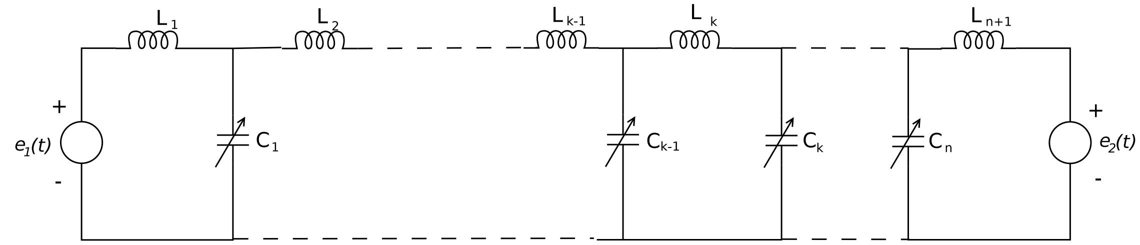


Figure 3.15: Nonlinear transmission line

WDF Implementation

WDF implementation of nonlinear transmission line is shown in Figure 3.16. The whole transmission line can be divided in to small sections, which only include a linear inductor and a nonlinear capacitor. Inductors and capacitors are connected by a series and a parallel adaptor. Between adjacent sections, circulators are inserted. This WDF implementation can be regard as an extension of WDF configuration 1 (Figure 3.9) in section 3.2.2. For each section at certain time step, the circulator isolate the incident wave from adjacent sections, and this make it possible for each section to be calculated independently. Other configurations as shown in last section can be used for simulating this nonlinear transmission line, but they do not lend to a parallel implementation or as efficiently as this configuration.

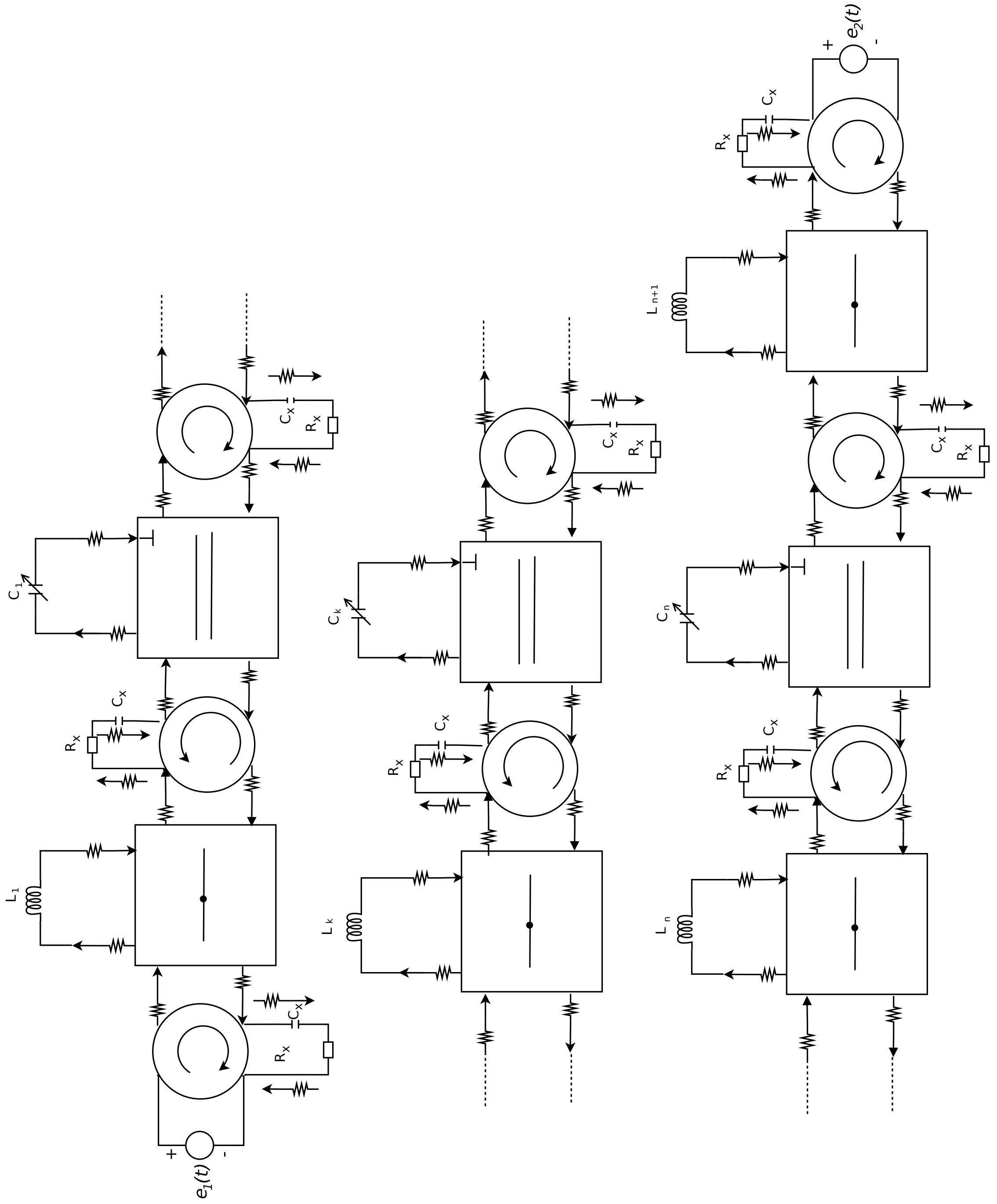


Figure 3.16: WDF implementation of Nonlinear transmission line

WDF simulation of transmission line implementation in a parallel algorithm shown in Algorithm 3.1.

Algorithm 3.1 Parallel computing

```
set $tor$
set $t \in (0, t_{end})$
repeat
    calculate each section separately
    Initialize guess for Jacobi-like
    repeat
        $a_0(n) = \gamma a_0(n-1) + (1-\gamma)[S_{22}a_0(n-1) + S_{21}a(t)]$
        $error = |a_0(n) - a_0(n-1)|$
        exchange adjacent section value
        calculate each section separately
    until $error < tor$
until $t = t_{end}$
```

Simulation Results

WDF simulation is implemented in a C program and the simulation results are compared with Carrot, a circuit simulator [25]. The C program is a series code program running on single processor, but the code and subroutines are designed in a pattern for parallel computing: calculation on different transmission line sections are individual, data exchange only happens when each section has completed its calculation.

The simulated transmission line has 99 nonlinear capacitors, voltage impulse from both sides pass through transmission line and show solitons traveling and colliding on this transmission line. Table 3.2 shows the simulation compare between Carrot and WDF method.

Simulation Time and Comparison of Carrot and WDF			
Methods		Carrot	WDF $\gamma = 0.5$
Time Step (seconds)		1×10^{-10}	1×10^{-10}
Time Interval		$2.5\mu s$	$2.5\mu s$
Simulation Time (second)		30.30	47.94
Jacobi Iteration Steps	Min	—	4
	Max	—	6

Table 3.2: Simulation time comparison of two method

Figure 3.17 shows comparison of voltage on first capacitor between the two simulations. Figure 3.18 shows the voltage of twentieth nonlinear capacitor. The two simulators get same voltage waveform. On this figure the soliton that travels through the transmission line can be observed. Comparing the voltage on different nonlinear capacitors, WDF simulation match the result of Carrot.

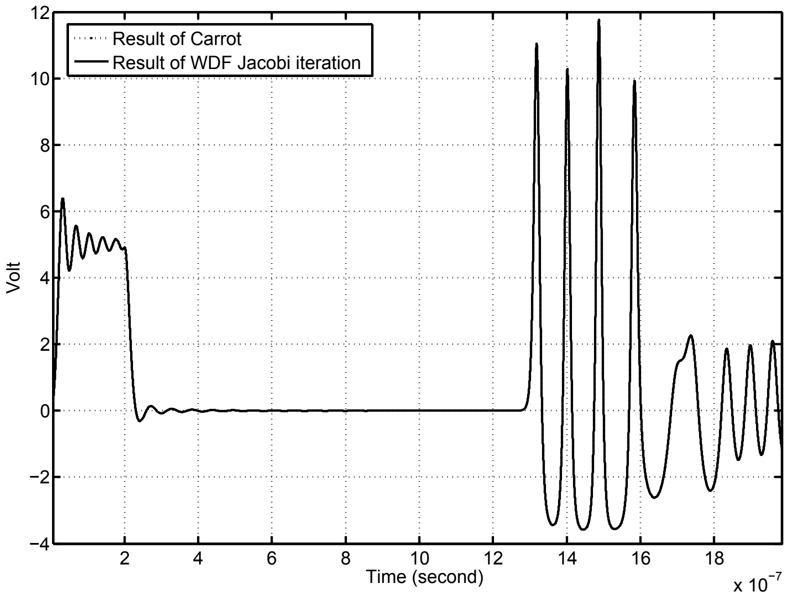


Figure 3.17: Voltage on first capacitor

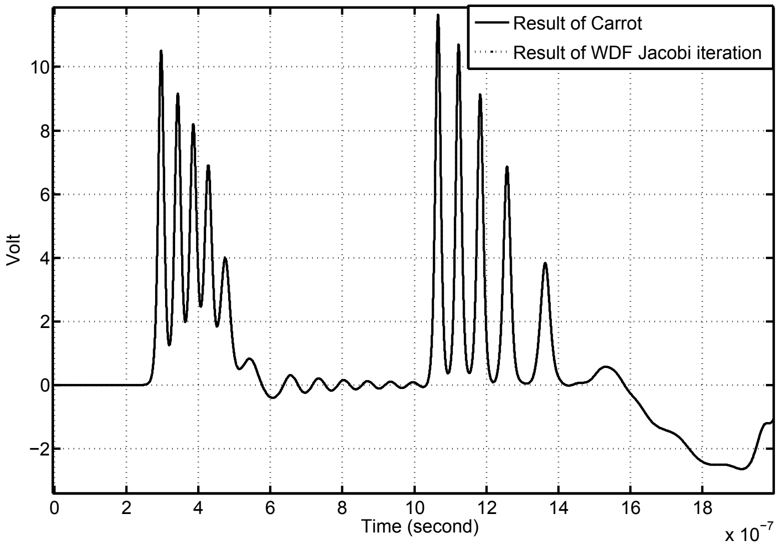


Figure 3.18: Voltage on 20th capacitor

To observe the soliton traveling and colliding on the transmission line, Figure 3.19 shows the voltage of 99 capacitors on one plot. To distinguish each voltage waveform, each capacitor's voltage is added additional voltage

$$v_n = v + n \times 5, \quad n \in (0, 1, \ldots 98)$$

From Figure 3.19, two rectangular impulses of ideal voltage sources have been separated into solitons of different amplitude, and travel with different velocities, finally solitons collide in the middle of transmission line.

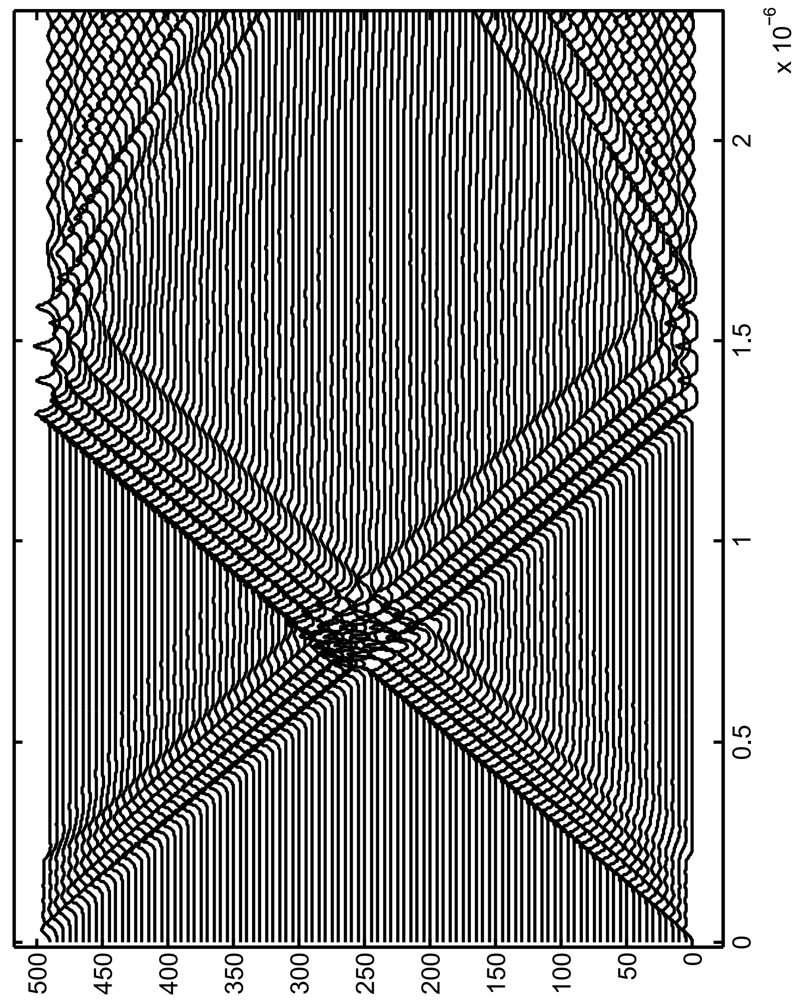


Figure 3.19: Collision of solitons

Effect of γ on the convergence rate

The Jacobi method convergence rate is affected by parameter γ. Choosing an appropriate value for γ would increase simulation speed dramatically. The number of Jacobi iteration step with different γ values is shown in Table 3.3. From this table $\gamma = 0.5$ is the best choice for simulation for this circuit. Figure 3.20 shows change of different γ value in simulation time interval.

value of γ	Iteration steps
0.1	80
0.2	35
0.3	20
0.4	12
0.5	5
0.6	11
0.7	18
0.8	29
0.9	60

Table 3.3: Jacobi iteration steps comparison for γ

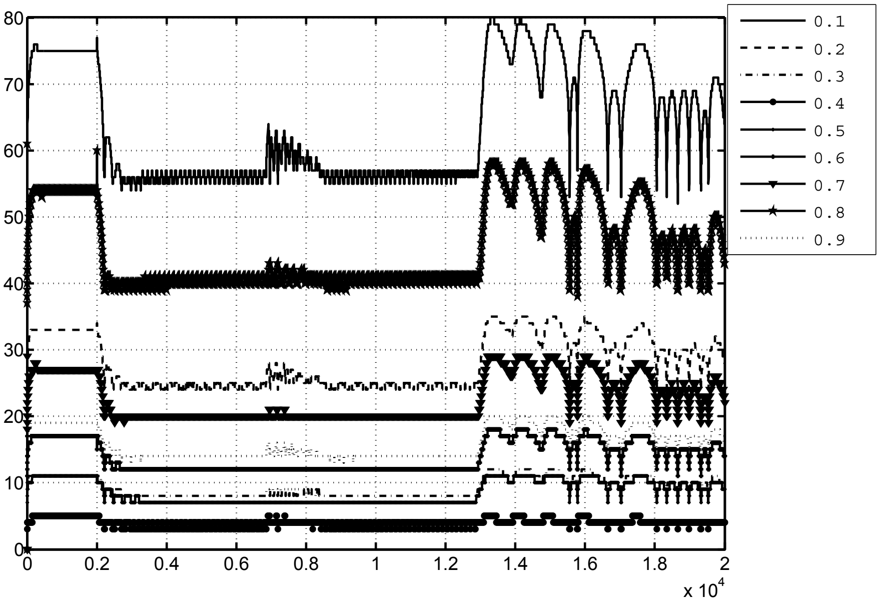


Figure 3.20: Jacobi iteration steps for different values of γ

3.4 WDF Simulation Based on Combined Adaptor

A parallel WDF implementation of the nonlinear transmission line, divides the transmission line into independent sections which contain one series and one parallel adaptor. For each section, the simulation program has to calculate two adaptors in turn, which means two matrix operations has to be implemented. Combining two different adaptors into one adaptor to reduce matrix operations can be attempted to improve efficiency of the simulation program.

3.4.1 Combined Adaptor and WDF Implementation

The divided section is shown in Figure 3.21.

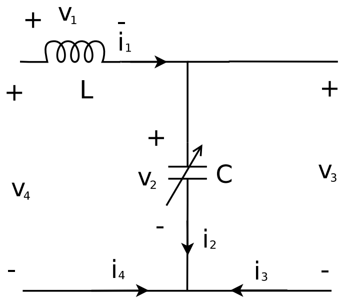


Figure 3.21: Divided section for parallel computing

From the loop set and the cut set, KCL and KVL equation can be found:

$$v_3 - v_2 = 0$$
$$v_4 - v_1 - v_2 = 0$$
$$i_1 + i_4 = 0$$
$$i_2 + i_3 + i_4 = 0$$

Constructing the scattering matrix by Equation 2.19, M_V, M_I and G are as follows:

$$M_V = \begin{bmatrix} 0 & -1 & 1 & 0 \\ -1 & -1 & 0 & 1 \end{bmatrix}$$

$$M_I = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

$$\underline{G} = diag\left(\frac{1}{R_l}, \frac{1}{R_c}, \frac{1}{R}, \frac{1}{R}\right)$$

Let the port connected with the nonlinear capacitor to be reflection-free port, port resistance $R_c = ((R_l + R) * R)/(R_l + 2R)$. The WDF implementation of the adaptor shown in Figure 3.22. The transmission line WDF implementation shown in (Figure 3.23)

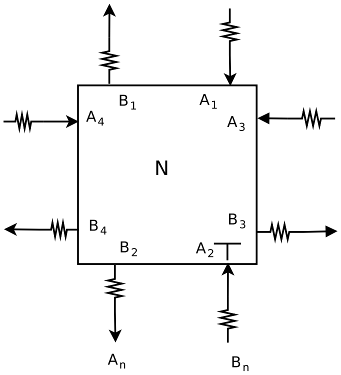


Figure 3.22: WDF implementation of adaptor

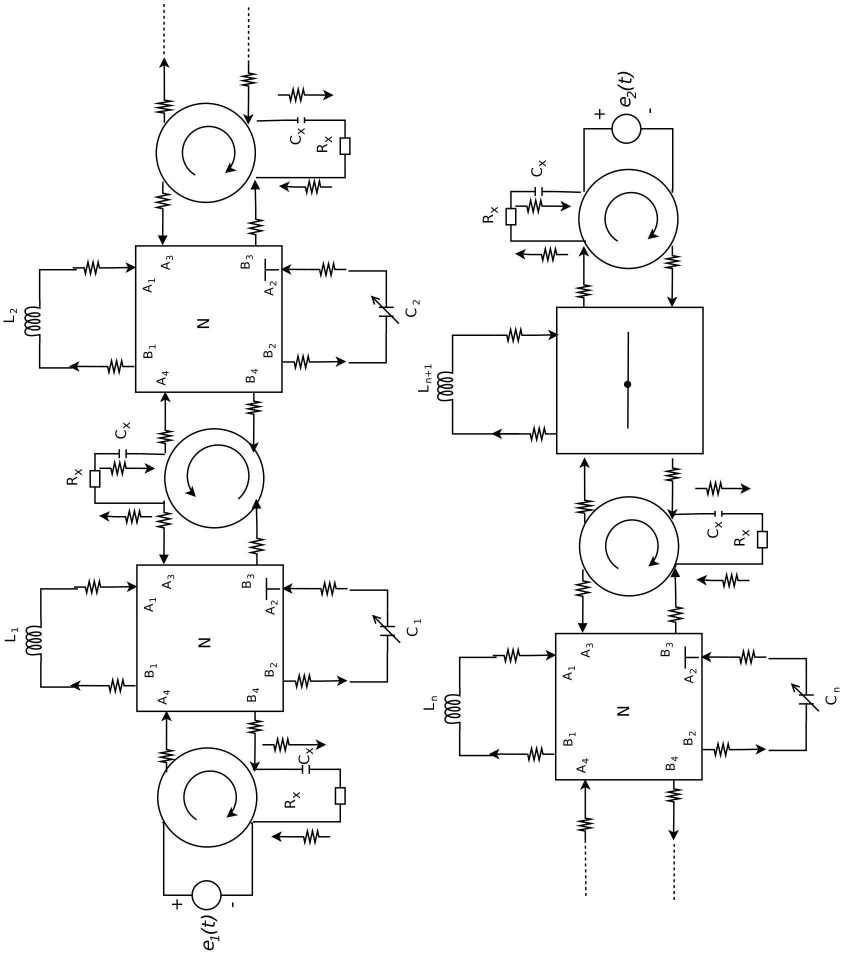


Figure 3.23: WDF implementation of rransmission line

3.4.2 Simulation Results of Combined Adaptor

The WDF implementation using the combined adaptor was implemented in a C program and simulation results compared with Carrot. The transmission line for simulation has 99 nonlinear capacitors. Using same error tolerance criterion as Section 3.3, the combined adaptor achieves the same results as expected. The C program is implemented in same pattern as last section: series code with parallel computing pattern. When $\gamma = 0.5$ the maximum number of iteration steps is 45.

Simulation time are compared in Table 3.4. WDF and WDF with Combined Adaptor both use $\gamma = 0.5$. Although Jacobi iteration step of using combined adaptor is more than WDF, but simulation time is not increased much. The main reason is that combined adaptor method simplified scattering matrix, therefore computing time of matrix calculation is reduced.

Simulation Time and Comparison of
Carrot, WDF and WDF with Combined Adaptor

Methods		Carrot	WDF	Combined Adaptor
Time Step (Seconds)		1×10^{-10}	1×10^{-10}	1×10^{-10}
Time Interval		$2.5\mu s$	$2.5\mu s$	$2.5\mu s$
Simulation Time (seconds)		30.30	47.94	79.37
Jacobi Iteration Steps	Min	—	4	37
	Max	—	6	45

Table 3.4: Simulation time comparison of three method

Figure 3.24 shows collision and traveling of solitons. Each capacitor voltage is treated same as in section 3.3.2. As we expected, using combined adaptor we get same result as using basic WDF adaptor.

Figure 3.25 and Figure 3.26 show voltage waveforms on different capacitors. Figure 3.25 shows voltage waveform of first capacitor in transmission line, results are matched. Figure 3.26 shows voltage wave form of fiftieth capacitor in transmission

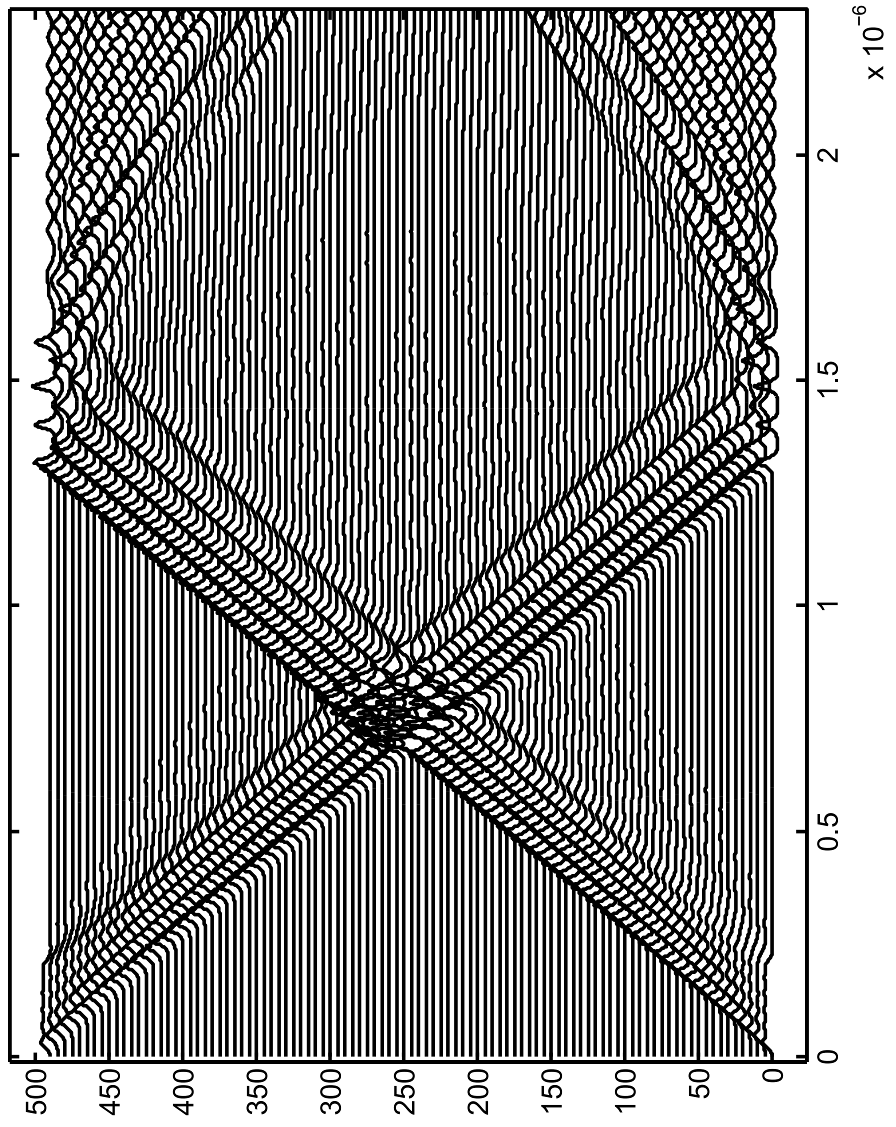


Figure 3.24: Collision of solitons with combined adaptor line. In this result the split solitons can be observed clearly.

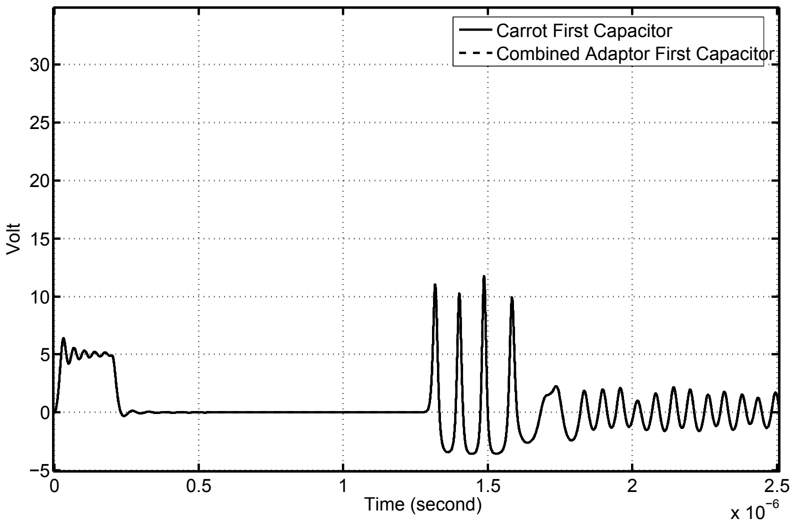


Figure 3.25: Voltage on first capacitor

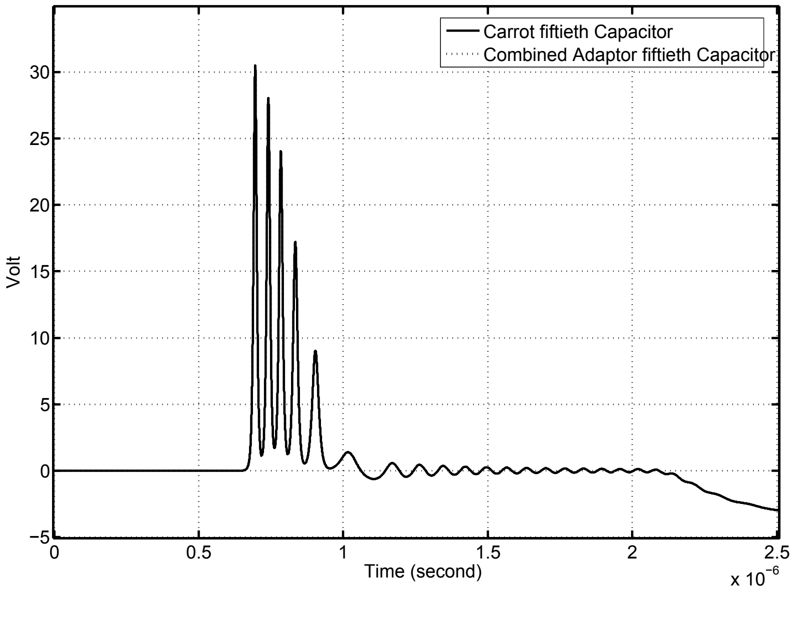


Figure 3.26: Voltage on fiftieth capacitor

Chapter 4

Waveform Relaxation Simulation

4.1 Introduction

With addition of MPI, the Jacobi-like parallel algorithm for the nonlinear transmission line can be implemented on a parallel computing environment, such as a computer cluster or a multi-core processor computer. Performance of a parallel computing program is not only dependent on well-organized program code, but also on an efficient parallel algorithm.

Comparing with a series program, the execution time of a parallel program is composed of several parts shown as follows [23, 24]:

- Computing time: the CPU time spent on executing processor instructions, include running time of program instructions code (user time) and necessary system kernel code running time (system time). Usually system time is negligible.
- Communication time: CPU time spent on communication between processors
- Synchronization time: time spent on synchronizing different processors.
- Processor idle time: a processor waits for other processor's message during blocking communication.

Define a series application program execution time as T_s, and the execution time of its parallel counterpart as T_p. Therefore the acceleration rate can be defined by

$$S_p = \frac{T_s}{T_p}$$

If there are P processes running on P processors, the efficiency of a parallel program is defined by

$$E_p = \frac{S_p}{P}$$

Parallel execution time can be expressed as below:

$$T_p = C_i + D_i \ , \quad i = 1, 2, \ldots P$$

where C_i is the execution time for i processor, D_i is the idle time of i processor, P is the total processors number of parallel program. C_i is composed by

$$C_i = L_i + O_i \ , \quad i = 1, 2, \ldots P$$

where L_i is the computing time of i processor, and O_i is the communication time and synchronization time of i processor. Defined $C_T = \sum_{i=1}^{P}(L_i + O_i)$, $D_T = \sum_{i=1}^{P} D_i$, then $C_T + D_T = P \times T_p$. Thus efficiency is presented as follows:

$$E_p = \frac{S_p}{P} = \frac{T_s}{C_T} \times \frac{C_T}{C_T + D_T}$$

When the parallel program is well load balanced, which means computing load is well-distributed on each processor, the D_T is small comparing with C_T. To improve parallel efficiency, parallel program has to reduce communication overhead.

O_i is composed by two parts, first part is the overhead of MPI communication function; second part is the data transfer time which can be approximated as $1/bandwith$. So if communication frequency can be reduced, the efficiency E_p will increase.

A simple parallel pseudo code for the Jacobi-like algorithm is shown below:

From the algorithm, we can find that data has to exchange in every Jacobi-like iteration. Thus waveform relaxation can help to reduce the data exchange

Algorithm 4.1 Pseudo code for Jacobi-like algorithm

```
$t \leftarrow 0$
Begin of parallel computing
initialize each processor
*For each processor execute codes below*
**repeat**
    $t = t + T_{step}$
    Calculate section wave variables
    $n \leftarrow 0$
    **repeat**
        $n = n + 1$
        $a_0(n) = \gamma a_0(n-1) + (1-\gamma)[S_{22}a_0(n-1) + S_{21}a(t)]$
        Send data to adjacent section
        Receive data from adjacent section
        Calculate section wave variables
    **until** $|a_0(n) - a_0(n-1)| \leq \varepsilon$
    Wait for other sections converge
**until** $t = t_{end}$
End of parallel computing
```

frequency of Jacobi-like iteration. Applying waveform relaxation on the Jacobi iteration, each Jacobi iterative step calculates a time interval, which significantly increases the proportion of L_i in C_i. Thus waveform relaxation can be applied on the parallel algorithm to improve E_p of the parallel program.

4.2 Implementation of Waveform Relaxation

WDF implementation of waveform relaxation has same WDF structure as Jacobi-like iteration, which is shown in Figure 3.16. The algorithm used in this simulation is similar to Jacobi iteration, but for each Jacobi iteration, wave variables vector of a time interval will be computed (instead of wave variables for one time step). Equation (2.23) can be written as follows:

$$\overline{a_0(n)_{wr+1}} = \gamma\overline{a_0(n-1)_{wr}} + (1-\gamma)[[S_{22}]\overline{a_0(n-1)_{wr}} + [S_{21}]\overline{a_{wr}}]$$

Where, wr denotes the iteration steps of waveform relaxation, and $\overline{a_0}$ denotes a vector containing wave variables of a time interval. The Algorithm for waveform relaxation is shown in Algorithm (4.2).

Algorithm 4.2 WR on Jacobi iteration

```
$n \leftarrow 0$
guess waveform $a_0(t), t \in [0, T_w]$
repeat
  $n = n + 1$
  for each $t$ in $[0, T_w]$ do
    $a_0^n(t) = \gamma a_0^{n-1}(t) + (1-\gamma)[S_{22}a_0^{n-1}(t) + S_{21}a(t)]$
  end for
  exchange results with adjacent sections
until $|a_0^n - a_0^{n-1}(t)| \leq \varepsilon$
```

For a time interval containing many time steps, waveform relaxation may cause Jacobi iteration to converge slowly. A technique can be used to reduce this disadvantage, as the time interval of waveform relaxation can be broken into “windows” [21], $[0, T_1], [T_1, T_2], \ldots [T_n, T_{n+1}]$, so that for each window wave relaxation converge faster. Waveform relaxation is applied to the first window, $[0, T_1]$ and the values of the wave variables at T_1 are then used as the initial conditions for the analysis of the second window. This procedure is repeated until all windows have been simulated. This approach will be denoted as “windows” technique.

4.3 Simulation Results

All simulation were performed on a computer with dual-core processor, the code is single-thread and was writren with C. The Waveform relaxation results show similar behavior as other simulations. Solitons travel and collide on the transmission line. However error compared with the Carrot increase with the time. Tolerance value of Jacobi-like iteration is chosen as 10^{-6}.

When WR is used to calculate a whole simulation time interval, results converge slowly. When the maximum number of iteration steps set to 100 times, result does not converge and is quite different compared with result of Carrot. Figure 4.1 shows result after 100 iteration steps. Only in the first $1\mu s$ time interval WR has a similar voltage waveform as Carrot.

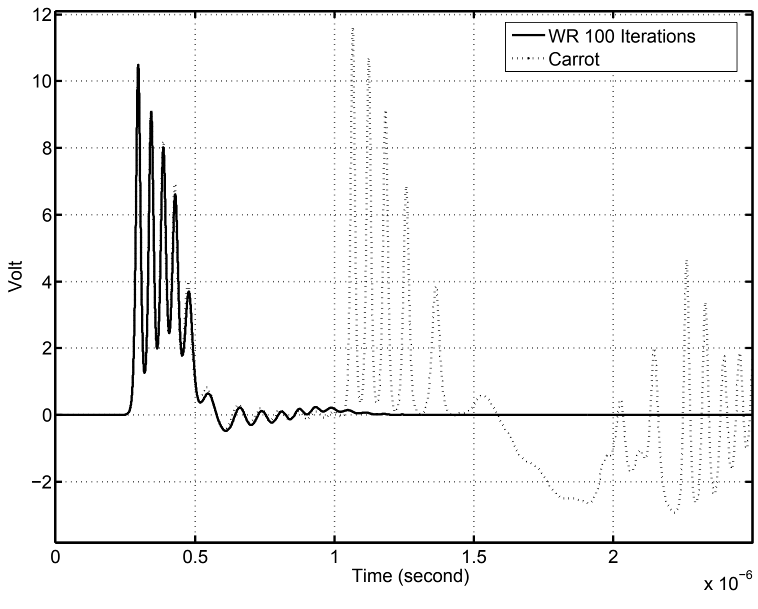


Figure 4.1: Voltage on twentieth capacitors

If the maximum number of iterations is increased to 300, WR results are closer to Carrot. Figure 4.3 shows that the voltage waveform obtained with WR is similar to the one obtained with Carrot, but after $1\mu s$ error between two results is large

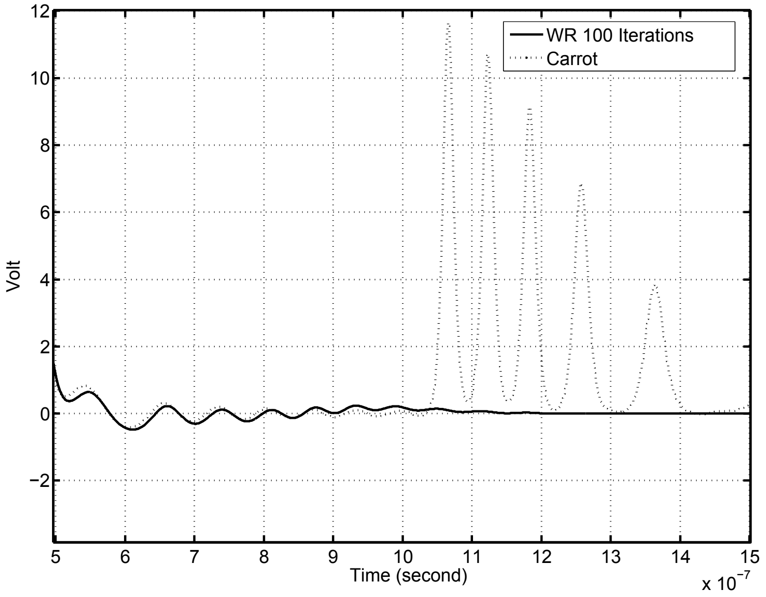


Figure 4.2: Voltage on twentieth capacitors

(Figure 4.4). Solitons of the WR result can not reach the same amplitude of result of Carrot. The detail also shows that the WR simulation result has a minor phase shift.

When the maximum number of iterations is increased to 600, solitons of WR result reach the same amplitude of Carrot's result. When this number is further increased to 1000, the result is almost same with 600's. But even with 1000 iterations, Figure 4.5 and Figure 4.6 show some discrepancy with Carrot.

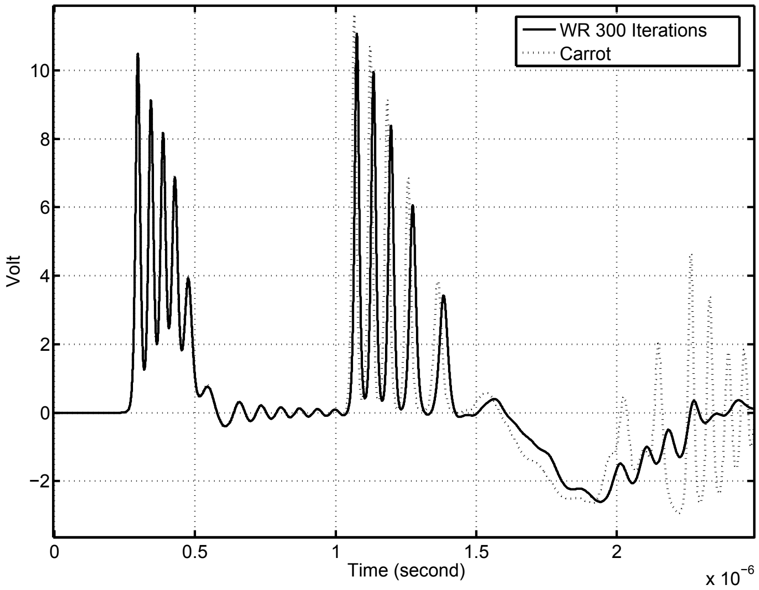


Figure 4.3: Voltage on twentieth capacitors

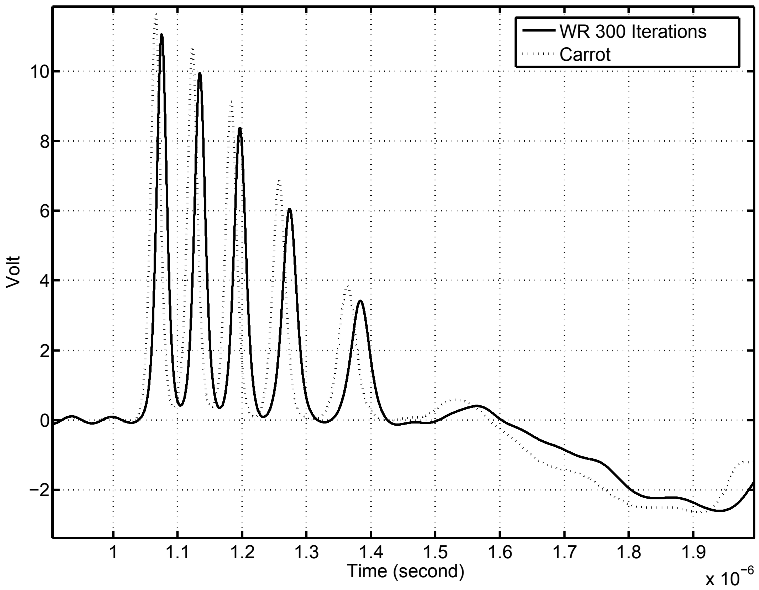


Figure 4.4: Voltage error on twentieth capacitors

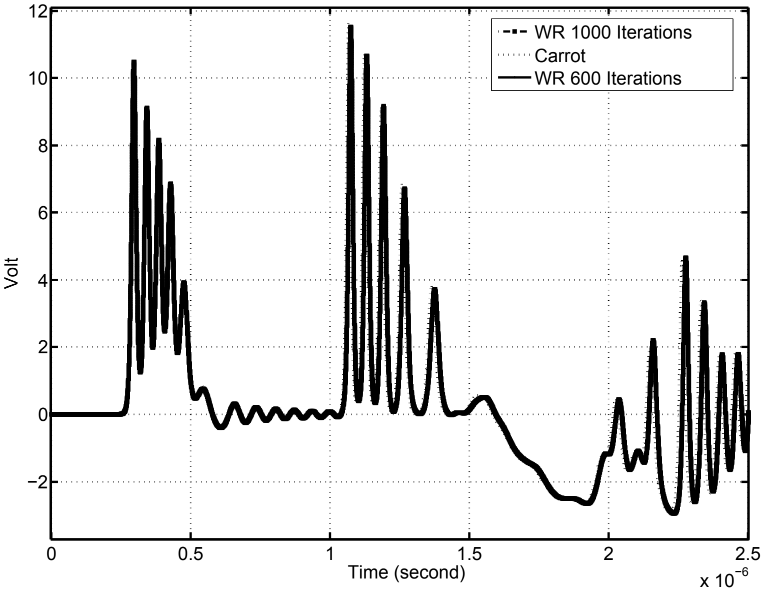


Figure 4.5: Voltage on twentieth capacitors

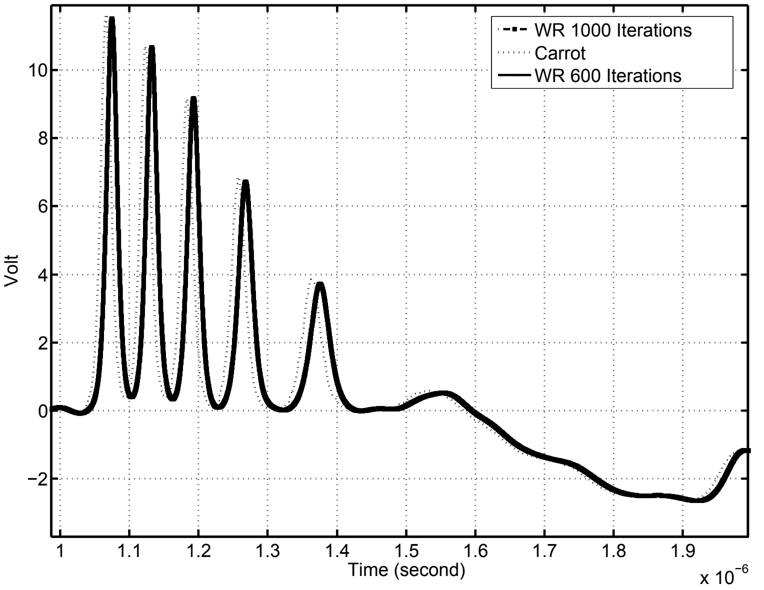


Figure 4.6: Voltage error on twentieth capacitors

When “windows” technique is used, the efficiency of simulation is improved, but improvement of accuracy is minor. Voltage on different capacitors are compared with results of Carrot. In Figures 4.7, 4.8, 4.9, detailed analyses of the voltage on first capacitor are shown. In a time interval of $2.5\mu s$, three results of different methods compared with each other: Carrot, waveform relaxation and waveform relaxation with a twenty-time-steps window. From the zoom of results of different time intervals, we can observe that at the interval between 0 to $1.3\mu s$ all three results match perfectly. After $1.3\mu s$ both waveform relaxations have approximately similar phase shift when compared to Carrot. The cause of this phase shift is not known at this time.

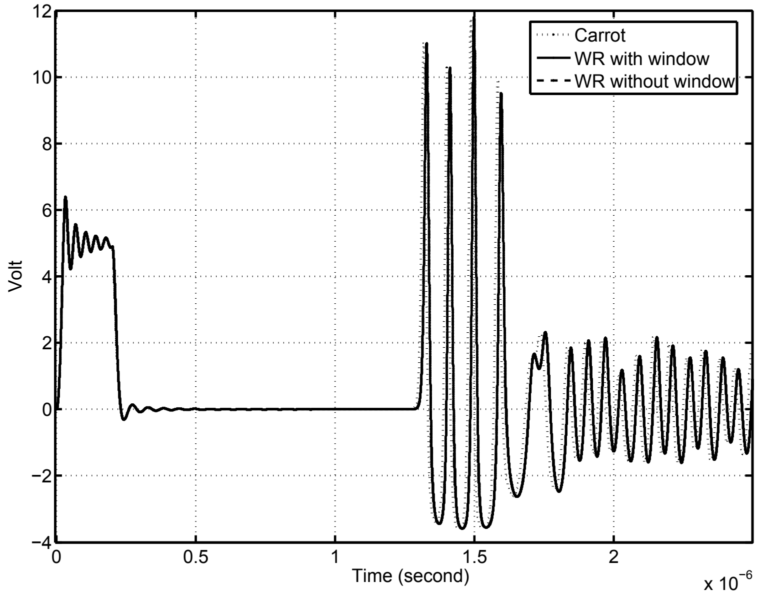


Figure 4.7: Voltage on first capacitor

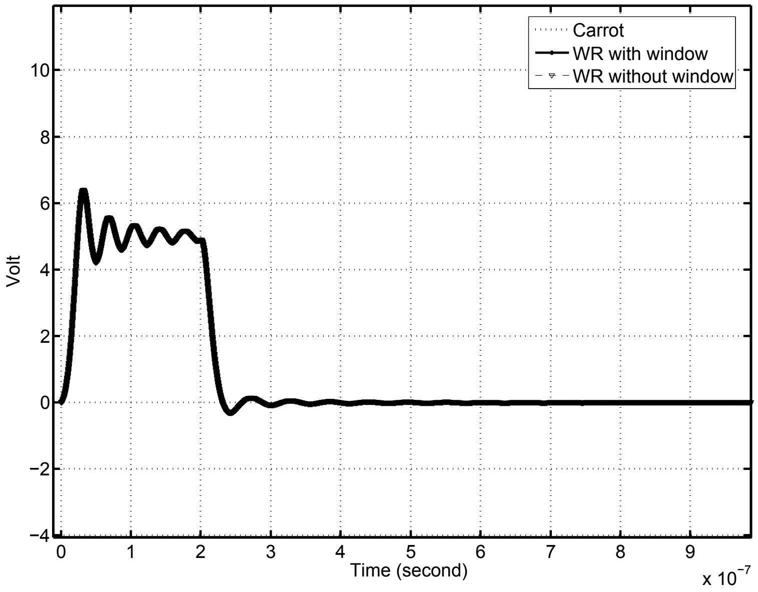


Figure 4.8: Zoom of 0 to 1μs

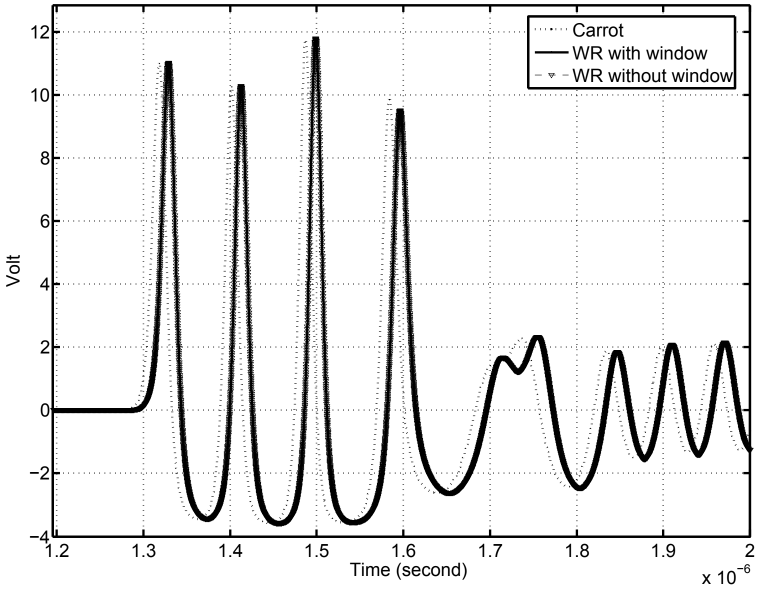


Figure 4.9: Zoom of 1μs to 2.5 μs

Simulation by waveform relaxation with windows is faster than waveform relaxation. In the results in above figures, using a maximum number of iterations of 1000 for waveform relaxation, with a tolerance 10^{-6}, not every time step can satisfy the converge condition. While for a small time interval of twenty time steps, the maximum iterations number is 85. Simulation times for the different methods is shown in Table 4.1.

Last column of the Table 4.1 is the program computing time for each iteration before program exchanges data. We can find that the WDF without WR has to exchange data only after $0.39\mu s$, but WR and WR with window technique use much more computing time before exchanging data. From Section 4.1 of this chapter, we know that reduced communication overhead can improve parallel program efficiency. Using WR with 20 samples window as an example, and assuming the parallel program only calculate for 20 time steps, the approximate improvement of efficiency is:

$$E_{pwr} = \frac{T_s}{9.747 + O_T + D_T}$$

Approximated acceleration rate for WDF without WR is:

$$E_{pwdf} = \frac{T_s}{20 \times (0.39 + O_T + D_T)} \Rightarrow$$

$$E_{pwdf} = \frac{T_s}{7.8 + 20 \times (O_T + D_T)}$$

We can find that C_T of WR is only $1.9\mu s$ slower than C_T of Jacobi-like iteration, but communication just happens once. Jacobi-like iteration has to call MPI communication function 19 times more than WR. Although this performance comparison is based on an approximate assumption, it shows the potential advantage of WR on parallel computing.

Simulation Time and Comparison of
Jacobi-like iteration and WR combined with Jacobi-like iteration

Methods	Time Interval	Simulation Time (s)	Samples per Window	Maximum Iterations	Computing Time for 1 section between data exchange (μs)
WR (100 iterations)	2.5μs	77.32	25000	100	4149.29
WR (300 iterations)	2.5μs	467.87	25000	300	4149.29
WR (600 iterations)	2.5μs	1132.56	25000	600	4149.29
WR (1000 iterations)	2.5μs	2050.08	25000	1000	4149.29
WR window	2.5μs	165.72	20	85	9.747
WR window	2.5μs	332.63	50	156	19.929
Jacobi-like iteration	2.5μs	47.94	1	6	0.39

Table 4.1: Simulation time comparison of Jacobi-like iteration and WR combined with Jacobi-like iteration

Chapter 5

Conclusions and Future Work

5.1 Summary of Research Works

WDF-based transient simulations of a nonlinear transmission line have been studied in this thesis. Several simulation methods were applied to this circuit. A Jacobi iteration suited for parallel computing has been studied and verified. For the first time waveform relaxation method is applied combined with WDF simulation. The main conclusions are as follows:

- The method presented in [13] has been verified.

- The optimum γ parameter is unpredictable and this limits the application of this method.

- If circuit topology is not changed, different WDF adaptor implementations for interconnection do not affect simulation results, but have an important effect on convergence rate.

- Newton method is suited for nonlinear WDF element simulation.

- Waveform relaxation can be used along withe WDF implementation.

- Using "window" technique for WR improves WR simulation efficiency.

- WR use more CPU time than time-marching Jacobi-like method before data exchange. This increases acceleration rate of a parallel program. A large acceleration rate means that a more efficient parallel program can be implemented.

5.2 Future Work

WR has been studied in this thesis. It can improve parallel computing program efficiency, but the simulation results show that accuracy is need to be improved. A thorough study on waveform relaxation is left for future work.

In Chapter Four, a window technique is applied on WR, but every window is independent from other windows. An improved window technique for WR can be studied in future work. (Figure 5.1) shows the basic concept of this method, where windows overlap on previous window; thus the next windows will get initial guess value when first time step converge in previous window. With this method, WR should obtain a more accurate initial guess value to improve the overall simulation accuracy.

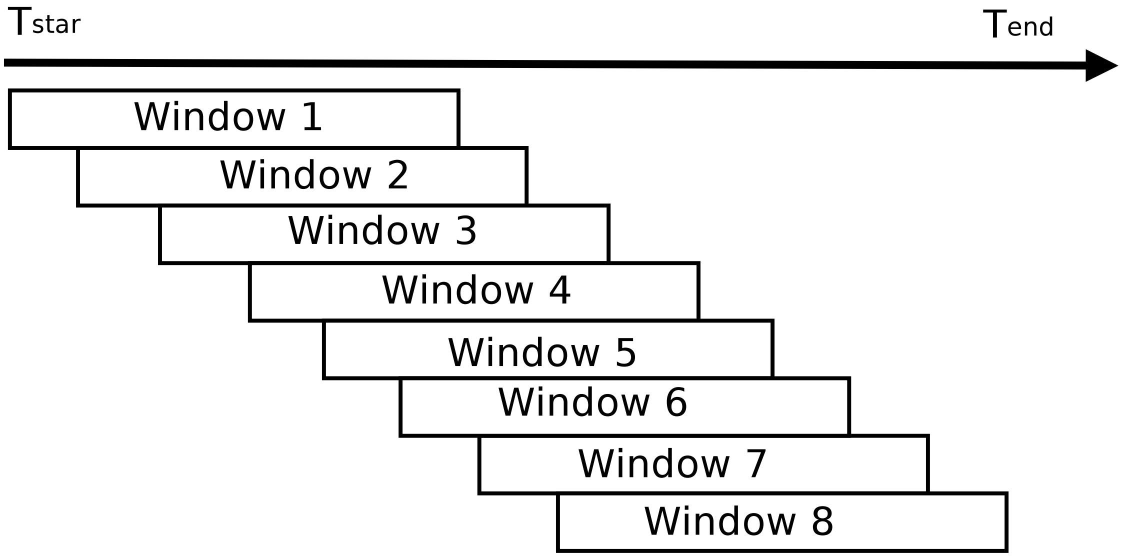


Figure 5.1: Improved window technique

Appendices

Appendix A

Numerical Algorithm

Jacobi iterative Method

The Jacobi iterative method is an algorithm for solving a system of linear equations in the form of $Ax = b$, where

$$A = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ A_{21} & A_{22} & \dots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \dots & A_{nn} \end{bmatrix}, \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \qquad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Split matrix A into its diagonal and off-diagonal part, where D denotes diagonal matrix whose diagonal is same as S , L and U are lower and upper triangular parts of A. Equation is shown as follow:

$$A = \begin{bmatrix} A_{11} & 0 & \dots & 0 \\ 0 & A_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & A_{nn} \end{bmatrix} - \begin{bmatrix} 0 & \dots & \dots & 0 \\ -A_{21} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ -A_{n1} & \dots & -A_{n,n-1} & 0 \end{bmatrix} - \begin{bmatrix} 0 & -A_{12} & \dots & -A_{1n} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & -A_{n-1,n} \\ 0 & \dots & \dots & 0 \end{bmatrix}$$

$$A = D - L - U$$

$Ax = b$ is transferred to form of $x = D^{-1}(L+U)x + D^{-1}b$. Jacobi iterative method solve the i`th` equation in $Ax = b$ for x_i is

$$x_i = \sum_{j=1, j\neq i}^{n} (-\frac{a_{ij}x_j}{a_{ii}}) + \frac{b_i}{a_{ii}}, \qquad \texttt{for } i = 1,2,3\ldots n$$

For x^k, it is generated of x^{k-1}, `where` $k \geq 1$ by equation shown as follows

$$x^k = D^{-1}(L+U)x^{k-1} + D^{-1}b$$

or for each x_i^k

$$x_i^k = \frac{\sum_{j=1,j\neq i}^{n}(-a_{ij}x_j^{k-1}) + b_i}{a_{ii}} \qquad \texttt{for } i = 1,2,3\ldots n$$

Jacobi iterative algorithm is shown as follow:

Algorithm A.1 Jacobi iterative algorithm

```
initial guess value $X0$
tolerance $tor$
max number of iteration $K$
$k \leftarrow 1$
$x^0 \leftarrow X0$
**repeat**
    **for** $i = 1$ $to$ $n$ **do**
        $x_i^k = \frac{\sum_{j=1,j\neq i}^{n}(-a_{ij}x_j^{k-1}) + b_i}{a_{ii}}$
    **end for**
    **if** $||x^k - x^{k-1}|| < tor$ **then**
        STOP
    **end if**
    $k = k + 1$
**until** $k > K$
```

Gauss-Seidel method

A set of linear equation in form $Ax = b$, for solving the ith equation for x_i

$$x_i^k = \frac{-\sum_{j=1}^{i-1}(a_{ij}x_j^k) - \sum_{j=i+1}^{n}(a_{ij}x_j^{k-1}) + b_i}{a_{ii}} \qquad \texttt{for } i = 1, 2, 3 \ldots n$$

This is called Gauss-Seidel method, the matrix form of it is shown as follows:

$$(D - L)x^k = Ux^{k-1} + b$$

or,$x^k = (D - L)^{-1}Ux^{k-1} + (D - L)-1b$, where where D denotes diagonal matrix whose diagonal is same as S , L and U are lower and upper triangular parts of A. Algorithm is shown as follow:

Algorithm A.2 Gauss-Seidel algorithm

```
initial guess value $X0$
tolerance $tor$
max number of iteration $K$
$k \leftarrow 1$
$x^0 \leftarrow X0$
repeat
    for $i = 1$ $to$ $n$ do
        $x_i^k = \frac{-\sum_{j=1}^{i-1}(a_{ij}x_j^k) - \sum_{j=i+1}^{n}(a_{ij}x_j^{k-1}) + b_i}{a_{ii}}$
    end for
    if $||x^k - x^{k-1}|| < tor$ then
        STOP
    end if
    $k = k + 1$
until $k > K$
```

Bibliography

[1] A. Fettweis, “ Wave digital filters: Theory and practice”, Proc. IEEE, vol. 74, no. 2, pp. 270-327, February 1986.

[2] Bilbao, Stefan D. “ Wave and scattering methods for numerical simulation”, John wiley & Sons, Ltd, ISBN 0-470-87017-6

[3] V. Belevitch, “ Classical Network Theory”, Holden-Day, San Francisco, 1968

[4] A. Fiedler and H. Grotstollen, “ Simulation of power electronic circuits with principles used in wave digital filters”, IEEE Transaction on Industry Applications, v.33, n.3, pp.49-57, Jan,-Feb 1997

[5] Carlos Christoffersen, “ Transient Analysis of Nonlinear Circuits Based on Waves”, Proceeedings of the 7th International Conference on Scientific Computing in Electrical Engineering (SCEE 2008), September 28 - October 3 2008, Helsinki Institute of Technology, Finland

[6] K. Meerkotter and R. Scholtz, “ Digital simulation of nonlinear circuits by wave digital filter principles”, Proc. IEEE Intl. Symp. Circuits Syst., Portland, OR, May 811, 1989, vol. 1, pp. 720-723.

[7] A. Sarti and G. De Poli, “ Toward nonlinear wave digital filters”, IEEE Transactions on Signal Processing, v 47, n 6, p 1654-68, June 1999

[8] Felderhoff, T., “ A new wave description for nonlinear elements”, Proc. IEEE

International Symposium on Circuits and Systems ISCAS '96., Pp. 221 - 224, 1996

[9] Klaus Meerkotter and Thomas Felderhof, " Simulation of nonlinear transmission lines by wave digital filter principles" ISCAS, pp. 875-878, San Diego, CA, 1992

[10] L. O. Chua, " Nonlinear circuits", IEEE Trans. Circuits Syst., vol. CAS-31, pp. 6987, Jan. 1984

[11] A. Sarti and G. De Poli, " Generalized adaptors with memory for nonlinear wave digital structures", in Proc. EUSIPCO, Eighth EuroSignal Process. Conf., Trieste, Italy, Sept. 1013, 1996, vol. 3, pp. 1941-1944.

[12] M. Toda, " Nonlinear lattice and soliton theory", Trans. on Circuits and Systems, vol. CAS-8, pp. 542-554, 1983

[13] Thomas Felderhoff, " Jacobi's method For Massive Parallel Wave Digital Filter Algorithm", IEEE ICASSP – 96. pp. 1621-1624, May 1996.

[14] Thomas Felderhoff, " Simulation of Nonlinear Circuits with Period Doubling and Chaotic Behavior by Wave Digital Filter Principles", IEEE Trannsactions on Circuits and Systems Fundamental Theory and Applications, Vol. 41 No.7, July 1994

[15] Alfred Fettweusand and Klaus Meerkqtter, " On Adaptors for Wave Digital Filters", IEEE Trannsactions on Acostics, Speech, and Signal Processing, VOL. ASSP-23, NO. 6, DECEMBER 1975

[16] T.S.K.V. Iyer, " Circuit Theory", McGraw-Hill Education, ISBN-13: 978-0074516812

[17] Jiri. Vlach and Kishore Singhal, " Computer Methods for Circuit Analysis and Design", Van Nostrand Reinhold Publishing, ISBN 0-442-28108-0

[18] Farid N. Najm, “ Circuit Simulation”, John Wiley & Sons, Inc. , ISBN 978-0-470-53871-5

[19] Alan V. Oppenheim and Alan S. Wlllsky , “ Signals and systems” 2nd ed, Prentice-Hall, Inc, ISBN 978-0138147570

[20] Richard L. Burden and J. Douglas Faires, “ Numerical Analysis” 5th edition, PWS Publishing Company

[21] A.R.Newton and A.L.Sangiovanni-Vincentelli, “ Relaxation-Based slectrical simulation”, IEEE Trans. on Electron Devices, vol. 30, issue-9, PP. 1184-1207, 1983

[22] Message Passing Interface Forum, “ MPI: A Message-Passing Interface Standard Version 2.2”

[23] G.E.Karniadakis and R .M. Kirby II , “ Parallel Scientific Computing in C++ and MPI”, Cambridge University Press 2003, ISBN-13: 978-0521520805

[24] Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, “ Introduction to Parallel Computing” , Addison Wesley, ISBN: 0-201-64865-2

[25] C. Christoffersen, “ Implementation of Exact Sensitivities in a Circuit Simulator Using Automatic Differentiation”, 20th European Conference on Modeling and Simulation, May 2006, pp. 238-243.