Tunable Difference Equations for the Time-Domain Simulation of Power System Operating States

by

JESÚS CALVIÑO-FRAGA

Ingeniero Electrónico, Universidad Simón Bolívar, Venezuela, 1987

M.Sc., The University of British Columbia, 1999

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES

(Department of Electrical and Computer Engineering)

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

July 2003

Jesus Calvino-Fraga

Department of Electrical and Computer Engineering

The University of British Columbia
Vancouver, Canada

Date July, 21 2003

Abstract

This report presents the development of a new family of difference equations for the numerical solution of systems of differential equations arising in electric power circuits. The main advantage of these difference equations is their adjustability, which allows for the tuning of the resulting formulas while using large time steps. The tuned formulas have been used successfully in the time-domain simulation of 50/60-Hz solutions of electric power circuits for linear and non-linear problems. The linear steady state and the non-linear power flow problems can be solved, with no error either in magnitude or in phase, at a comparable computational cost to traditional techniques such as the well-known phasor analysis and Newton-Raphson iterations. Additionally, the techniques developed do not use complex numbers, resulting in a faster matrix factorization for linear systems, or a smaller Jacobian matrix for non-linear systems. The main advantage of the proposed techniques is that due to the time-domain nature of the simulation, where the transition from one time solution to the next represents a small transition between states, the technique is able to closely follow the trajectory of the system even for ill-conditioned power flow problems where traditional Newton-Raphson iterations fail to converge, such as when the system buses are beyond or near the point of voltage collapse. With the proposed novel formulas, the computational cost of generating PV curves across all busses of the power system network is greatly reduced, and the resulting curves clearly show the point of voltage collapse as well as any unstable voltage/power combination thereafter. The ability to quickly obtain PV curves makes the proposed techniques particularly well suited for online power system monitoring of operating states. Also, since these new difference equations introduce very little error for frequencies lower than the tuned frequency, they are also well suited for the solution of dynamic power flow problems such as those resulting from loads variation, transformers tap change, and outages.

Table of Contents

List of Tables

List of Figures

Acknowledgments

I owe a tremendous debt of gratitude to Dr. Jose R. Marti, my Ph. D. thesis supervisor. His constant guidance, advice, help, encouragement, and patience as well as financial support, have been invaluable during my Ph. D. research.

I want to thank my friends Dr. Roberto Rosales and Dr. Luis Linares for the help, advice, and encouragement they gave me in all these years at UBC. Also, I want to thank Dr. Ray Burge and Peter Dickson for helping me when I needed it the most.

Finally I want to thank my senseis, Dr. George Iwama, Dr. Darin Bennet, and Claudio Lerner not only for all their teachings but also for their example of excellence and discipline.

CHAPTER 1: Introduction and Overview

1.1 Background and Motivation

Numerical methods for the solution of systems of differential equations are at the heart of most transient analysis electric circuit simulators. Popular computer programs such as the EMTP, SPICE, as well as many of their predecessors and derivatives use such methods. For some, such as the EMTP, only the well known Backward Euler and Trapezoidal one step methods are used [1;2]. Multistep methods have been also used to simulate electric circuits. For instance, the well-known Adams-Bashforth and Adams-Moulton methods are sometimes used, although they lack much of the stability required for a general-purpose electric circuit simulator [3].

It was in 1952 when the multistep Backward Differentiation Formulae (BDF) methods were first described [4], and in 1963 when Dahlquist [5] described the properties of such multistep methods. Dahlquist demonstrated that such methods could be derived using Taylor series approximations from which a truncation error is computed. Moreover, he studied the methods stability properties; in particular, a very desirable property called A-stability. This property is defined as the ability of a given method to converge to a stable solution when it is used to solve a system of first order differential equations, where the real part of the eigenvalues of the system are all negative. Since most passive electric circuit have negative real part eigenvalues, A-stability is a very desirable property to have in the numerical method used to simulate such circuits.

BDF methods came to prominence in 1971 when Gear [6] used them in his general-purpose ODE solver[1]. At the time it was known that such methods were A-stable for orders one and two, and A(α)-stable for orders three to six [7]. The A(α)-stability property has to do with the ability of a given multistep method to converge to a stable solution not for all real part negative eigenvalues but for just a range which often comprises eigenvalues far apart from each other. This seemed to be an appropriate solution since most electric circuits show this eigenvalues range. Systems with this characteristic are often referred as 'stiff systems' [8]. They comprise not only electric circuits, but many other mechanical and chemical systems as well [9].

BDF methods are the most widely used methods for the solution of stiff systems of differential equations [10]. Unfortunately this class of methods become unstable for orders larger than six [11;12]. Many techniques have been suggested to increase the stability and order of BDF methods: [12;13;14;15;16;17] just to name a few. Although very effective, in general, these approaches require very small or adjustable time steps that impose an important burden in the performance of a general-purpose circuit simulator.

Simpler numerical methods often result in faster simulators. One such method that stands over the rest is the trapezoidal method, whose properties are well described by Dahlquist in [5]. The trapezoidal method has also drawbacks, in particular, the oscillatory behaviour of this method when discontinuities are came across is well known [2;18;19]. One of the first solutions to this problem consisted of the addition of external damping [20], which is functionally equivalent to a class of adjustability or tuning presented by many other authors[3;21]. Since the addition of damping results in an effective modification of the

[1] Reference [6] includes the FORTRAN II source code for the general-purpose ODE solver DIFSUB that became very popular in the 1970's. That is the reason why BDF methods are also known as Gear's Formulas.

problem being solved, techniques that are more convenient were eventually developed. Gear and Osterby in [22] explored the alternative of order reduction to get rid of undesirable numerical oscillations. It was not until the Critical Damping Algorithm (CDA) was introduced by Marti and Lin [2] that an efficient, simple, and elegant method to suppress the numerical oscillations of the trapezoidal rule was obtained. Nevertheless, the problem remains when the solution of the system has to be obtained in real-time, and the time step reduction required by CDA is not possible [23].

Another important problem often found in electric circuit simulations is the so-called oscillatory problem. In many systems of ordinary differential equations, such as those resulting from electric circuits, complex conjugate eigenvalues or forcing periodical functions result in oscillatory responses. The use of phasor analysis, described in many basic circuit analysis textbooks, is a well-established technique for the solution of the linear oscillatory problem in electric circuits. Unfortunately, the presence of non-linear elements in the system being simulated renders this technique unusable and alternative methods, such as Newton-Raphson iterations are used [24],[25]. Even for the simpler non-linear power flow problem involving linear network elements but non-linear loads, phasor analysis is not appropriate, and again iterative techniques such as Newton-Raphson iterations, or one of its variations, are used [26].

This thesis describes a family of new methods used for the simulation of electric circuits. These methods have very desirable properties such as A-stability, high accuracy, and low computational burden. In particular, two kinds of methods are described. The first one, derived from a third-order interpolating polynomial, shows stability properties similar to that of the Backward Euler method, and the accuracy of the Trapezoidal method. This method is useful for the real time simulation of circuits with discontinuities, but it is mainly presented in this report as a simple example of the general method's derivation. The other group of methods are derived from interpolating polynomials of fourth or greater order, tuned at a particular frequency. These are fixed time-step methods, which produce an exact solution for linear and non-linear AC circuits, both in magnitude and phase for the frequency they are tuned to. Since almost all electric power systems work at one fixed frequency, the potential use of these methods for the solution of such circuits is evident.

In this thesis, tuned time-domain methods are used to accurately solve linear steady-state circuits as well as non-linear static and dynamic power flow electric power circuits. For the simulation of linear steady-state circuits, the methods are faster than the well-known phasor analysis for circuits with a large number of nodes. When used to solve the static power flow problem, the cost of the tuned time-domain methods is similar to that of Newton-Raphson iterations. But unlike Newton-Raphson iterations, the tuned time-domain methods produce useful results for ill-conditioned systems, such as those undergoing voltage collapse. The ability of the tuned time-domain methods to solve electric power systems at or beyond the point of voltage collapse makes them ideal for the economical generation of PV curves. Also, since the methods are still accurate enough for frequencies much lower than the tuned frequency, they can also be used to simulate the dynamics of power flow electric circuits.

1.2 Electric Power Circuit Analysis Techniques

In this section, four electric power circuit analysis techniques, relevant to this research are briefly described. The first two, nodal analysis and modified nodal analysis are used to implement the algorithms and techniques described in this research. The other two, phasor analysis and power flow analysis are the techniques against which this research is validated and compared.

1.2.1 Nodal Analysis

Of the available techniques for the computer simulation of electric circuits, nodal analysis is the easiest to implement [27]. A variation of nodal analysis, the so called modified nodal analysis or MNA [28], was used on this research to implement a circuit simulator to develop and test the new simulation techniques. Nodal analysis is based on Kirchhoff's current law, which states that the algebraic sum of currents entering a node is zero. Nodal analysis consist on finding the admittance matrix Y and the vector of independent current sources I so that

$$YV = I, \qquad (1.1)$$

where the dimension of matrix Y is equal to the number of nodes in circuit minus one (for the reference node), and solving for the voltages V. Matrix Y and vector I can be built by inspection following these three simple rules:

1) The diagonal elements of the Y matrix come from the sum of all admittances at the corresponding node.

2) The non-diagonal elements of the Y matrix come from the negative of the admittance connecting the corresponding nodes.

3) The elements of the independent current sources I results from the sum of all current sources coming into the corresponding node.

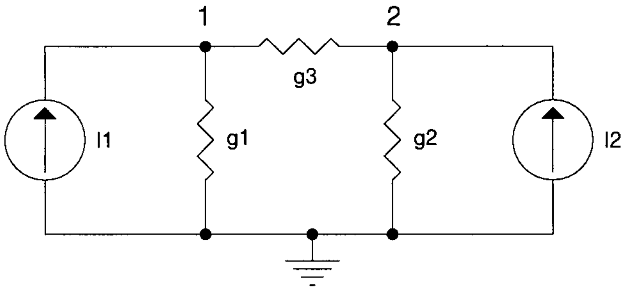


Figure 1.1 Simple circuit for Nodal Analysis example.

For instance, consider the circuit of Figure 1.1. Notice first that it has two nodes (excluding the ground reference node), therefore the *Y* matrix will have two rows and two columns. The sum of admittances at node 1 is *g1+g3*. Similarly, for node 2 the sum is *g2+g3*. The negative of the admittance connecting nodes 1 and 2 is *–g3*. The sum of all current sources entering node 1 is *I1*, and the sum of all current sources entering node 2 is *I2*. Therefore

$$\begin{bmatrix} g1+g3 & -g3 \\ -g3 & g2+g3 \end{bmatrix} \begin{bmatrix} V1 \\ V2 \end{bmatrix} = \begin{bmatrix} I1 \\ I2 \end{bmatrix}. \tag{1.2}$$

The system of equations (1.2) can be generalized to any number *n* of nodes as

$$\begin{bmatrix} Y_{11} & Y_{12} & \dots & Y_{1n} \\ Y_{21} & Y_{22} & \dots & Y_{2n} \\ \dots & \dots & \dots & \dots \\ Y_{n1} & Y_{n2} & \dots & Y_{nn} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \dots \\ V_n \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 \\ \dots \\ I_n \end{bmatrix}, \tag{1.3}$$

or for one particular node *i* as

$$I_i = \sum_{k=1}^{n} Y_{ik} V_k . \tag{1.4}$$

Nodal analysis can be applied to dynamic elements such as capacitors and inductors as well. This is achieved by discretizing the continuous time element using a difference equation. The application of a difference equation in the simulation of electric circuits is based in the substitution of the element being simulated by an equivalent one which will perform the same task as the differential equation that represents such element [1]. Let us, for example, perform this substitution for the voltage across an inductor, given by

$$V_L(t) = L\frac{di_L(t)}{dt}. \tag{1.5}$$

The idea is to replace the differential part of the equation by a difference formula. Let us choose, for simplicity, the Backward Euler formula

$$\frac{di_L(t)}{dt} \approx \frac{i(t) - i(t-h)}{h}, \tag{1.6}$$

where *h* is a time step, small enough so the approximation does not introduce too much error. Replacing (1.6) back into (1.5)

$$V_L(t) \approx L\frac{i(t) - i(t-h)}{h}, \tag{1.7}$$

and getting *i(t)* out results in

$$i(t) \approx \frac{h}{L}V_L(t) + i(t-h)\ . \tag{1.8}$$

Equation (1.8) is usually represented by the equivalent circuit shown in Figure 1.2, in which the discrete equivalent of the inductor is a resistor with a current source in parallel, whose value depends on the previous value of the current on the inductor:

$$\begin{aligned} h_L &= i(t-h) \\ g_L &= \frac{h}{L} \end{aligned} \tag{1.9}$$

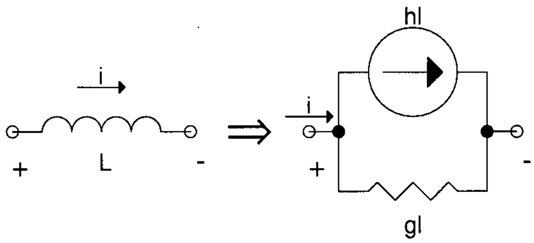


Figure 1.2 Equivalent circuit of an inductor when a difference equation is used to model it.

The procedure described above is also valid for capacitors, resulting in a similar equivalent circuit. Figure 1.3 shows a circuit with a capacitor and an inductor as well as its equivalent discretized representation. Applying nodal analysis to the discretized circuit results in the system of equations

$$\begin{bmatrix} g1+gL & -gL \\ -gL & gL+gC \end{bmatrix} \begin{bmatrix} V1 \\ V2 \end{bmatrix} = \begin{bmatrix} I1-hL \\ hL+hC+I2 \end{bmatrix}, \tag{1.10}$$

where hL and hC change from time step to time step accordingly to discretization formula used.

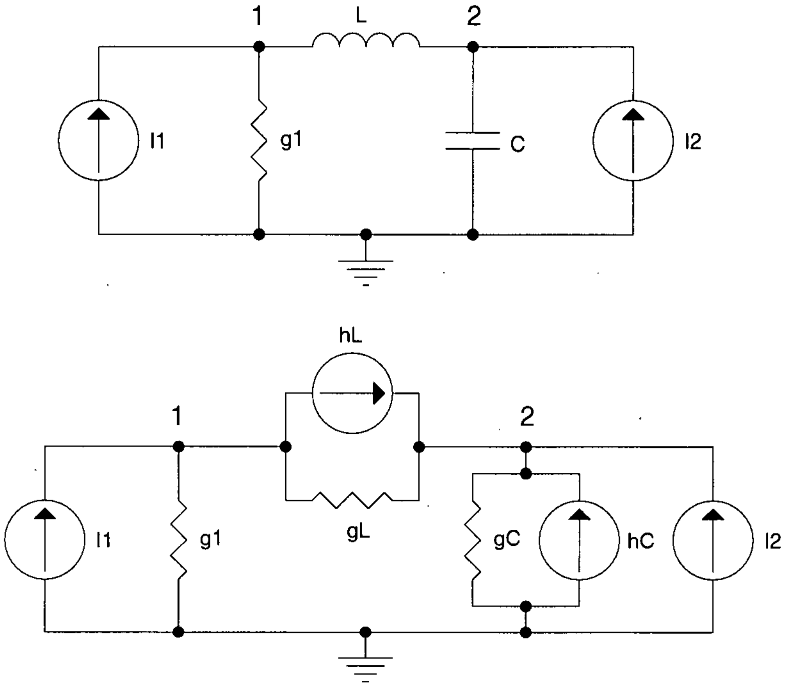


Figure 1.3 Circuit with a capacitor and an inductor (top), and its equivalent discrete representation (below).

As mentioned before, the circuit of Figure 1.2 is just an approximation. How good this approximation is can be determined either by finding the transfer function of the equivalent circuit and comparing it with that of the real circuit as proposed in [2], or plotting the frequency response of the Z-transform of the difference equation [29][2]. Then, for an inductor of one Henry, discretized using the backward Euler formula, it can be shown [2] that

$$X(z) = \frac{1}{h}\frac{z}{z-1} \tag{1.11}$$

is its Z-transform. Now, replacing *z* with $e^{j\omega h}$, the magnitude and phase of *X(z)* can be plotted as shown in Figure 1.4. These plots represent both the magnitude and phase errors for the discretized inductor for the range of frequencies DC to *1/(2h)* .

[2] Page 494, equation 12-83, but it is easier to follow from equations 12-86 and 12-87 in page 495.

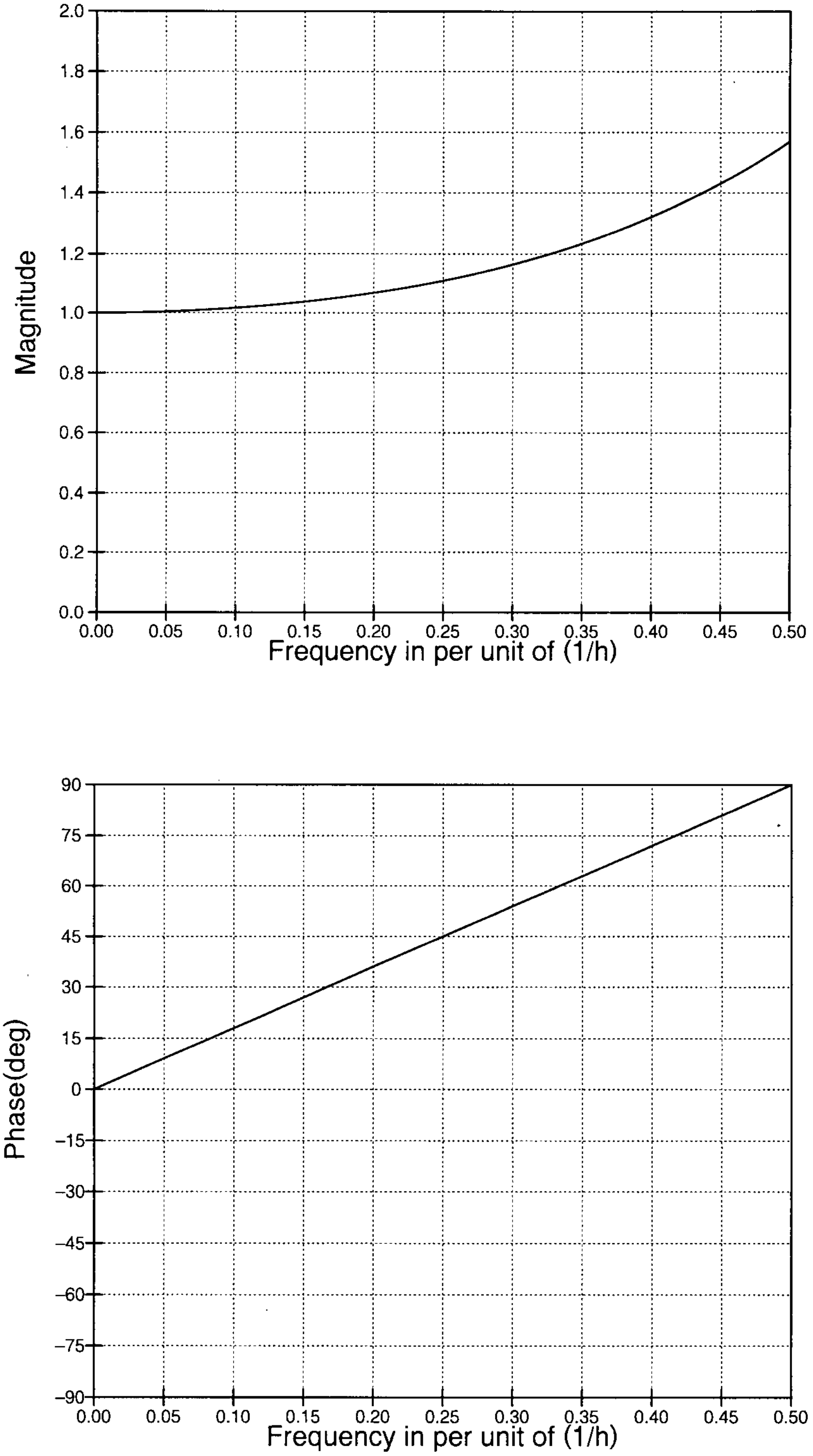


Figure 1.4 Magnitude distortion and phase error introduced by the backward Euler formula.

To analyze the stability properties of a given rule many authors ([6], [29], [27], [3]) refer to the initial-value problem

$$v'(t) = \lambda v(t), \;\; v(0) = 1 \;, \tag{1.12}$$

which has the known solution

$$v(t) = e^{\lambda t}, \tag{1.13}$$

where λ is the time constant or eigenvalue of equation (1.12). This time constant is, in general, a complex number whose real part is negative: Re(λ)<0. For larger systems, it is possible to apply a modal transformation in such a way that a set of independent equations with the same elementary form can be obtained. In that case, λ will represent the eigenvalues of the state matrix of the system and *v(t)* will be a vector.

For example, consider the following formula, which is the BDF[3] second order formula, and analyze its stability properties:

$$3v(t) - 4v(t-h) + v(t-2h) \approx h(2v'(t)) \,. \tag{1.14}$$

If we use the initial value problem of equation (1.12) and replace it in equation (1.14), we get

$$3v(t) - 4v(t-h) + v(t-2h) = h(2\lambda v(t)) \,. \tag{1.15}$$

Following the procedure outlined in [27] we want the product $h\lambda$

$$h\lambda = \frac{3v(t) - 4v(t-h) + v(t-2h)}{2v(t)} \,. \tag{1.16}$$

Now, lets evaluate the function using equation (1.13) for all possible eigenvalues $j\omega$, and see what values of $h\lambda$ lay inside it:

$$h\lambda(\omega) = \frac{3e^{j\omega t} - 4e^{j\omega(t-1)} + e^{j\omega(t-2)}}{2e^{j\omega t}} \,. \tag{1.17}$$

[3] Backward differentiation formulas.

Simplifying

$$h\lambda(\omega) = \frac{3 - 4e^{-j\omega} + e^{-2j\omega}}{2}. \tag{1.18}$$

Regrouping we get the loci of the roots of equation (1.14)

$$h\lambda(\omega) = \frac{3e^{2j\omega} - 4e^{j\omega} + 1}{2e^{2j\omega}}. \tag{1.19}$$

If we evaluate equation (1.19) for $0 \leq \omega < 2\pi$, and plot the result in the complex plane, we get the locus of $h\lambda$ (Figure 1.5). Now, by plotting the actual values $h\lambda$ of the simulated system in the complex plane (the eigenvalues multiplied by the time step), it is possible to determine if the rule is stable or not for such system. For some formulas the area of stability is inside the closed curve, for example, all Adams-Bashforth formulas, and all Adams-Moulton formulas of order larger than two[4]; for others, such as Backward Euler, Trapezoidal[5], and all BDF formulas, the area of stability is outside the curve [29]. If the area of stability lays inside the closed curve the method is explicit, otherwise the method is implicit.

Observe from Figure 1.5 that the first and second order BDF[6] are stable for all eigenvalues λ with negative real part, despite the value of the time step h. This property, once more, is known as A-stability [5]. A-stability is important because for most practical electric circuits the real part of the eigenvalues is negative.

To demonstrate the importance of A-stability, consider the circuit of Figure 1.6, whose eigenvalues are the roots of

[4] The second order Adams-Moulton is also the Trapezoidal rule.

[5] The stability zone of the Trapezoidal rule is the complete complex left plane. It can be pictured as a circle of infinite ratio centered in the right hand side of the complex plane touching the origin.

[6] The first order BDF is also the Backward Euler formula.

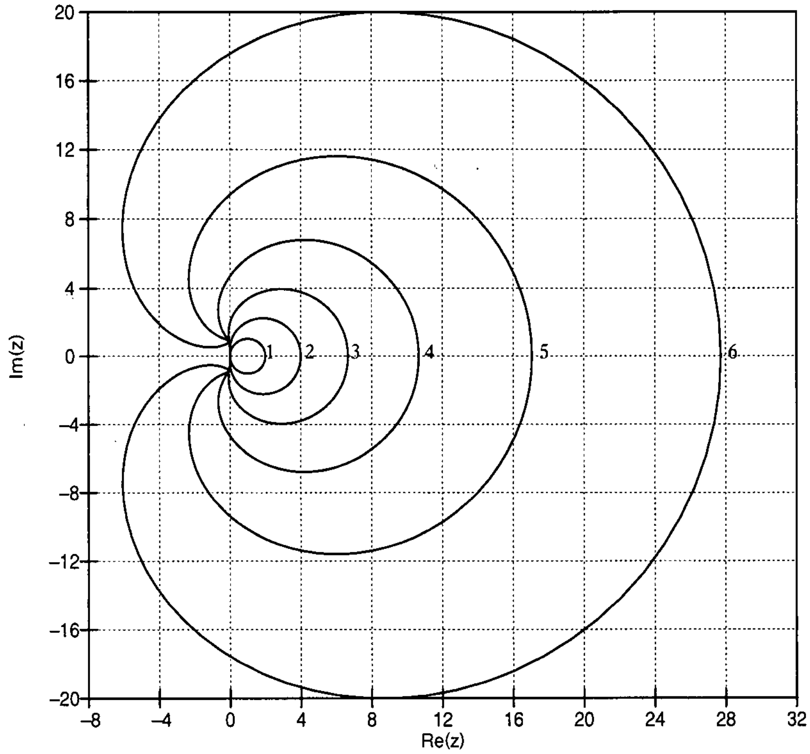


Figure 1.5 Loci of the roots for the BDF formulas of order 1 to 6.

$$(L_1C_1R_2)S^2 + (L + C_1R_1R_2)S + (R_1 + R_2) = 0\,. \tag{1.20}$$

First, the circuit of Figure 1.6 is simulated using the fourth order BDF formula and h=1.0 ms. The area of stability for the BDF fourth order formula, and the values $h\lambda$ for the given circuit are shown in Figure 1.7, which in this case lie inside the curve, that is, in the area in which the rule is not stable. The effect on the time-domain simulation is shown in Figure 1.8 where the voltage across the capacitor is plotted. Notice that after about 0.04 s the output is noticeable unstable. If the value of h is set to 0.5 ms, then the product $h\lambda$ lies outside the curve (Figure 1.9), and the output of the time-domain simulation is numerically stable (Figure 1.10).

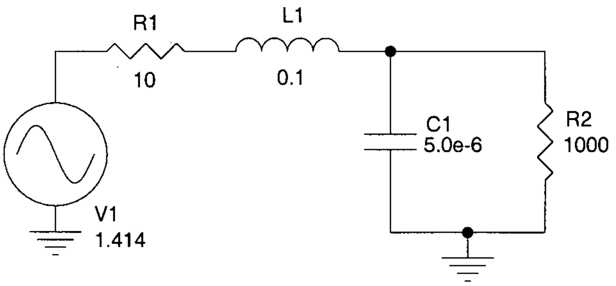


Figure 1.6 Circuit used in the simulations to test the stability of different numerical methods.

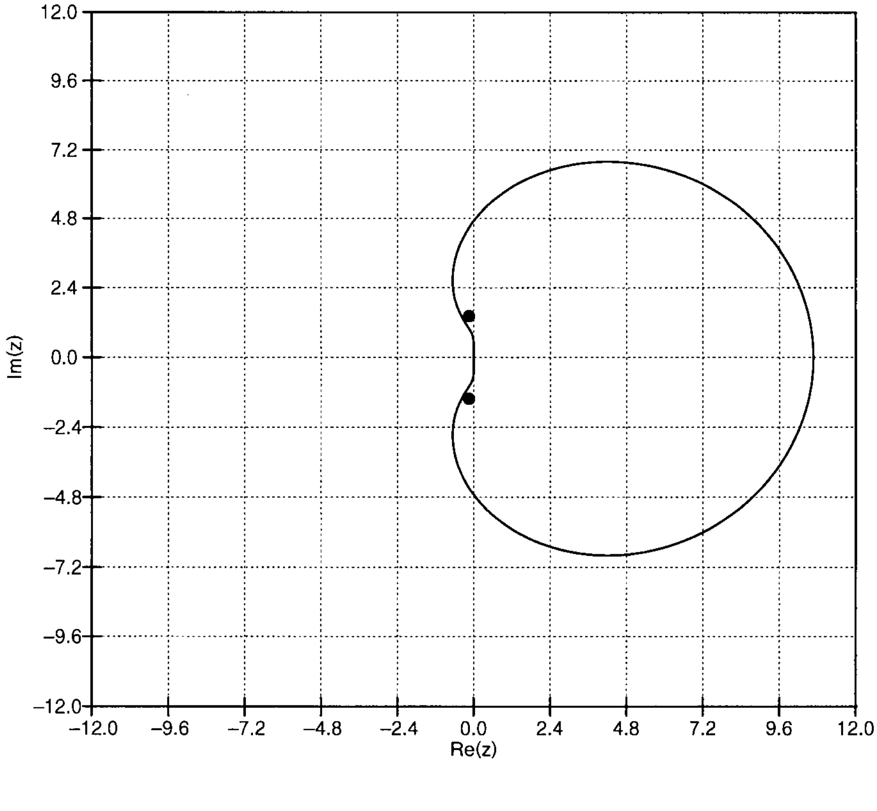


Figure 1.7 For the circuit of Figure 1.6 the product $h\lambda$ (the two dots, h=1ms) is inside the area in which the difference equation is unstable.

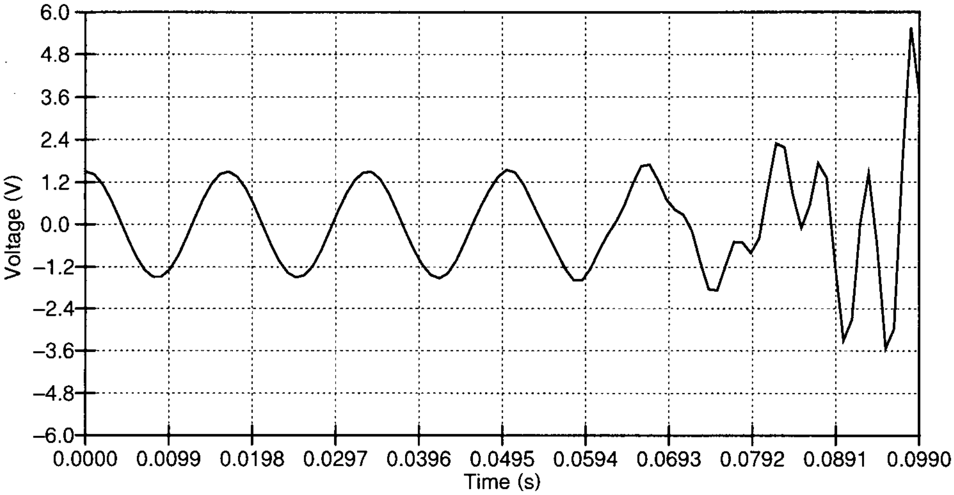


Figure 1.8 Voltage across the capacitor of Figure 1.6. After a number of time steps the simulation becomes unstable since the product $h\lambda$ is not in the area of stability (Figure 1.7).

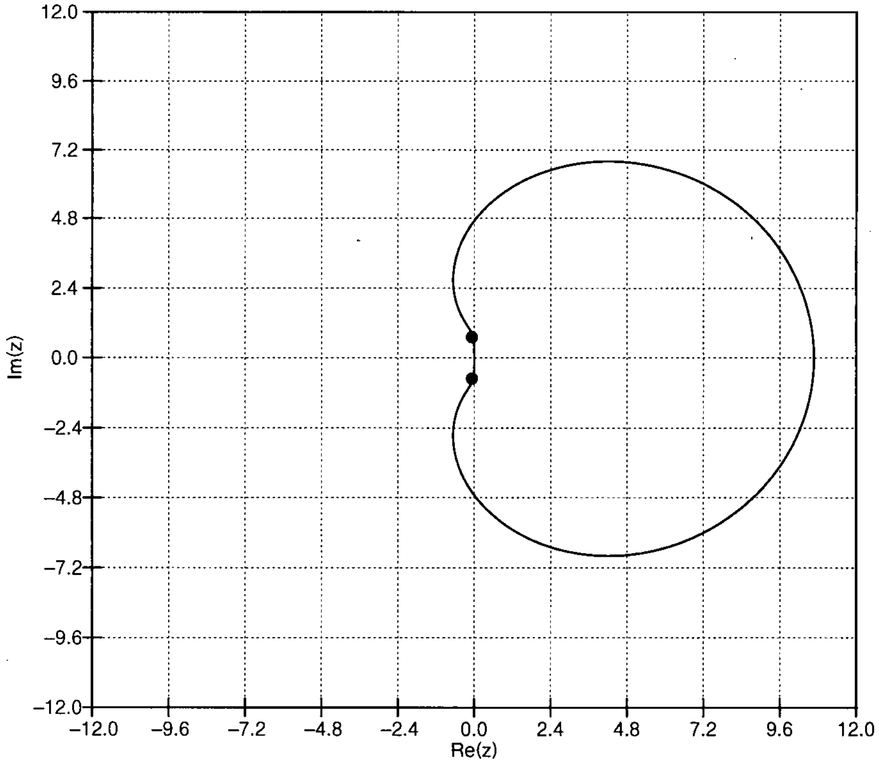


Figure 1.9 One way of obtaining a stable response is to decrease (or increase) h so the product $h\lambda$ is in the area of stability.

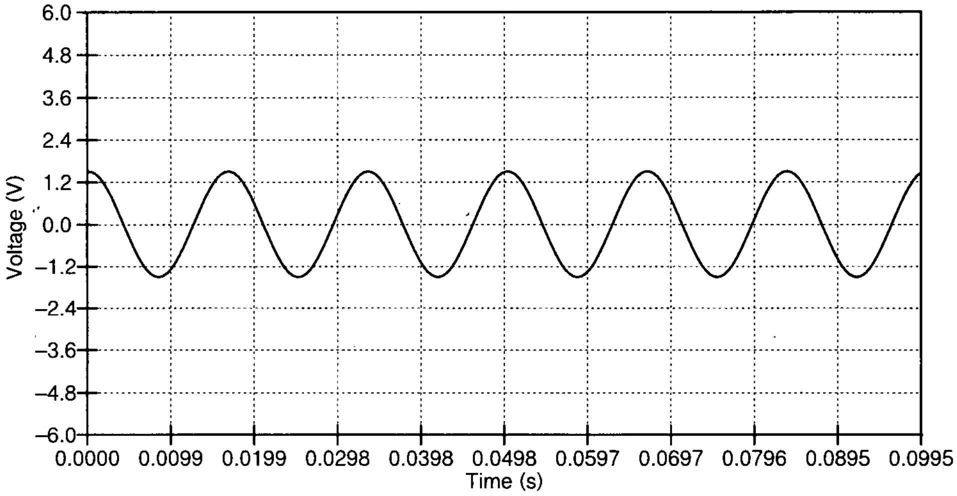


Figure 1.10 The voltage in the capacitor of Figure 1.6 after decreasing the value of h to 0.5 ms is stable, since the product $h\lambda$ is in the area of stability as shown in Figure 1.9.

1.2.2 Modified Nodal Analysis

Standard nodal analysis, as described in the previous section, can only deal with independent current sources. Although it is possible to add independent voltage sources to the nodal analysis formulation by means of Norton equivalent circuits, modified nodal analysis or MNA [28] is often simpler and easier to implement. Additionally, it solves for the current coming out of the voltage sources, a handy figure to have when power computations are needed. MNA basically consists on the aggregation of extra equations and variables to the standard nodal analysis formulations so to impose new circuital conditions, as for example, an applied voltage between two nodes.

Consider for instance the circuit of Figure 1.11, which is the same circuit of Figure 1.1 with a voltage source of value E between nodes 1 and 2. Assuming the current from this additional voltage source moves from the positive terminal to the negative terminal, it will add to the total current of node 2 and subtract to the total current of node 1 as well as to set the voltage difference between nodes 1 and 2 to a given value:

$$\begin{aligned} V1 - V2 &= E \\ I_{E-} &= -I_{E+} \end{aligned}, \tag{1.21}$$

where I_{E+} and I_{E-} are the currents flowing into the positive and negative terminals of the voltage source. The modified system of equations is therefore

$$\begin{bmatrix} g1+g3 & -g3 & 1 \\ -g3 & g2+g3 & -1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} V1 \\ V2 \\ I_E \end{bmatrix} = \begin{bmatrix} I1 \\ I2 \\ E \end{bmatrix}. \tag{1.22}$$

Comparing the system of equations (1.22) and (1.2) we observe that:

a) The matrix dimension has been incremented by one, despite the number of nodes remaining constant.

b) The current through the voltage source is one of the unknowns, along with the node voltages.

Not only independent voltage sources, but also dependent current and voltage sources, ideal transformers, and operational amplifiers can be represented using MNA [27]. But since they are not relevant for this report the reader is referred to the aforementioned reference.

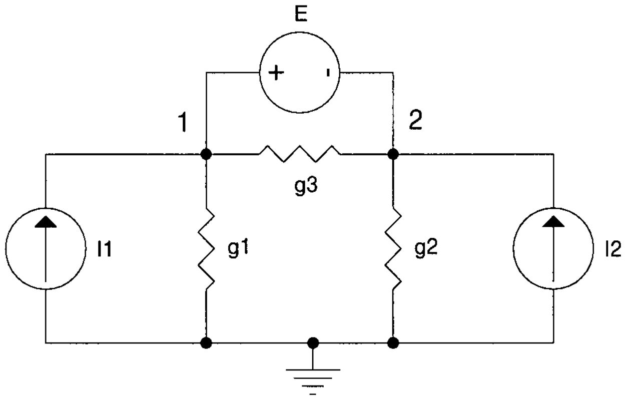


Figure 1.11 Simple circuit for Modified Nodal Analysis (MNA) example.

1.2.3 Phasor Analysis

In many electrical circuits, such as those arising in electric power and communications, the most frequent type of source excitation is sinusoidal. Phasor analysis is similar to nodal analysis, but using complex numbers instead. This is possible because phasor representations for sources, inductors, and capacitors can be easily obtained. For instance, consider the top circuit of Figure 1.12, for which the following transformations are performed

$$\begin{aligned}
&I1 * \cos(wt + Ph1) \rightarrow I1\underline{|Ph1} \\
&I2 * \cos(wt + Ph2) \rightarrow I1\underline{|Ph2} \\
&L \rightarrow \frac{-j}{\omega L}, j = \sqrt{-1} \\
&C \rightarrow j\omega C, j = \sqrt{-1}
\end{aligned} \tag{1.23}$$

resulting in the bottom circuit of Figure 1.12. The voltages of the resulting circuit can be solved using nodal analysis:

$$\begin{bmatrix} g1 - \dfrac{j}{\omega L} & \dfrac{j}{\omega L} \\ \dfrac{j}{\omega L} & j\left(\omega C - \dfrac{1}{\omega L}\right) \end{bmatrix} \begin{bmatrix} V1 \\ V2 \end{bmatrix} = \begin{bmatrix} I1\underline{|Ph1} \\ I2\underline{|Ph2} \end{bmatrix}. \tag{1.24}$$

Similarly to the admittance matrix and the vector of independent sources, the vector of voltages resulting from the solution of (1.24) is made up of complex numbers as well. MNA techniques can be applied to phasor analysis too, as described in the preceding section, allowing for the easy incorporation of independent voltage sources.

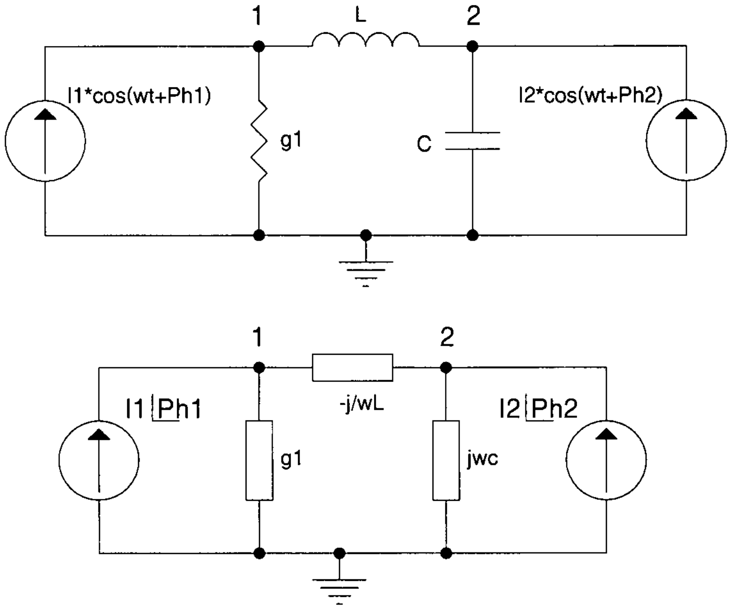


Figure 1.12 Simple circuit for Phasor Analysis example.

1.2.4 Power flow Analysis

"Power flow" or "load flow" analysis as it is some times called, is a fundamental tool for power system operation and planning. According to Bergen & Vittal [26], power flow is the most common of the electric power system computer simulations performed. It is somehow similar to phasor analysis in the sense that the magnitude and phase of the system buses voltages have to be computed, but instead of using currents as the independent variables, it uses powers. To describe how power flow analysis is performed, let us start with the definition of complex power[7]

$$S_i = V_i I_i^*, \tag{1.25}$$

[7] The complex conjugate of the current is used because of the way the complex power S is defined. The real part of the complex power is called active power or P and is given by $P=|V||I|cos(\theta)$, where θ is the angle between the voltage and the current; the number resulting from $cos(\theta)$ is known as the power factor. The imaginary part of S is called reactive power or Q and is given by $Q=|V||I|sin(\theta)$. If $V=|V|e^{j\phi}$ and $I=|I|e^{j(\phi-\theta)}$ the complex power is $S=VI^*=|V||I|e^{j\phi}\,e^{-j(\phi-\theta)}=|V||I|(cos(\theta)+\mathrm{j}\,sin(\theta))=P+jQ$.

where S_i, V_i, and I_i, are phasor quantities. As in the nodal analysis formulation equation (1.4), the net current into a bus can be computed as

$$I_i^* = \left(\sum_{k=1}^{n} Y_{ik} V_k \right)^* = \sum_{k=1}^{n} Y_{ik}^* V_k^* \tag{1.26}$$

Replacing equation (1.26) into equation (1.25) results in

$$S_i = V_i \sum_{k=1}^{n} Y_{ik}^* V_k^* \tag{1.27}$$

To simplify the computations, the following substitutions are performed to equation (1.27)

$$V_i = |V_i| e^{j\theta_i}, \theta_{ik} = \theta_i - \theta_k, Y_{ik} = G_{ik} + jB_{ik}, \tag{1.28}$$

resulting in

$$S_i = \sum_{k=1}^{n} |V_i||V_k| e^{j\theta_{ik}} (G_{ik} - jB_{ik}). \tag{1.29}$$

Applying Euler's substitution to the exponential part, results in

$$S_i = P_i + jQ_i = \sum_{k=1}^{n} |V_i||V_k| (\cos(\theta_{ik}) + j\sin(\theta_{ik}))(G_{ik} - jB_{ik}). \tag{1.30}$$

Finally, separating real and imaginary parts yields the equations to be solved for each of the system buses

$$P_i = \sum_{k=1}^{n} |V_i||V_k| (G_{ik} \cos(\theta_i - \theta_k) + B_{ik} \sin(\theta_i - \theta_k)). \tag{1.31}$$

$$Q_i = \sum_{k=1}^{n} |V_i||V_k| (G_{ik} \sin(\theta_i - \theta_k) - B_{ik} \cos(\theta_i - \theta_k)) \tag{1.32}$$

In typical power flow problems the active and reactive power on some buses is specified, therefore those are known as PQ buses, for which both equation (1.31) and equation (1.32) must be used to compute the bus voltage magnitude and phase. For other buses, the magnitude of the voltage and active power are specified, the so called PV buses, for which only the active power equation (1.31) is needed to compute the bus voltage angle. Since the total flow of power is not known a priori, it is customary to set one system bus as free to provide any *P* or *Q*. For this swing or slack bus, both the voltage magnitude and phase are assumed, and neither equation (1.31) nor equation (1.32) are required.

Notice from equations (1.31) and (1.32) that the resulting system of equations is nonlinear due to the products of voltages. The preferred method for the solution of this nonlinear system of equations is Newton-Raphson iterations. It consists of the linearization of the system to be solved using the first term of the Taylor's series expansion. Before proceeding with such linearization, equations (1.31) and (1.32) can be rewritten as

$$\begin{aligned} f_{P_i}(V_1,\dots,V_n,\theta_1,\dots,\theta_n) &= \sum_{k=1}^{n}|V_i||V_k|(G_{ik}\cos(\theta_i-\theta_k)+B_{ik}\sin(\theta_i-\theta_k))-P_i \\ f_{Q_i}(V_1,\dots,V_n,\theta_1,\dots,\theta_n) &= \sum_{k=1}^{n}|V_i||V_k|(G_{ik}\sin(\theta_i-\theta_k)-B_{ik}\cos(\theta_i-\theta_k))-Q_i \end{aligned}, \qquad (1.33)$$

which after linearization with the first term of the Taylor's series becomes,

$$\begin{bmatrix} \frac{\partial f_{P1}}{\partial\theta_1} & \cdots & \frac{\partial f_{P1}}{\partial\theta_n} & \frac{\partial f_{P1}}{\partial|V_1|} & \cdots & \frac{\partial f_{P1}}{\partial|V_m|} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f_{Pn}}{\partial\theta_1} & \cdots & \frac{\partial f_{Pn}}{\partial\theta_n} & \frac{\partial f_{Pn}}{\partial|V_1|} & \cdots & \frac{\partial f_{Pn}}{\partial|V_m|} \\ \frac{\partial f_{Q1}}{\partial\theta_1} & \cdots & \frac{\partial f_{Q1}}{\partial\theta_n} & \frac{\partial f_{Q1}}{\partial|V_1|} & \cdots & \frac{\partial f_{Q1}}{\partial|V_m|} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f_{Qm}}{\partial\theta_1} & \cdots & \frac{\partial f_{Qm}}{\partial\theta_n} & \frac{\partial f_{Qm}}{\partial|V_1|} & \cdots & \frac{\partial f_{Qm}}{\partial|V_m|} \end{bmatrix} \begin{bmatrix} \Delta\theta_1 \\ \cdots \\ \Delta\theta_n \\ \Delta|V_1| \\ \cdots \\ \Delta|V_m| \end{bmatrix} = \begin{bmatrix} f_{P1} \\ \cdots \\ f_{Pn} \\ f_{Q1} \\ \cdots \\ f_{Qm} \end{bmatrix}, \qquad (1.34)$$

where *n* is the sum of the number of PQ and PV buses for which the phase angle θ of the voltages need to be computed; and *m* is the number of PQ buses, for which only the voltages magnitude need to be computed. This results in a Jacobian matrix of dimension (*n*+*m*). The system of equations (1.34) is solved by assuming an initial value for the buses voltage magnitude and phase with which to compute the values of the functions in equation (1.33) and the Jacobian matrix of the system of equations (1.34). Then the changes on the buses voltage magnitude and phase are computed from (1.34), and the process is repeated until the changes in the buses voltage magnitude and phase reach a given tolerance.

1.3 Outline of the Thesis

In Chapter 1 we reviewed some of the required background information relevant for this thesis. In Chapter 2 the derivation of adjustable difference equations is presented as well as their application to time-domain simulations. The tuning of the adjustable methods is presented in Chapter 3, with their application in the solution of forced sinusoidal circuits in the time-domain and comparison to phasor analysis. Chapter 4 deals with the application of the tuned time-domain methodology described in Chapter 3 to the solution of the power flow problem and its comparison to Newton-Raphson iterations. Also in Chapter 4, some unique properties of the tuned time-domain methodology are used to assess the voltage stability of electric power systems. In Chapter 5, the dynamics of the power flow are modelled and simulated. Finally, Chapter 6 presents the conclusions of this work, as well as some recommendations for future work.

CHAPTER 2: Adjustable Methods

2.1 Introduction

Linear multistep methods are usually denoted with the following difference equation[3]

$$\begin{gathered}\sum_{j=0}^{q}\alpha_j y_{n-j} = h\sum_{j=0}^{q}\beta_j f_{n-j} \quad , \\ y' = f(t,y), t \geq 0, f_l = f(t_l, y_l)\end{gathered} \tag{2.1}$$

where α_j and β_j are the method's coefficients, h is a constant time step, and q is usually the order of the method. Equation (2.1) can represent, for example, the Trapezoidal, Backward Euler, Adams-Moulton, Adams-Bashforth, and BDF methods by selecting adequate values for α_j and β_j. In fact, the values of α_j and β_j can be chosen so that equation (2.1) meets any desired criteria. Generally the criteria to choose the method's coefficients α_j and β_j intends to maximize the order of the method p so that[8]

$$C_0 = C_1 = \ldots = C_p = 0, C_{p+1} \neq 0 \tag{2.2}$$

where C_0 to C_{p+1} are given by

$$\begin{gathered}C_0 = \sum_{j=0}^{q}\alpha_j \\ C_i = (-1)^i\left[\frac{1}{i!}\sum_{j=1}^{q} j^i\alpha_j + \frac{1}{(i-1)!}\sum_{j=0}^{q} j^{i-1}\beta_j\right], i = 1,2,\ldots.\end{gathered} \tag{2.3}$$

In this chapter a technique for choosing the values of α_j and β_j is described, so that the coefficients render a method with some stability and accuracy properties that are useful for the simulation of electric circuits. Before going to the mathematical derivation of the adjustable methods, a graphical explanation of the technique can be useful. Consider, for

[8] Reference [3], starting at page 126 has a more complete and detailed explanation.

example, the six discrete points represented by the diamonds in Figure 2.1. A polynomial of order five can be derived from those six points (using for example the Matlab function polyfit), from which the dashed curve of Figure 2.1 is obtained.

Now, let us estimate the derivative at the last discrete point. If the previous discrete point is used, the derivative results in the slope represented by line S0. This is equivalent to the well known Backward Euler method. If the derivative of the fifth order polynomial represented by the dashed line is used, the results is the slope represented by line S3, which is the same as the fifth order BDF. Intermediate points from the derived polynomial can be used as well, as depicted by the slopes of lines S1 and S2. In this chapter, the formulation for methods resulting in slopes similar to those of S1 and S2 is derived, and their stability and accuracy properties described.

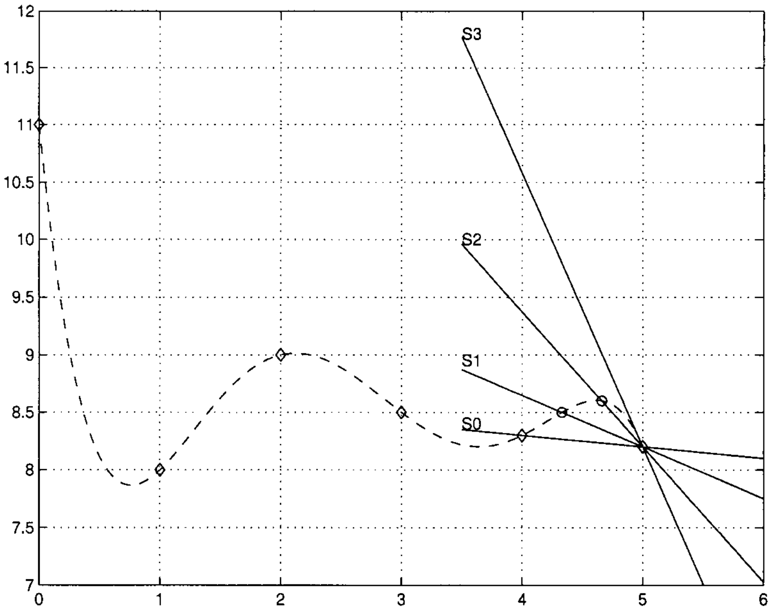


Figure 2.1 For a number of discrete points (diamonds) a fitting polynomial can be derived (dashed line). The derivative at the last discrete point can be approximated using different points, resulting in slopes of lines S0 (*k=0.0*) to S3 (*k=1.0*).

2.2 Derivation of the Adjustable Methods

Given a number of $n+1$ values $v(t)$, $v(t\text{-}h)$, ..., $v(t\text{-}nh)$, sampled at points t, $(t\text{-}h)$, ..., $(t\text{-}nh)$ we can assume there is a polynomial of degree n

$$v(t)=\sum_{m=0}^{n} a_m t^m \tag{2.4}$$

which produces the exact values at the selected sampled points. The coefficients of such polynomial can be found by solving the system of equations

$$\begin{aligned} v(t)&=\sum_{m=0}^{n} a_m t^m \\ v(t\text{-}h)&=\sum_{m=0}^{n} a_m (t-h)^m \\ &\;\;\cdot \qquad \cdot \\ v(t\text{-}nh)&=\sum_{m=0}^{n} a_m (t-nh)^m \end{aligned} \tag{2.5}$$

The solution of (2.5) produces coefficients that are a function of the sampled values $v(t)$, $v(t\text{-}h)$, ..., $v(t\text{-}nh)$, and the time step h such that

$$\begin{aligned} a_0 &= f(v(t), v(t-h), ..., v(t-nh), h) \\ a_1 &= f(v(t), v(t-h), ..., v(t-nh), h) \\ &\;\cdot \\ &\;\cdot \\ &\;\cdot \\ a_n &= f(v(t), v(t-h), ..., v(t-nh), h), \end{aligned} \tag{2.6}$$

from which the interpolating polynomial of equation (2.4) can be completed. If the derivative of equation (2.4) is taken, the well-known backward difference formula of order equal to the order of the interpolating polynomial is obtained:

$$\frac{dv(t)}{dt}=\sum_{m=1}^{n} m \cdot a_m t^m . \tag{2.7}$$

On the other hand, from the basic definition of the derivative,

$$\frac{dv(t)}{dt} = v^{'}(t) = \lim_{h\to 0}\frac{v(t)-v(t-h)}{h}, \qquad (2.8)$$

and replacing a variable *h* by *kh* where *h* is fixed and *k* is a real variable,

$$v^{'}(t) = \lim_{k\to 0}\frac{v(t)-v(t-kh)}{kh}. \qquad (2.9)$$

The delay term *v(t-kh)* in (2.9) can be represented by the interpolating polynomial of equation (2.4). Furthermore, it is possible, by selecting small, positive, and different from zero values for *k* and *h* to get a close approximate value of *v'(t)*

$$v^{'}(t) \approx \frac{v(t)-v(t-kh)}{kh}. \qquad (2.10)$$

By setting *k* to one, the backward Euler formula is obtained. Also, by setting *k* to zero the BDF formula whose degree corresponds to the used interpolating polynomial is obtained. Therefore, this technique permits the creation of a customized formula for the solution of differential equations which accuracy, stability, and frequency response can be adjusted by choosing a value of *k* between zero and one.

2.3 Example Method Derivation

The above-described methodology can be better illustrated with an example. Consider the third order polynomial

$$v(t)=a_3t^3+a_2t^2+a_1t+a_0 \qquad (2.11)$$

to approximate the value of *v'(t)* as a function of *k*. To find the coefficients of this polynomial four values are required, *v(t), v(t-h), v(t-2h),and v(t-3h)*:

$$\begin{aligned} v(t) &= a_3t^3 + a_2t^2 + a_1t + a_0 \\ v(t\text{-}h) &= a_3(t\text{-}h)^3 + a_2(t\text{-}h)^2 + a_1(t\text{-}h) + a_0 \\ v(t\text{-}2h) &= a_3(t\text{-}2h)^3 + a_2(t\text{-}2h)^2 + a_1(t\text{-}2h) + a_0 \\ v(t\text{-}3h) &= a_3(t\text{-}3h)^3 + a_2(t\text{-}3h)^2 + a_1(t\text{-}3h) + a_0 \end{aligned} \; . \tag{2.12}$$

Solving this system of equations, the coefficients a_3, a_2, a_1, and a_0 are given by

$$\begin{aligned} a_3 &= \frac{-v(t-3h)+3v(t-2h)-3v(t-h)+v(t)}{6h^3} \\ a_2 &= \frac{-v(t-3h)+4v(t-2h)-5v(t-h)+2v(t)}{2h^2} \\ a_1 &= \frac{-2v(t-3h)+9v(t-2h)-18v(t-h)+11v(t)}{6h} \\ a_0 &= v(t) \end{aligned} \; . \tag{2.13}$$

What is needed now is $v(t\text{-}kh)$. This term comes from the interpolating polynomial of equation (2.4) and the calculated values of a_3, a_2, a_1, and a_0, as follows

$$\begin{aligned} v(t-kh) = &-\frac{1}{6}(-v(t-3h)+3v(t-2h)-3v(t-h)+v(t))k^3 \\ &+\frac{1}{2}(-v(t-3h)+4v(t-2h)-5v(t-h)+2v(t))k^2 \\ &-\frac{1}{6}(-2v(t-3h)+9v(t-2h)-18v(t-h)+v(t))k+v(t) \end{aligned} \; . \tag{2.14}$$

What is now left is to evaluate the approximate derivative using equation(2.10):

$$\begin{aligned} v'(t) \approx (&\frac{1}{6}(-v(t-3h)+3v(t-2h)-3v(t-h)+v(t))k^3 \\ &-\frac{1}{2}(-v(t-3h)+4v(t-2h)-5v(t-h)+2v(t))k^2 \\ &+\frac{1}{6}(-2v(t-3h)+9v(t-2h)-18v(t-h)+v(t))k)/kh \end{aligned} \; . \tag{2.15}$$

If terms are collected from equation (2.15) the result is

$$\begin{aligned} v'(t) \approx ((&\frac{1}{6}k^2-k+\frac{11}{6})v(t)+(-\frac{1}{2}k^2+\frac{5}{2}k-3)v(t-h)+ \\ &(\frac{1}{2}k^2-2k+\frac{3}{2})v(t-2h)+(-\frac{1}{6}k^2+\frac{1}{2}k-\frac{1}{3})v(t-3h))/h \end{aligned} \; . \tag{2.16}$$

Equation (2.16) is the difference equation as a function of k. It is more convenient, especially for higher order interpolating polynomials, to write equation (2.16) as

$$v'(t) \approx \frac{1}{h}\sum_{m=0}^{n} k_m v(t-mh), \tag{2.17}$$

where n is the order of the used polynomial, in this case $n=3$ and

$$\begin{aligned} k_0 &= (\frac{1}{6}k^2 - k + \frac{11}{6}), \\ k_1 &= (-\frac{1}{2}k^2 + \frac{5}{2}k - 3), \\ k_2 &= (\frac{1}{2}k^2 - 2k + \frac{3}{2}), \\ k_3 &= (-\frac{1}{6}k^2 + \frac{1}{2}k - \frac{1}{3}). \end{aligned} \tag{2.18}$$

Notice that when k=1 equation (2.17) becomes

$$v'(t) \approx \frac{v(t)-v(t-h)}{h}, \tag{2.19}$$

which is the Backward Euler formula, similar to equation (1.6). On the other hand, when k=0, the third order BDF is obtained as

$$v'(t) \approx \frac{1}{h}(\frac{11}{6}v(t) - 3v(t-h) + \frac{3}{2}v(t-2h) - \frac{1}{3}v(t-3h)) \tag{2.20}$$

2.4 Magnitude and Phase Distortion

Figure 2.2 shows the magnitude and phase distortion introduced by equation (2.17), computed as described in section 1.2.1. As expected, setting k to one results in the same distortion as the Backward Euler formula shown in Figure 1.4. Similarly, setting k to zero results in the distortion introduced by the third order BDF. Any intermediate values will result in a method with different distortion characteristics. In Chapter 3 of this work, this ability to select the distortion characteristic of the method by adjusting k is used to find methods tuned for a particular frequency, so that the resulting formula produces exact results at that frequency.

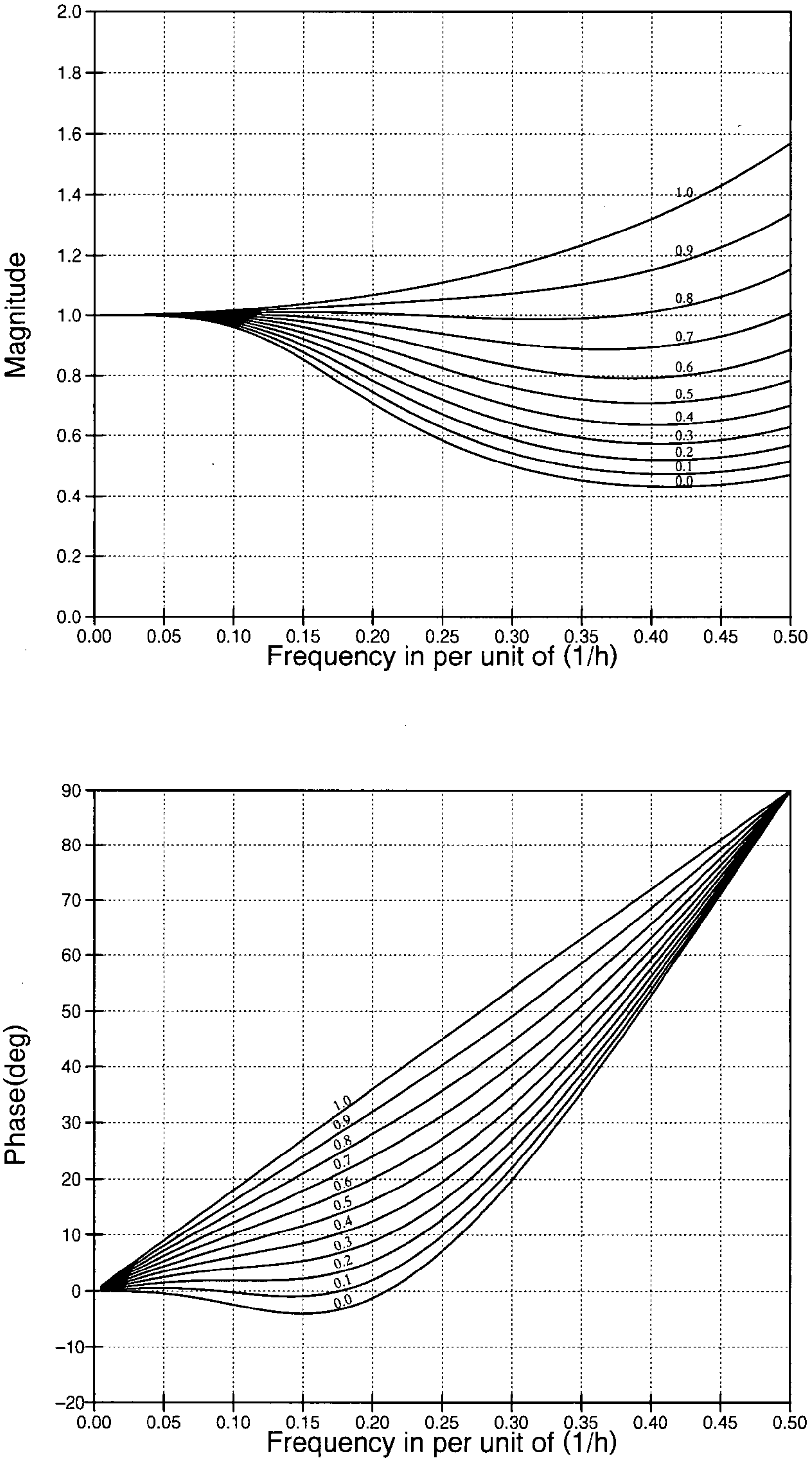


Figure 2.2 Magnitude distortion and phase error introduced by equation (2.17) for the indicated values of *k*.

2.5 Stability

Although the main focus of this thesis is in the tuned difference equations of next chapter and their applications, it is interesting to see how it is possible to adjust a difference equation so it becomes A-stable. For example, Figure 2.3 shows how the stability zone of equation (2.17) changes with the value of *k* computed as described in section 1.2.1. For instance, it is possible by carefully selecting the value of *k* to obtain a formula which is A-stable. It so happens that this the case for equation (2.17) when *k* is set to approximately 0.14, resulting in the difference equation

$$v'(t) \approx \frac{1}{h}(1.6966v(t)-2.6598v(t-h)+1.2298v(t-2h)-0.2666v(t-3h)). \qquad (2.21)$$

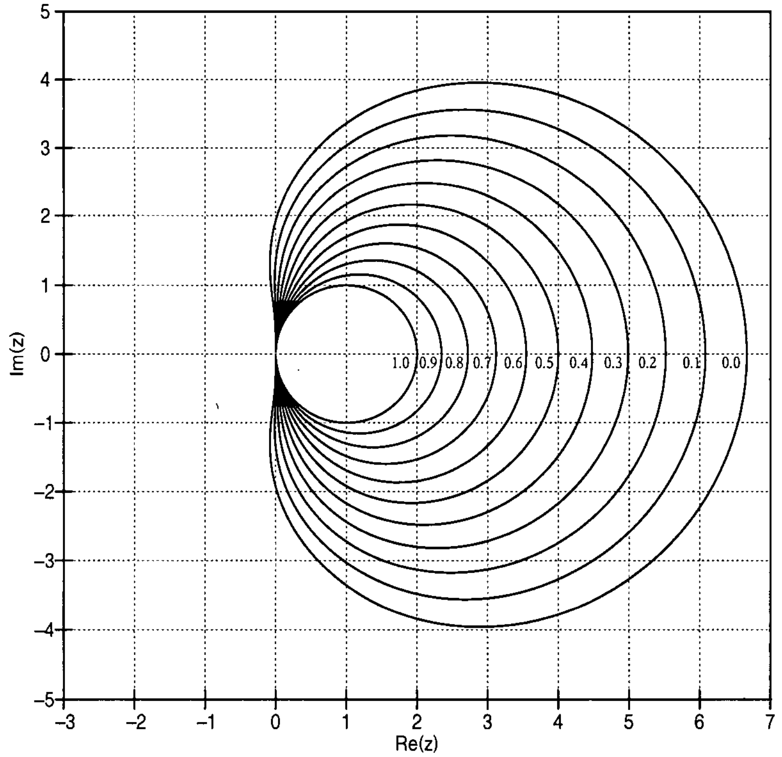


Figure 2.3 Stability zone for equation (2.17) for the indicated values of *k*.

It is interesting to see the distortion introduced by equation (2.21). The solid line of Figure 2.5 corresponds to the magnitude and phase characteristics of equation (2.21) while the dashed line corresponds to the trapezoidal rule. The main characteristics of the proposed method are A-stability and low error at low frequencies. The properties of the method are very similar of those of the trapezoidal rule up to one tenth of the maximum frequency *1/h*. For instance, when the circuit of Figure 2.4 is simulated using the trapezoidal rule, the output voltage of phase 'a' of Bus_1 shows the numerical oscillations of Figure 2.6, after circuit breaker CB1 is tripped[9]. On the other hand, when the circuit is simulated with equation (2.21), the output voltage at phase 'a' of Bus_1 is free of such numerical oscillations, as shown in Figure 2.7.

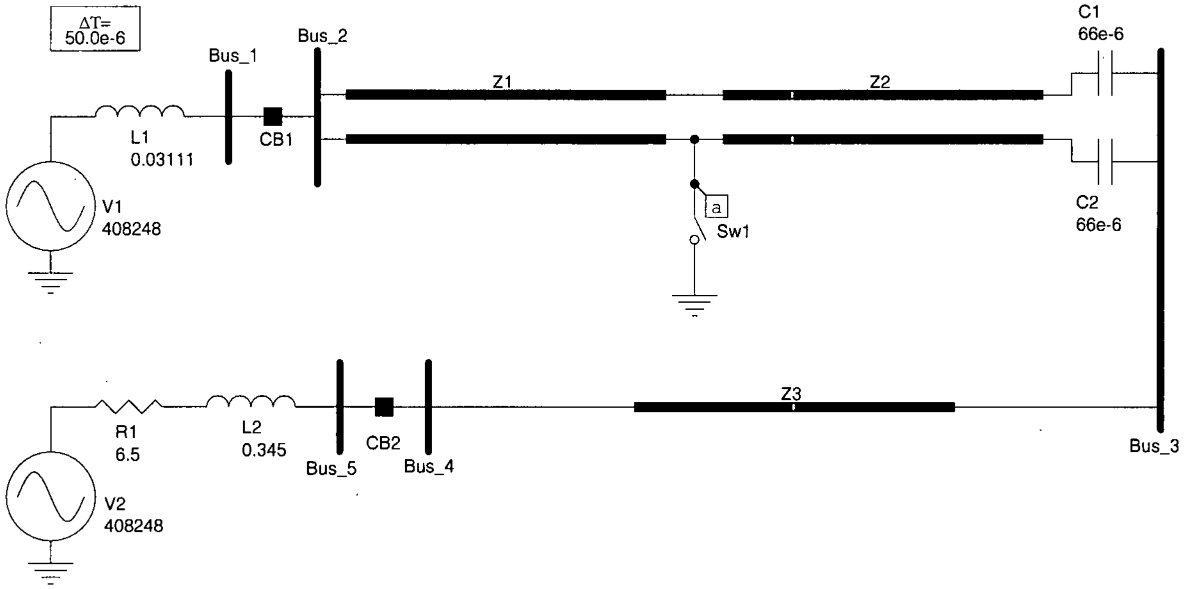


Figure 2.4 Circuit used to test equation (2.21)

[9] In Microtran, the UBC version of the EMTP, numerical oscillations are suppressed using the critical damping algorithm (CDA) as described in [2].

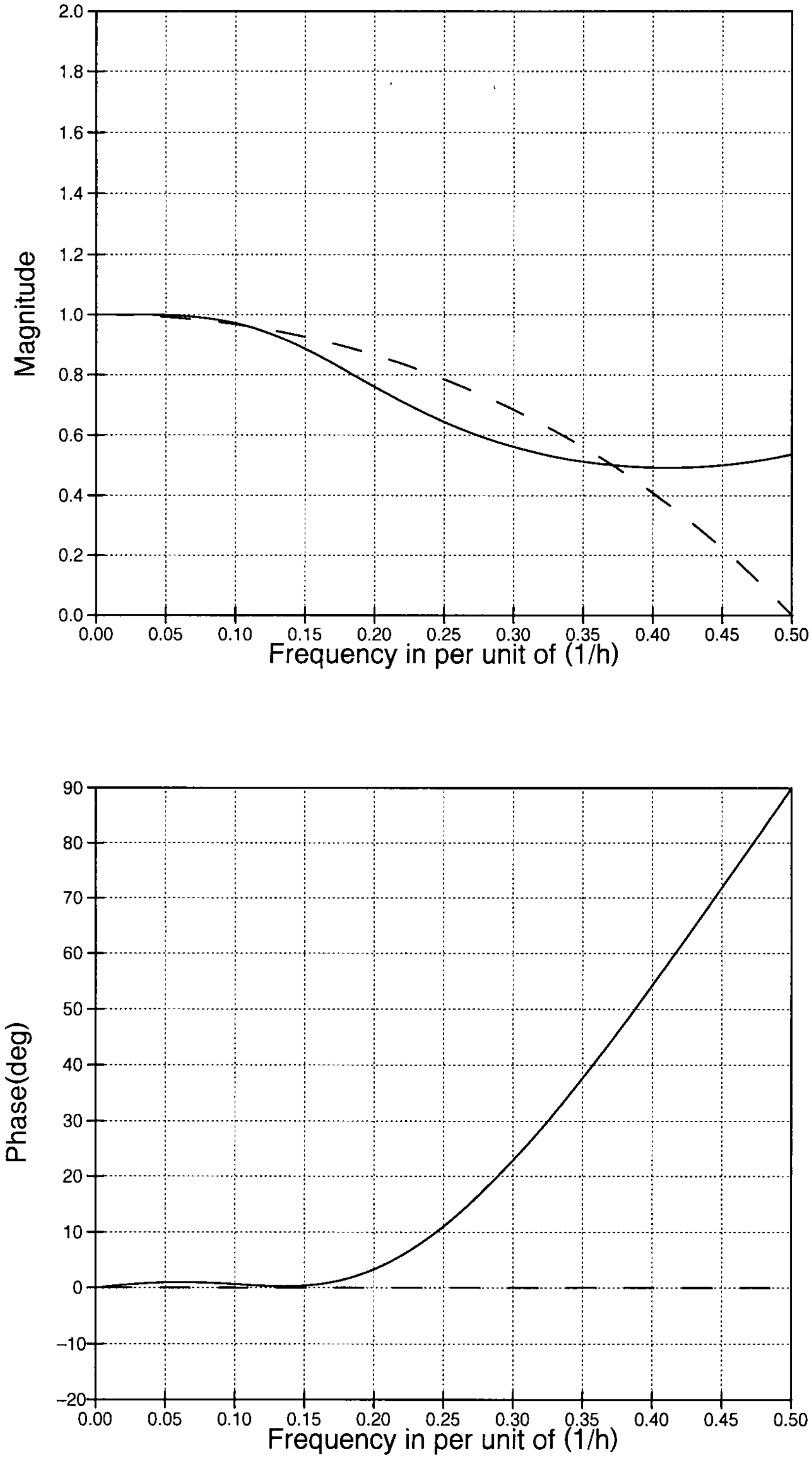


Figure 2.5 Magnitude distortion and phase error introduced by equation (2.21). The dashed line corresponds to the trapezoidal rule, included here for comparison purposes.

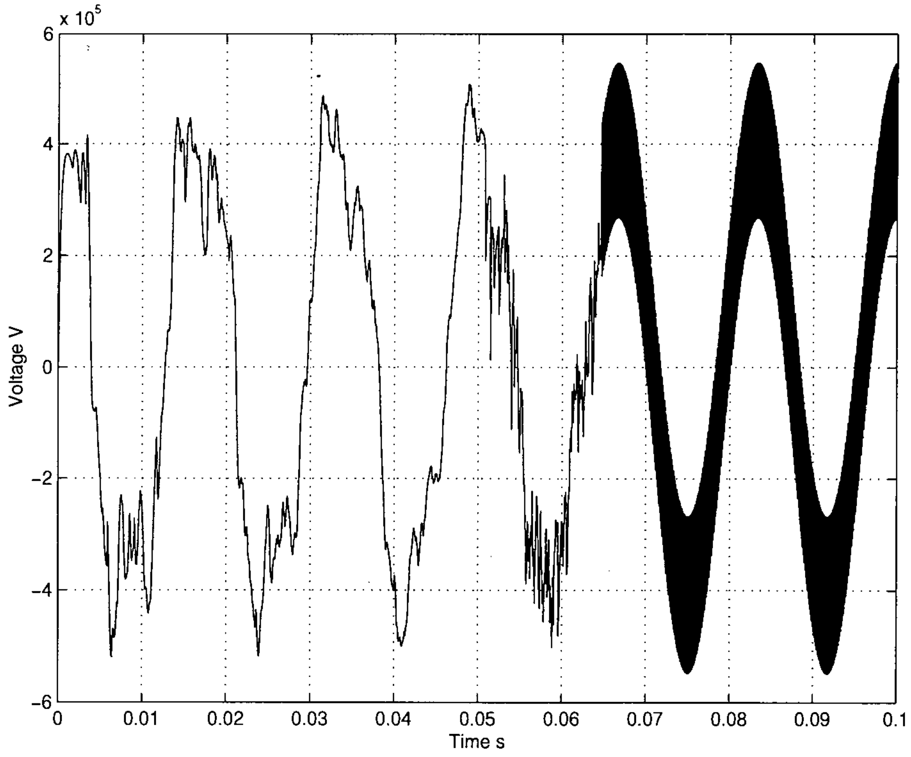


Figure 2.6 Voltage in phase 'a' of Bus_1 when the trapezoidal rule is used to simulate the circuit of Figure 2.4. Notice the numerical oscillations when CB1 is tripped at 60 ms.

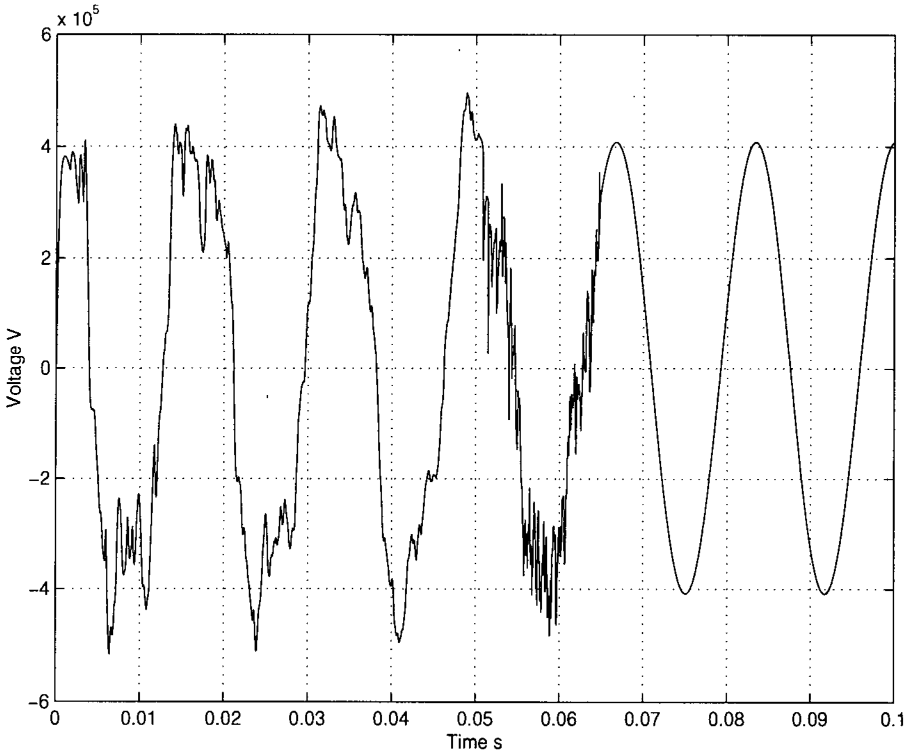


Figure 2.7 Voltage in phase 'a' of Bus_1 when equation (2.21) is used to simulate the circuit of Figure 2.4.

CHAPTER 3: Tuned Time-Domain Linear AC Steady State Solutions

3.1 Introduction

Two possible techniques for the solution of linear AC steady state circuits are phasor analysis and time-domain simulations using small time-steps. An alternative to these techniques, discussed in this thesis, is the adjustable methods described in the previous chapter, which can be tuned-up to produce exact results at one frequency. To illustrate this finding, consider for example the magnitude and phase distortion introduced by the adjustable method of order five of Figure 3.1 plotted for k=0 to 0.4 in 0.1 increments.

Notice from Figure 3.1 that the magnitude plot, for any of the displayed values of k, starts at one, raises slightly, and then decreases, crossing one again. Similarly, for the phase plot, the phase distortion starts at zero, but before reaching the final value of 90°, it crosses zero twice. Furthermore, notice that for $k=0.4$[10] the magnitude plot crosses one and the phase plot crosses zero for almost the same value ω of the normalized frequency[11]! Not only they are exact, but also the time step of these methods is considerably larger than that of traditional non-tuned time-domain methods. In this chapter, the derivation of tuned methods is presented, as well as their application for the simulation of one-frequency linear steady state electric power circuits. Later, in Chapter 4, tuned methods are used to solve the non-linear static power flow problem, and in Chapter 5 to solve the dynamic power flow problem.

[10] A more accurate value from Table 3.1 is k=0.4029346295063755.

[11] The normalized frequency is the horizontal axis of Figure 3.1. It is defined as the product ω=f h.

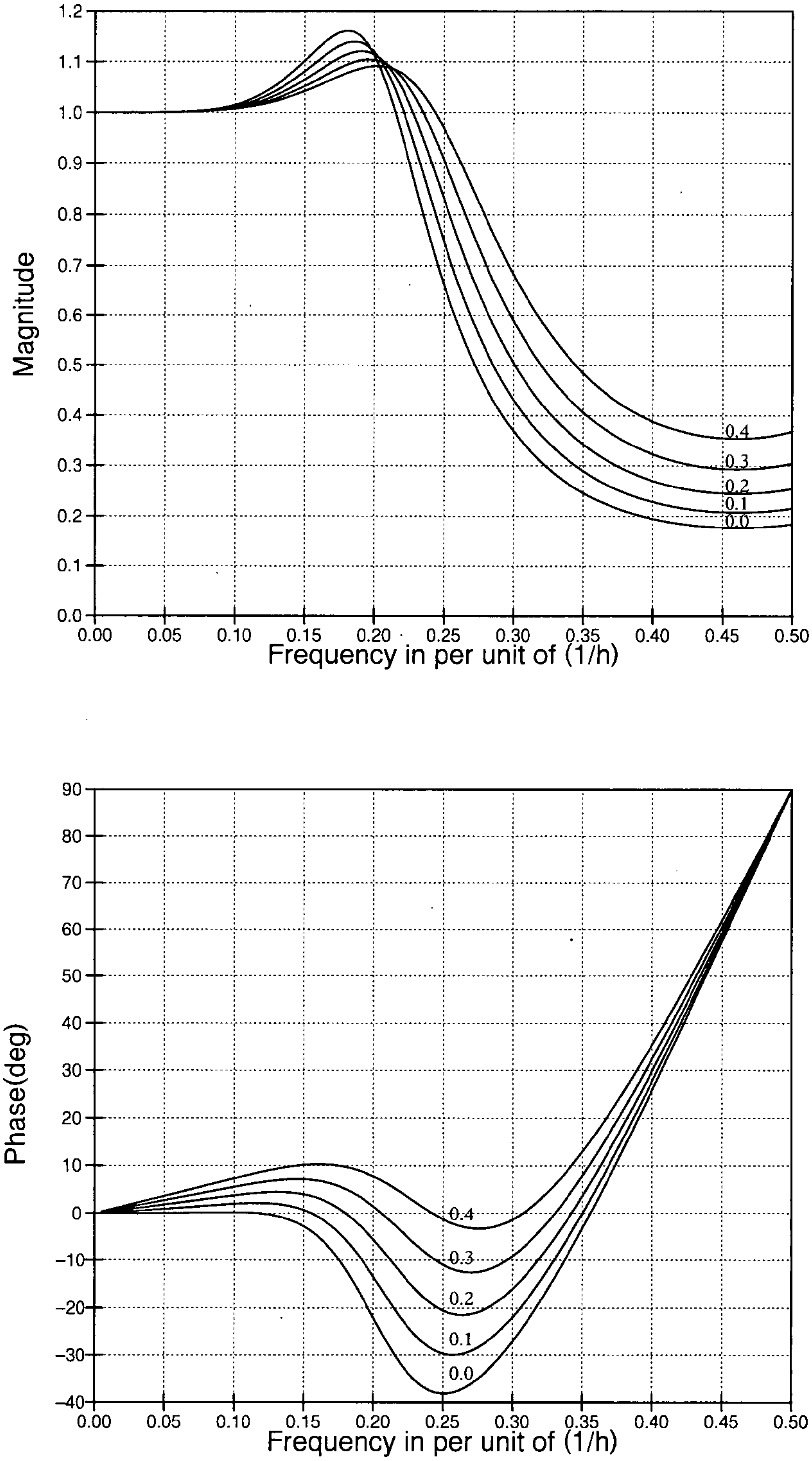


Figure 3.1 Magnitude distortion and phase error introduced by the adjustable method of order 5 for *k*=0.0 to 0.4 in 0.1 increments.

3.2 Tuning the Adjustable Methods

For polynomials of orders equal or greater than four, we can compute at least one value of k and one value of the normalized angular frequency ω for which the magnitude distortion is one and the phase error is zero, for ω different from zero. Since the normalized frequency is directly proportional to the time-step h (note 11, page 34), once ω is found h is known. The values of k and ω can be found from the z-domain transfer function of the corresponding difference equation. For BDF-like methods, the transfer function of the difference equation has the form

$$H(\omega)=\frac{j\omega}{\sum_{m=0}^{n} k_m e^{-jm\omega}},\quad j=\sqrt{-1}\,. \tag{3.1}$$

where n is the order of the interpolating polynomial and k_m are the difference equation coefficients as a function of k, similar to those of equation (2.18). Also, since

$$e^{-jm\omega}=cos(m\omega)-j\cdot sin(m\omega)\,, \tag{3.2}$$

equation (3.1) becomes

$$H(\omega)=\frac{j\omega}{\sum_{m=0}^{n} K_m(cos(m\omega)-j\cdot sin(m\omega))}\,. \tag{3.3}$$

Assuming that at one particular ω there is no distortion, that is $H(\omega)=1$, we have

$$\sum_{m=0}^{n} K_m(cos(m\omega)-j\cdot sin(m\omega))=j\omega\,. \tag{3.4}$$

Separating real and imaginary parts, a system of two equations and two unknowns is formed,

$$\begin{cases} \sum_{m=0}^{n} K_m \cos(m\omega) = 0 = g_1(k,\omega) \\ \omega + \sum_{m=0}^{n} K_m \sin(m\omega) = 0 = g_2(k,\omega) \end{cases} . \tag{3.5}$$

This system of equations can be solved using Newton-Raphson iterations:

$$\begin{bmatrix} \frac{\partial g_1}{\partial k} & \frac{\partial g_1}{\partial \omega} \\ \frac{\partial g_2}{\partial k} & \frac{\partial g_2}{\partial \omega} \end{bmatrix} \begin{bmatrix} \Delta k \\ \Delta \omega \end{bmatrix} = \begin{bmatrix} \Delta g_1 \\ \Delta g_2 \end{bmatrix}, \tag{3.6}$$

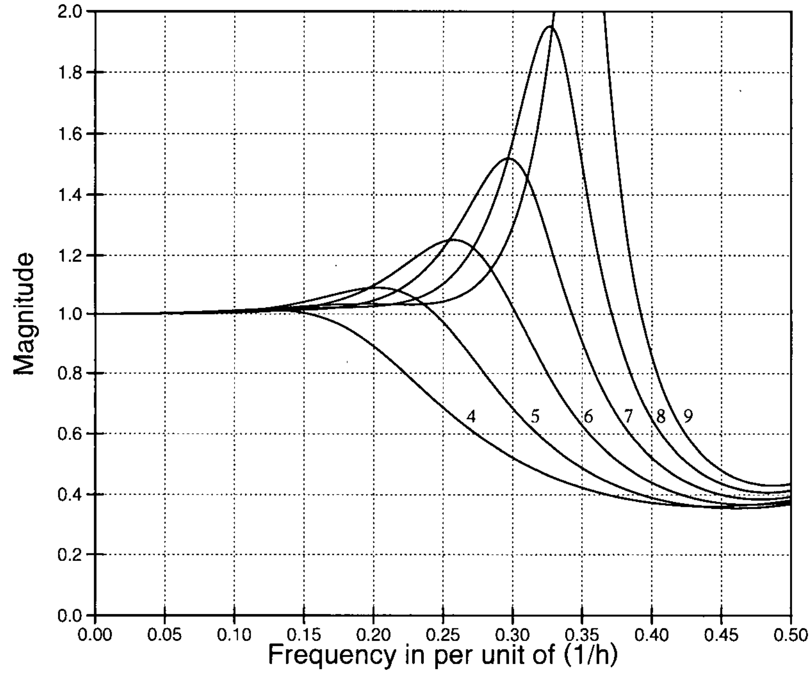
where

$$\begin{aligned} \frac{\partial g_1}{\partial k} &= \sum_{m=0}^{n} \frac{dk_m}{dk} \cos(m\omega) \\ \frac{\partial g_2}{\partial k} &= \sum_{m=0}^{n} \frac{dk_m}{dk} \sin(m\omega) \\ \frac{\partial g_1}{\partial \omega} &= -\sum_{m=0}^{n} k_m m \cdot \sin(m\omega) \\ \frac{\partial g_2}{\partial \omega} &= 1 + \sum_{m=0}^{n} k_m m \cdot \cos(m\omega) \end{aligned} . \tag{3.7}$$

Solving the system of equations (3.5) for difference equations obtained with polynomials of orders four to nine results in the values of Table 3.1.

n	k	ω
4	0.1995183215439153	0.9648838221480567
5	0.4029346295063755	1.5328796706951471
6	0.5577432820878876	1.9006879375994667
7	0.6761264770927172	2.1526576264068528
8	0.7672635252838457	2.3321297134871948
9	0.8369482736463172	2.4636521899340823

Table 3.1 Tuned values of k and ω for the difference equations obtained with polynomials of orders 4 to 9.



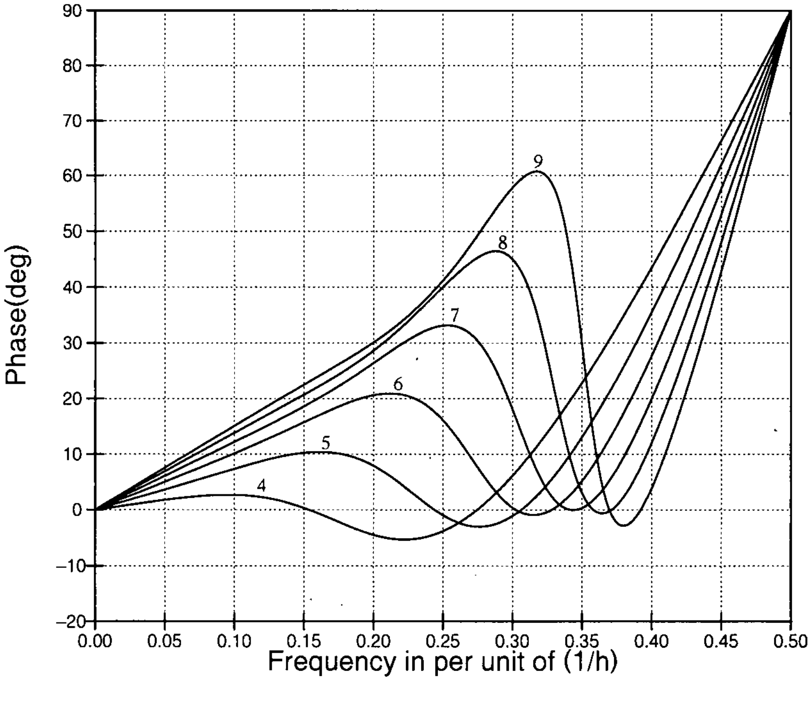


Figure 3.2 Magnitude distortion and phase error of tuned difference equations derived with polynomials of orders 4 to 9 and the values of *k* in Table 3.1.

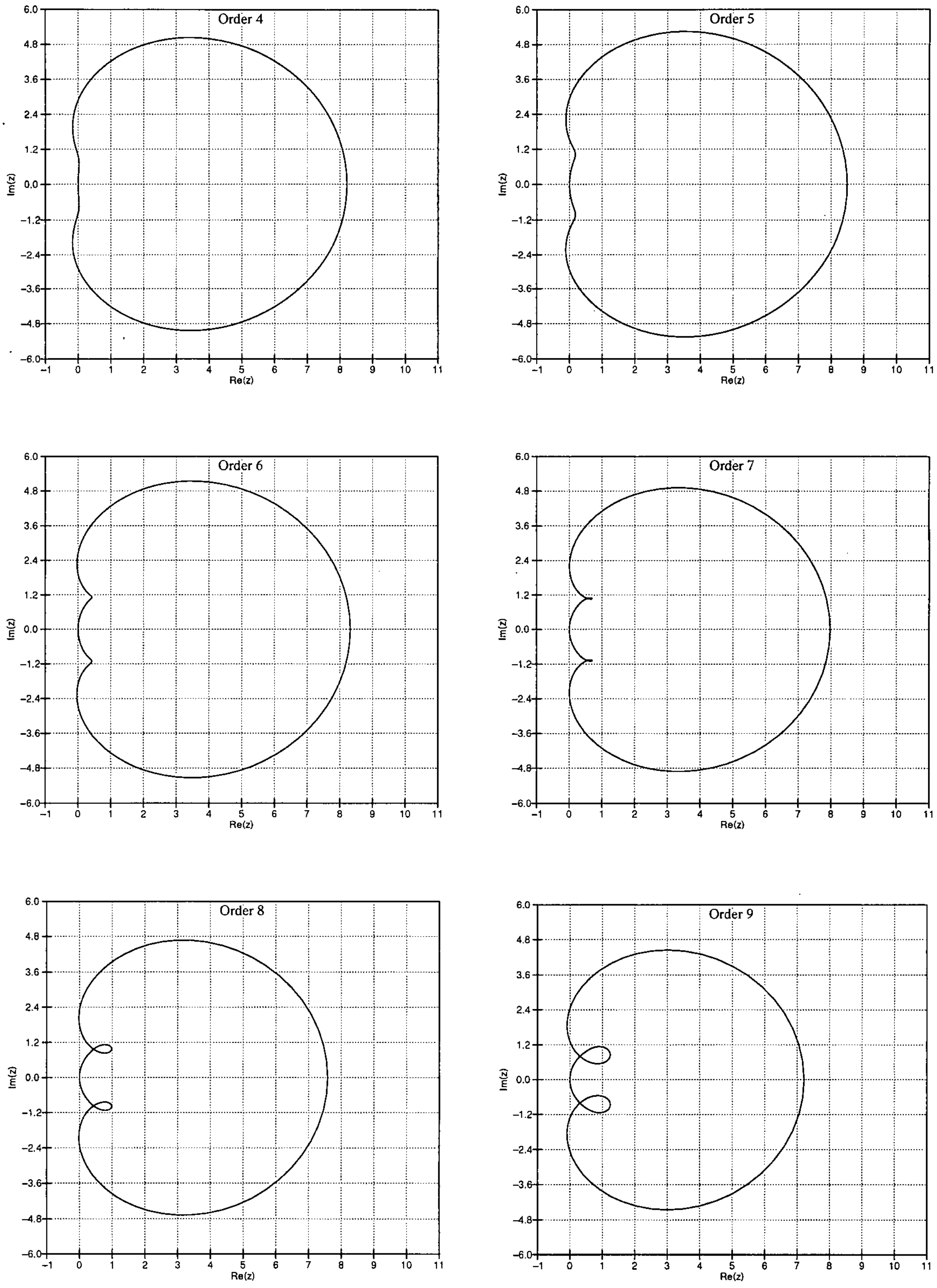


Figure 3.3 Loci of the roots for the tuned methods of orders four to nine.

From the tuned value of ω, the value of the time-step h can be computed using

$$h = \frac{\omega}{\omega_s}, \quad \omega_s = 2\pi f_s, \tag{3.8}$$

where f_s is the frequency at which the difference equation is required to produce exact results. Figure 3.2 shows the magnitude and phase characteristics of the tuned difference equations obtained with the values of k in Table 3.1 for polynomials of orders four to nine, while Figure 3.2 shows the stability regions for such methods, both obtained as described in section 1.2.1.

From Figure 3.3 it can be observed that all the tuned methods are reasonably stable, but the tuned difference equation obtained with a seventh order interpolating polynomial is almost A-stable[12] [5] as shown in Table 3.2. The difference equation for this curve is

$$\begin{aligned} v'(t) \approx (&1.338v(t)-2.068v(t-h)+1.518v(t-2h)-1.441v(t-3h) \\ &+1.007v(t-4h)-0.465v(t-5h)+0.126v(t-6h)-0.015v(t-7h))/h. \end{aligned} \tag{3.9}$$

The coefficients of equation (3.9) have been rounded to three decimal digits. More accurate coefficients or coefficients for different order interpolating polynomials can be obtained with the Maple 6 script of Figure 3.4.

n	α(deg)
4	84.65845
5	86.96453
6	89.07581
7	89.98968
8	89.38110
9	87.13993

Table 3.2 The tuned methods obtained with interpolating polynomials of order 4 to 9 are A(α)-stable. For the method obtained with a seventh order interpolating polynomial is almost A-stable.

[12] α for an A-stable method is 90°.

```
# ctune.m: Find the tuned coefficients of the difference
# equation using a polynomial of order n.

restart;
Digits:=16:
n:=7: # this is the order of the polynomial to use.
v:=array(0..n):
a:=array(0..n):
equ:=array(0..n):
k:=array(0..n):

# Create a polynomial of the desired order
i:='i': ipoly:=sum((a[n-i]*t^(n-i)), i=0..n);

# Create the system of equations
for i from 0 to n do
   t:=-i*h;
   equ[i]:=v[i]=ipoly;
end do:

# Solve the system of equations
assign(solve({seq(equ[i],i=0..n)},{seq(a[i],i=0..n)})):

# Construct the difference equation
t:=-K*h:
diffeq:=simplify((v[0]-ipoly)/K*h):


# From the difference equation get the formulas for k[0..n]
for i from 0 to n do
   v[i]:=0:
end do:
for i from 0 to n do
   v[i]:=1;
   k[i]:=sort(simplify(diffeq/h));
   v[i]:=0;
end do;

# Display k[0..n] as a function of K.
for i from 0 to n do
   k[i]:=k[i];
end do;

# Compute the tuned values of K and w.
i:='i': eqR:=sum(k[i]*cos(i*w0), i=0..n)=0:
i:='i': eqI:=w0+sum(k[i]*sin(i*w0), i=0..n)=0:
assign(fsolve({eqR, eqI}, {K, w0}, {K=0.15..1.0, w0=0.9..evalf(Pi)}));

# Display the results
K:=K; w0:=w0;
for i from 0 to n do
   k[i]:=k[i];
end do;
```

Figure 3.4 This Maple 6 script can be use to computed the coefficients of the adjustable difference equations, as well as to find their tuned values for interpolating polynomials of order ≥ 4.

3.3 Circuit Simulation Example

Before using equation (3.9) to simulate a circuit, the value of *h*, the time-step, must be computed. Suppose a circuit with a steady state solution at 60 Hz has to be solved. Using equation (3.8) for $n=7$ results in

$$h = \frac{2.1526576264068528}{2\pi 60} = 5.7101017004519526ms\,. \tag{3.10}$$

The circuit of Figure 3.5 was simulated using difference equation (3.9) and the time step given by equation (3.10). The simulation was started using AC steady state initial conditions computed with phasor analysis. The voltage across inductor L1 is presented in Figure 3.6. The solid line represents the voltage obtained using phasor analysis, while the scattered dots are the voltages obtained after a time domain simulation performed with equation (3.9). The dots match perfectly with the corresponding values obtained using standard phasor analysis.

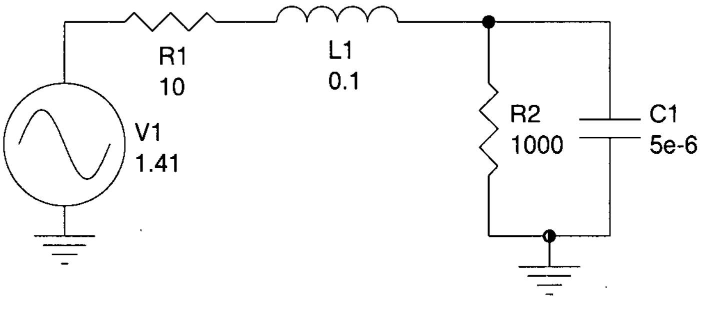


Figure 3.5 Circuit used to test equation (3.9).

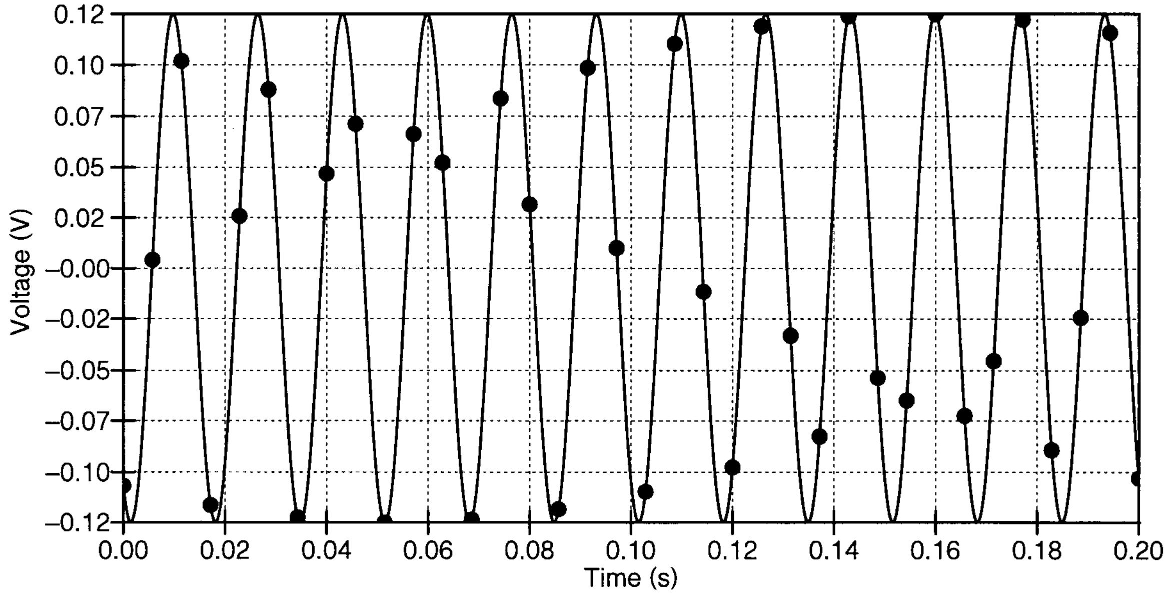


Figure 3.6 The solid line is the voltage in the inductor of Figure 3.5 obtained using phasor analysis. The dots are the values obtained when simulating the same circuit using the tuned difference equation (3.9).

3.4 Reducing the Number of Circuit Nodes

It will be shown later in this chapter, that the cost of the tuned time-domain simulation is proportional to the cube of the number of nodes in the circuit to be solved. Therefore, it is important to reduce the number of nodes in the circuit. The most important reduction is achieved when series passive elements are discretized together so that the internal nodes connecting them are unnecessary. In particular, the series connection of resistors and inductors (RL) as well as resistors and capacitors (RC) can be reduced. This is also necessary for the impedances represented with complex numbers found in phasor analysis, which need to be converted either to a RL or RC equivalent circuit for time-domain simulations. For instance, a complex impedance of the form $R+jX$ is converted to a RL series circuit, where $L=X/(2\pi f)$. If the imaginary part of the complex impedance is negative, that is R-jX, then it is converted to a RC series circuit, where $C=1/(2\pi fX)$.

The RL and RC circuits' discrete representation is similar to that of Figure 1.2, and depicted in Figure 3.7. For the RL series connection, the voltage of the series is given by

$$v(t)=Ri(t)+L\frac{di(t)}{dt}. \tag{3.11}$$

Replacing the derivative with equation (2.17) results in

$$v(t)=Ri(t)+\frac{L}{h}\sum_{m=0}^{n}k_m i(t-mh). \tag{3.12}$$

Collecting the first term of the current $i(t)$ results in the formula we are looking for

$$v(t)=\left(R+\frac{Lk_0}{h}\right)i(t)+\frac{L}{h}\sum_{m=1}^{n}k_m i(t-mh), \tag{3.13}$$

from which the values of the equivalent circuit are

$$\begin{aligned} g_{RL} &= \left(R+\frac{Lk_0}{h}\right)^{-1} \\ h_{RL} &= -\frac{g_{RL}L}{h}\sum_{m=1}^{n}k_m i(t-mh) \end{aligned}. \tag{3.14}$$

For the tuned time-domain steady state solution of a given circuit, the coefficients k_0 to k_n of equation (3.14) would be computed using the values of k of Figure 3.1.

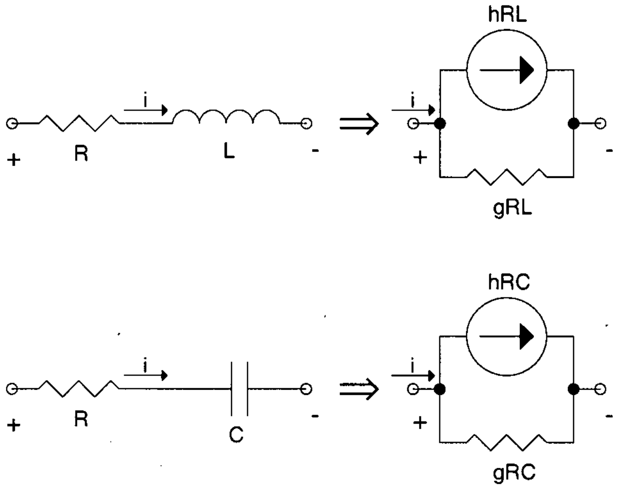


Figure 3.7 Discrete equivalents for RL and RC circuits.

Similarly for the RC circuit the voltage across the series combination is given by:

$$v(t) = Ri(t) + \frac{1}{C}\int i(t)dt\,. \tag{3.15}$$

Taking the derivative of equation (3.15) results in

$$\frac{dv(t)}{dt} = R\frac{di(t)}{dt} + \frac{1}{C}i(t)\,. \tag{3.16}$$

Replacing the derivatives with equation (2.17) results in

$$\frac{1}{h}\sum_{m=0}^{n} k_m v(t-mh) = \frac{R}{h}\sum_{m=0}^{n} k_m i(t-mh) + \frac{1}{C}i(t) \tag{3.17}$$

Collecting the *v(t)* and *i(t)* terms from equation (3.17) yields

$$k_0 v(t) + \sum_{m=1}^{n} k_m(v(t-mh) - Ri(t-mh)) = \left(k_0 R + \frac{h}{C}\right)i(t)\,, \tag{3.18}$$

From which the discrete equivalent circuit values are:

$$\begin{aligned} g_{RC} &= k_0\left(k_0 R + \frac{h}{C}\right)^{-1} \\ h_{RC} &= \frac{g}{k_0}\sum_{m=1}^{n} k_m(v(t-mh) - Ri(t-mh)) \end{aligned} \tag{3.19}$$

3.5 Estimating Magnitude and Phase

To compute the AC steady state of a circuit using any of the tuned difference equations such as equation (3.9), the magnitude and phase of each node in the circuit must be calculated using the discrete points resulting from the transient simulation. Since equation (3.9) is not only exact for the tuned frequency but also for DC, it is possible, by using three previously computed points to match a function of the form

$$v(t) = V_{dc} + \sqrt{2}V_{rms}\cos(2\pi ft + \theta) \tag{3.20}$$

by solving the system of equations

$$\begin{cases} v(t) = V_{dc} + V_P\cos(\omega t + \theta) \\ v(t-h) = V_{dc} + V_P\cos(\omega(t-h) + \theta) \\ v(t-2h) = V_{dc} + V_P\cos(\omega(t-2h) + \theta) \end{cases}, \tag{3.21}$$

where v*(t), v(t-h),* and *v(t-2h)* are the simulation results at times *t*, *t-h*, and *t-2h*, and

$$\begin{matrix} V_P = \sqrt{2}V_{rms} \\ \omega = 2\pi f \end{matrix}. \tag{3.22}$$

Also, by making the following substitutions in the systems of equations (3.21)

$$\begin{matrix} \cos(\omega(t-h)+\theta) = \cos(\omega t+\theta)\cos(\omega h) + \sin(\omega t+\theta)\sin(\omega h) \\ \cos(\omega(t-2h)+\theta) = \cos(\omega t+\theta)\cos(\omega 2h) + \sin(\omega t+\theta)\sin(\omega 2h), \\ V_{dc} = v(t) - V_P\cos(\omega t+\theta) \end{matrix} \tag{3.23}$$

we obtain in the reduced system of equations

$$\begin{cases} v(t-h) = v(t) - V_P\cos(\omega t+\theta) + V_P(\cos(\omega t+\theta)\cos(\omega h) + \sin(\omega t+\theta)\sin(\omega h) \\ v(t-2h) = v(t) - V_P\cos(\omega t+\theta) + V_P(\cos(\omega t+\theta)\cos(\omega 2h) + \sin(\omega t+\theta)\sin(\omega 2h)) \end{cases}. \tag{3.24}$$

Collecting terms in (3.24) results in

$$\begin{cases} v(t) - v(t-h) = V_P\cos(\omega t+\theta)(1+\cos(\omega h)) + V_P\sin(\omega t+\theta)\sin(\omega h) \\ v(t) - v(t-2h) = V_P\cos(\omega t+\theta)(1+\cos(\omega 2h)) + V_P\sin(\omega t+\theta)\sin(\omega 2h) \end{cases}. \tag{3.25}$$

Applying this substitutions to (3.25)

$$N_1 = V_P\,sin(\omega t + \theta), N_2 = V_P\,cos(\omega t + \theta), \tag{3.26}$$

yields the linear system of equations

$$\begin{cases} v(t) - v(t-h) = N_1\,sin(\omega h) + N_2(1 + cos(\omega h)) \\ v(t) - v(t-2h) = N_1\,sin(\omega 2h) + N_2(1 + cos(\omega 2h)) \end{cases}. \tag{3.27}$$

Solving the system of equation (3.27) for N_1 and N_2 allows for the computation of V_p, θ, and V_{dc}, as

$$\begin{aligned} &V_P = \sqrt{N_1^{\,2} + N_2^{\,2}} \\ &\theta = a\,tan(\frac{N_1}{N_2}) - \omega t \\ &V_{dc} = v(t) - V_P\,cos(\omega t + \theta) \end{aligned} \tag{3.28}$$

3.6 Tuned Time-Domain Steady State Simulations

The algorithm for the simulation of an electric circuit using the tuned difference equations is similar to the one used in the Electromagnetics Transient Program EMTP described in [30]. The flow chart of Figure 3.8 shows the program sequence, which includes most of the steps presented in section 1.2.1. The main difference, besides the use of the tuned formulas, is the computation of the AC magnitudes and phases for the node voltages as described in section 3.5. It is the change of the AC magnitudes and phases after each time step what is used to terminate the simulation. If the desired tolerance is not achieved in a preset number of steps, the simulation is terminated and an error message is displayed.

An electric power simulation program, which follows the flow chart of Figure 3.8, was developed using the C programming language. The program is not only capable of solving the linear steady state of a circuit, but also the static and dynamic power flow as well. Also, the discretization techniques used in the program are selectable, so it was possible to test the tuned difference equations generated with different interpolating polynomials.

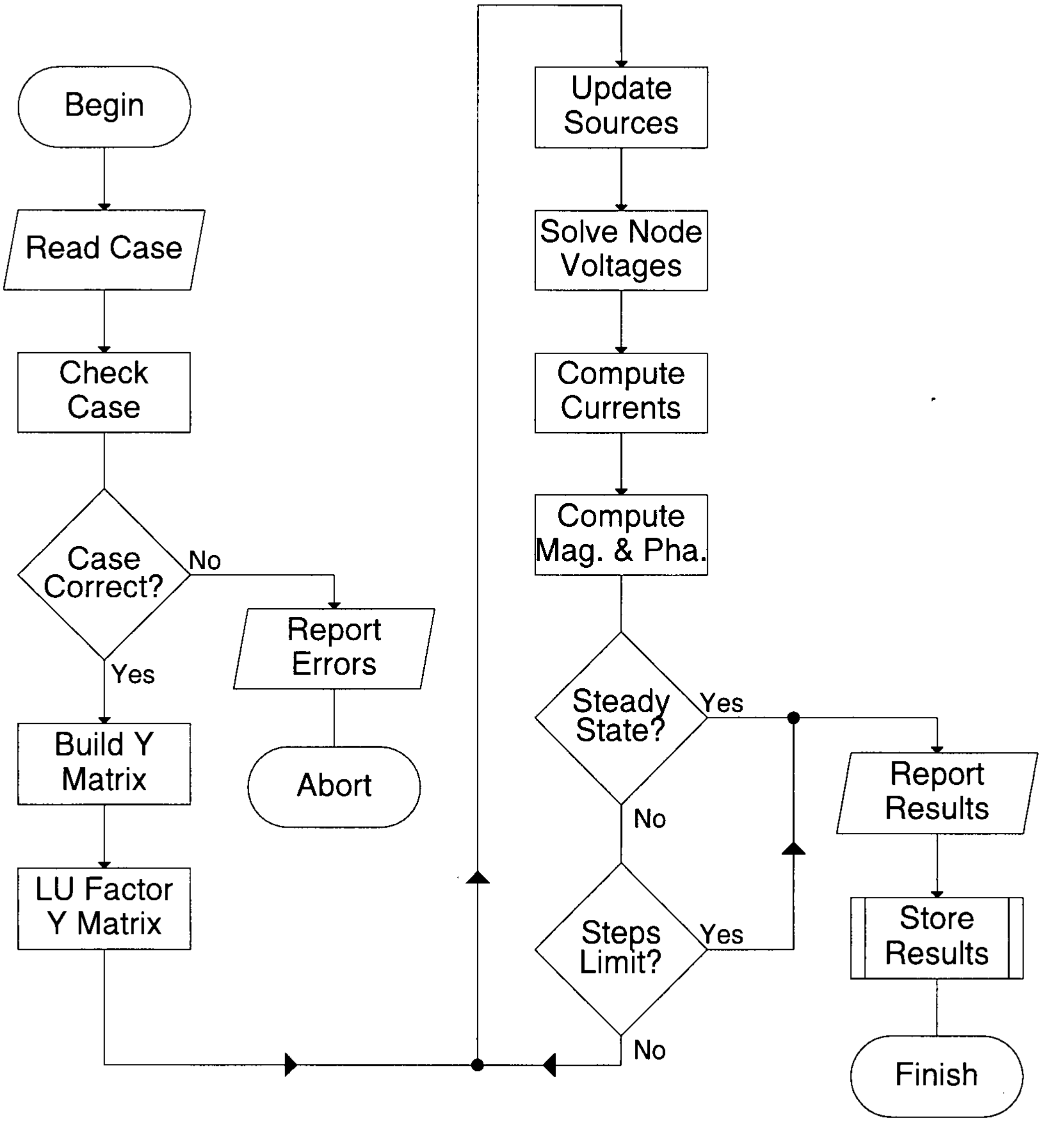


Figure 3.8 Program flow chart to compute the steady state with the tuned formulas. The simulation is stopped once the steady state is found, or a maximum number of time steps is reached.

Any of the tuned difference equations obtained with the values of Table 3.1 can be used to compute the steady state of linear AC circuits. Since a simulation performed with any of these methods is a transient simulation, a number of time steps have to be executed in order for the transients to damp. The main advantage these tunable methods have over other discretization methods is their considerably larger time step. For instance, to solve the 60 Hz steady state problem with the trapezoidal rule with an error of 1%, a time step of 0.5 ms is required. The same problem can be solved using equation (3.9) with a time step of about 5.71 ms[13] with no error. That is, the trapezoidal rule would require at least ten times more simulation steps than equation (3.9), even though it is less accurate for a steady state solution. In practice this difference is even larger, since equation (3.9) will introduce additional damping for any frequency different from the tuned frequency as shown in Figure 3.10.

As an example, consider the circuit of Figure 3.9, for which the AC steady state is computed. When this circuit is simulated using equation (3.9), the resulting node voltages, using the fitting function (3.20) on the discrete points obtained, are shown in Table 3.3. The steady state was obtained in 278 time steps using as stop criteria a change in θ of less than 10^{-5} degrees. These results are identical to those obtained with the steady state option of Microtran (UBC version of the EMTP).

Notice from Figure 3.2 that the tuned methods not only produce exact results for the tuned frequency ω, but also for DC. For instance, when a direct current voltage source of 10 V is connected in series with V1, the DC values of Table 3.3 are obtained. Therefore, the methods are capable of computing the DC steady state in addition to the AC steady state.

[13] See equation (3.10).

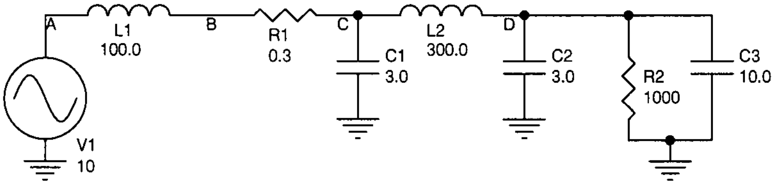


Figure 3.9 Circuit from which the steady state is computed using equation (3.10). Inductances are in mH and capacitances are in μF.

Node	Vdc (V)	Vrms (V)	θ(deg)
A	10.0000	7.071068	0.0000
B	10.0000	11.47991	-16.9305
C	9.99700	11.46359	-17.1180
D	9.99700	24.92909	-31.3557

Table 3.3 Steady state voltages of the circuit in Figure 3.9. The DC values are obtained when a 10 V dc voltage source is connected in series with V1.

Figure 3.10 shows how the magnitude and phase of the voltage phasor at node 'D' of the circuit of Figure 3.9 change over time when a 60 Hz source is applied and the seventh order tuned method is used (equation (3.9)). After an initial transient, the voltage phasor quickly converges to the steady state solution. On the other hand, Figure 3.11 shows the magnitude and phase for the voltage at the same node when the circuit is simulated using the trapezoidal rule with a time step of 166.67 μs. Since the trapezoidal rule does not introduce any damping by itself, all the circuit natural frequencies remain in the simulation outputs for a long time. After 10000 time steps the simulation using the trapezoidal rule was aborted, as indicated by the program flowchart of Figure 3.8, since the simple technique described in section 3.5 to compute the magnitude and phase of voltages fails to produce an acceptable result. If the trapezoidal rule were to be used, some kind of additional filtering would be required instead, as suggested in [31].

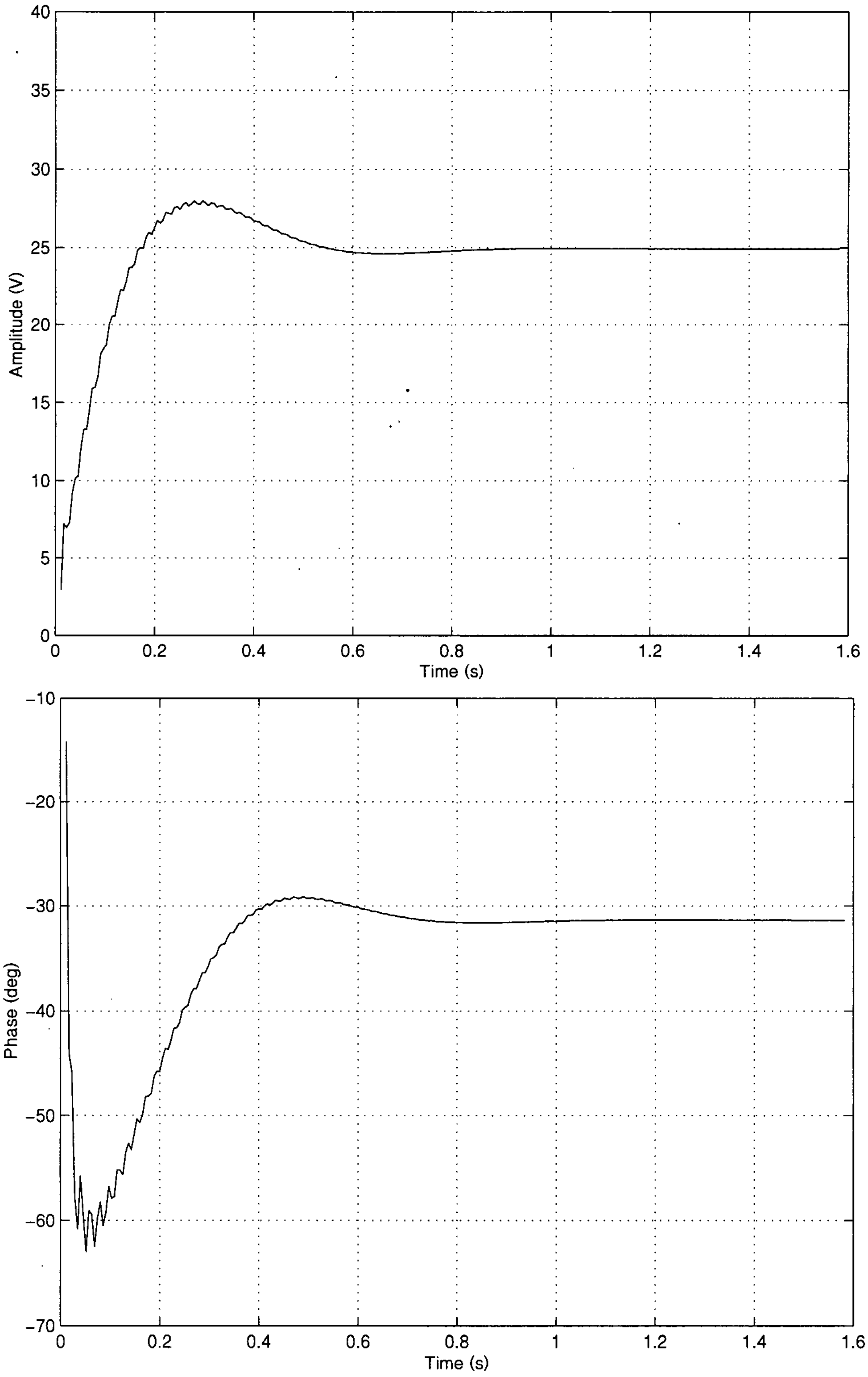


Figure 3.10 Magnitude and phase change over time for the voltage at node 'D' of the circuit of Figure 3.9 for AC excitation using the tuned formula of order 7.

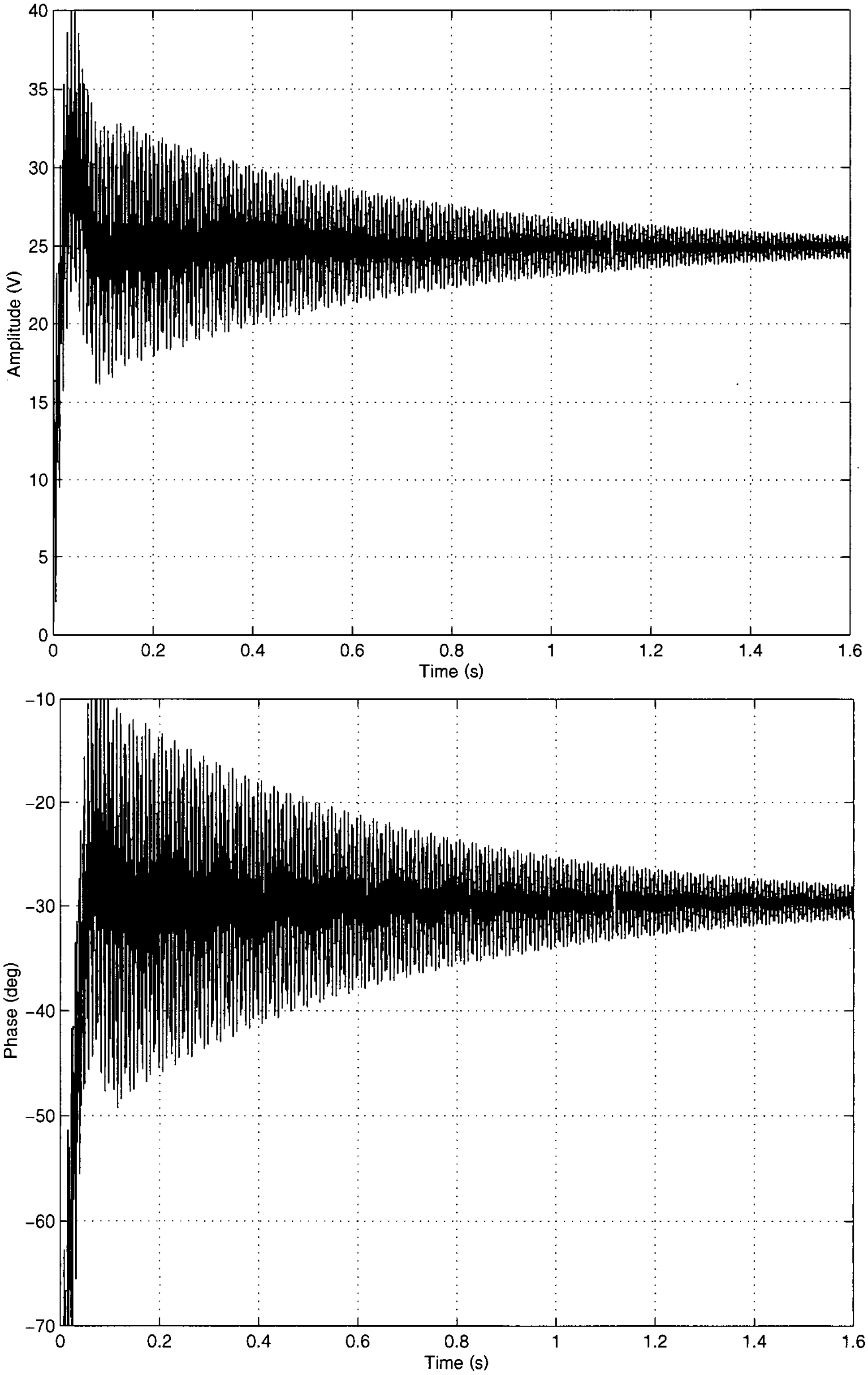


Figure 3.11 Magnitude and phase change over time for the voltage at node 'D' of the circuit of Figure 3.9 using the trapezoidal rule.

3.7 Computational Cost of the Tuned Time-Domain Methods

The time required to compute the steady state of a linear circuit using the tuned formulas may be smaller than the time required when using phasor analysis. For instance the number of numerical operations required to LU factor a $n \ x \ n$ real matrix is given approximately by [32]:

$$\begin{aligned} \#mult &= \frac{1}{3}n^3 + O(n^2) \\ \#adds/subs &= \frac{1}{3}n^3 + O(n^2) \\ \#divs &= \frac{1}{2}n^2 + O(n) \end{aligned} \tag{3.29}$$

and to solve the system of equations using back substitution:

$$BS = 2n^2 - n \approx 2n^2 . \tag{3.30}$$

Therefore if the tuned formulas are used to get the steady state, and it takes q steps to damp all non-tuned harmonic components, the total number of operations is given by

$$Sol(real) \approx \frac{2}{3}n^3 + 2qn^2 . \tag{3.31}$$

On the other hand, if the circuit is solved using phasor analysis, and we consider that each complex addition or subtraction takes two real additions or subtractions, each complex multiplication takes four real multiplications and two real additions, and each complex division takes six real multiplications plus two real divisions and three real additions, the number of operations would be:

$$\begin{aligned} \#mult &= \frac{4}{3}n^3 + O(n^2) \\ \#adds/subs &= \frac{4}{3}n^3 + O(n^2), \\ \#divs &= n^2 + O(n) \end{aligned} \tag{3.32}$$

or

$$Sol(complex) \approx \frac{8}{3}n^3 \tag{3.33}$$

The breakpoint at which the tuned time-domain and phasor solutions start to take approximately the same number of operations is given by:

$$\frac{2}{3}n^3 + 2qn^2 \approx \frac{8}{3}n^3 \Rightarrow q \approx n. \tag{3.34}$$

That is, when the number of step evaluations is smaller than the number of nodes in the circuit, the time required to obtain the solution using the tuned formulas will be less than the time required solving the same circuit using phasor analysis. To verify this, two programs were written in C/C++ to solve two standard IEEE cases (with 118 and 300 nodes), one using phasor analysis, the other using the tuned formula derived from a seventh order polynomial (3.9) as described in section 3.6. Both programs were compiled with Microsoft Visual C++ 6.0, with the same speed optimizations, and using the same algorithm to solve the system of equations: LU decomposition with back substitution. The stop criterion for the proposed tuned formula was a change of less than 0.0001 degrees in the final node angles. The results are shown in Table 3.4 for five different processor types and speeds. Notice that the tuned method solution is faster in all cases since the required accuracy was obtained with a number of evaluation steps q smaller than the number of nodes, as suggested by equation (3.34). Please notice that matrix sparsity techniques were not used, but the above comparison remains valid even if a different solution technique is applied to both the phasor analysis and tuned time-domain methods.

CPU type	118 nodes		300 nodes	
	Phasor	Tuned (q=78)	Phasor	Tuned (q=150)
Ppro 200MHz	0.93s	0.6s	9.34s	6.49s
PII 400MHz	0.34s	0.21s	4.026s	2.673s
Celeron 1GHz	0.45s	0.38s	4.797s	3.655s
Athlon 900MHz	0.22s	0.15s	2.553s	1.512s
PIII 866MHz	0.21s	0.16s	2.293s	1.852s

Table 3.4 Steady state solution times for phasor analysis and a seventh order tuned time-domain formula.

CHAPTER 4: Solving the Static Power flow Problem

4.1 Introduction

Power flow analysis is the fundamental tool for many electric power system studies. It is the primary tool used for contingency analysis and clearance evaluation. It is also required for transient and dynamic stability studies, as well as to generate the base cases for security analysis and transfer capability studies. The most frequently used technique for power flow analysis consists in the solution of a system of non-linear equations using iterative methods such as the Newton-Raphson method, described in section 1.2.4, or one of its derivatives, such as the decoupled power flow [26;33].

In this chapter, the difference equations used in the tuned time-domain solution of the steady state problem are used to solve the non-linear static power flow problem. Before proceeding with the solution of such non-linear power flow problems, suitable models for the elements commonly found in this kind of analysis, such as PQ loads and PV[14] sources, are developed. Then the new tuned time-domain power flow approach is used to solve three standard tests systems, and the results are compared with those obtained using Newton-Raphson iterations. Finally, the developed tuned time-domain power flow is used to obtain detailed PV curves[15].

[14] As they are connected to the PQ and PV buses described in section 1.2.4.

[15] A PV curve shows the variation of a bus voltage as a function of the reactive power, at either that bus or any combination of buses in the power system. They are useful to asses the stability of the power system.

4.2 Modeling the Power System Elements

In order to solve the non-linear power flow problem, adequate circuital models for PQ loads, PV sources, transmission lines, and transformers are required. For transmission lines and transformers the standard models available for iterative power flow solution are used. PQ loads and PV sources are modelled using current and voltage sources respectively. The swing generator is modelled in time domain simulations by using an independent voltage source. Other power flow elements such as phase shifters and loads different from the PQ load can be easily modelled as well. However, since they rarely appear in standard power flow cases, they will not be described in this chapter.

4.2.1 Modeling Transmission Lines

Transmission lines are modelled in the tuned time-domain power flow simulator in a similar way as they are treated in the traditional iterative methods. To this effect the equivalent π circuit of Figure 4.1 is used. The circuit is composed of four elements:

a) The series resistance R, representing ohmic losses and skin effect.

b) The shunt admittance G, representing leakage currents and corona effect. This value is so small that it is normally neglected in power flow analysis. For example, it is not even found in the IEEE common format for exchange of solved power flow data [34]. If used, it is represented as two admittances of half value at both ends of the transmission line.

c) Series reactance X, representing internal and external magnetic flux linkages of the transmission line conductors. For the tuned time-domain power flow, the series reactance X and series resistance R are represented as one discrete RL element as outlined in section 3.4.

d) Shunt susceptance B, representing the voltage charge between conductors and ground. It is modelled as two shunt capacitances of half value at both ends of the transmission line.

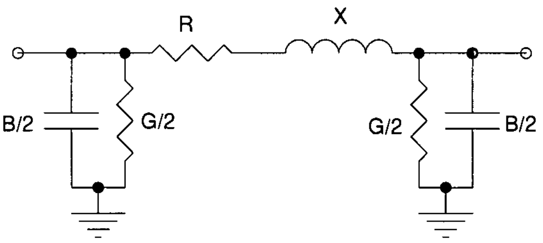


Figure 4.1 Equivalent π circuit of transmission line.

4.2.2 Modeling Transformers

Typically two-winding transformers in power flow studies are represented by the top circuit of Figure 4.2. This model consists of an equivalent branch impedance and an ideal transformer with the desired turns ratio '*a*'. To simplify its inclusion in the admittance matrix required in nodal analysis, the equivalent circuit at the bottom of Figure 4.2 is normally used [35]. The value of the equivalent admittances are given by

$$\begin{aligned} Y1 &= \frac{Y}{a} \\ Y2 &= \frac{a-1}{a}Y \cdot \\ Y3 &= \frac{1-a}{|a|^2}Y \end{aligned} \tag{4.1}$$

As in the case of the transmission line, the resulting complex admittances of Figure 4.2 are represented by their equivalent RL circuit. For the static power flow solution, a fixed-tap transformer model is used. Later, in section 5.2, the under-load tap-change transformer (ULTC) model is described, since it is needed for dynamic power flow simulations.

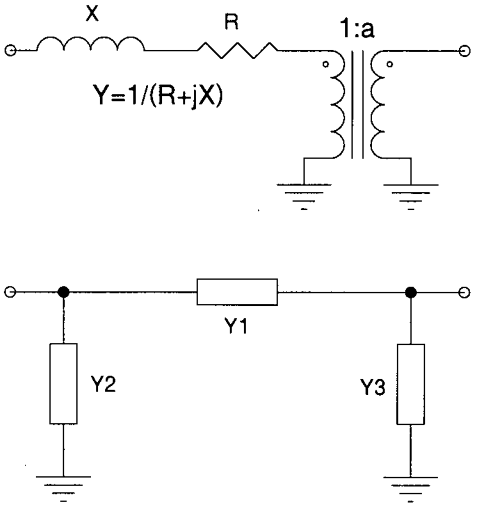


Figure 4.2 Typical two winding transformer model used in power flow studies (top) and its equivalent circuit (bottom). The values of Y1, Y2, and Y3 are given by equation (4.1).

4.2.3 Modeling Shunts

Shunts are admittances connected in parallel to the system buses. They are either capacitive or inductive, and are modelled as a capacitor or inductor. Accordingly to the IEEE common data format for power flow cases [34], capacitors are represented as a positive shunt reactance value +B, while inductors are represented as a negative reactance value –B.

4.2.4 Modeling PQ Loads

One way of modeling PQ loads in time-domain simulations consists of an adjustable sinusoidal current source whose magnitude and phase changes over time, until the specified complex power is obtained. This is accomplished by using the solved voltage at the node where the load is connected. For instance, from the definition of complex power:

$$P+Qj=V\times I^{*}. \tag{4.2}$$

If magnitude and phase are used instead,

$$S_{RMS}\angle\theta = V_{RMS}\angle\alpha \times I_{RMS}\angle{-\beta}. \tag{4.3}$$

On time-domain simulations the peak values of currents and voltages are used, therefore

$$S_{RMS}\angle\theta = \frac{1}{2} V_p\angle\alpha \times I_p\angle{-\beta}, \tag{4.4}$$

from which the magnitude and phase of the current source used to represent the load can be computed as

$$\begin{aligned} I_p &= 2\frac{S_{RMS}}{V_p} \\ \beta &= \alpha - \theta \end{aligned} \quad . \tag{4.5}$$

A very important factor to take into consideration for this load model to work is the numerical transient that occurs every time there is a change in the circuit. The characteristics of this transient depends both on the discretization technique used and how much the circuit changes[2]. For the formulas proposed in this report the numerical transient damps completely after the number of time steps, counting from the last circuit change, is equal to the order of the method used. This numerical transient is illustrated in Figure 4.3 for a non-tuned formula derived from a seventh order interpolating polynomial. For example, if the tuned formula obtained with a seventh order interpolating polynomial is used, then the current source should be adjusted every seventh time step.

To test the PQ model, the simple circuits of Figure 4.4, obtained from [33] are used. The circuit on the left produces a stable result as shown in Figure 4.5, while the circuit on the right, with the load values suggested in [33][16], is ill conditioned and the voltage collapses, as it is also shown in Figure 4.5.

[16] p. 1010. *Q* was slightly incremented from 0.15 to 0.152 to accelerate voltage collapse.

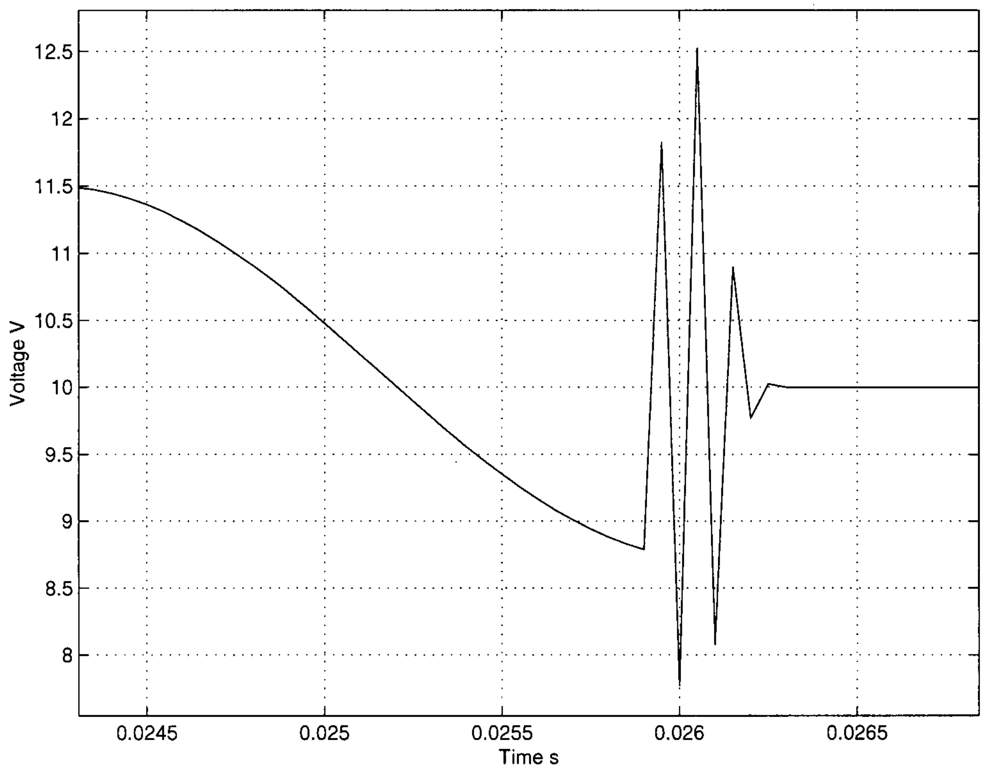


Figure 4.3 Damped numerical oscillations caused by a non-tuned formula derived from a seventh order interpolating polynomial. Similar behaviour is observed with the tuned formulas.

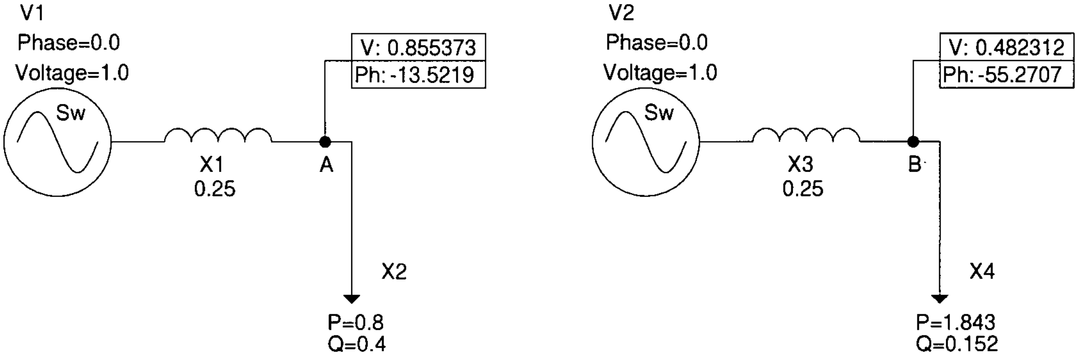
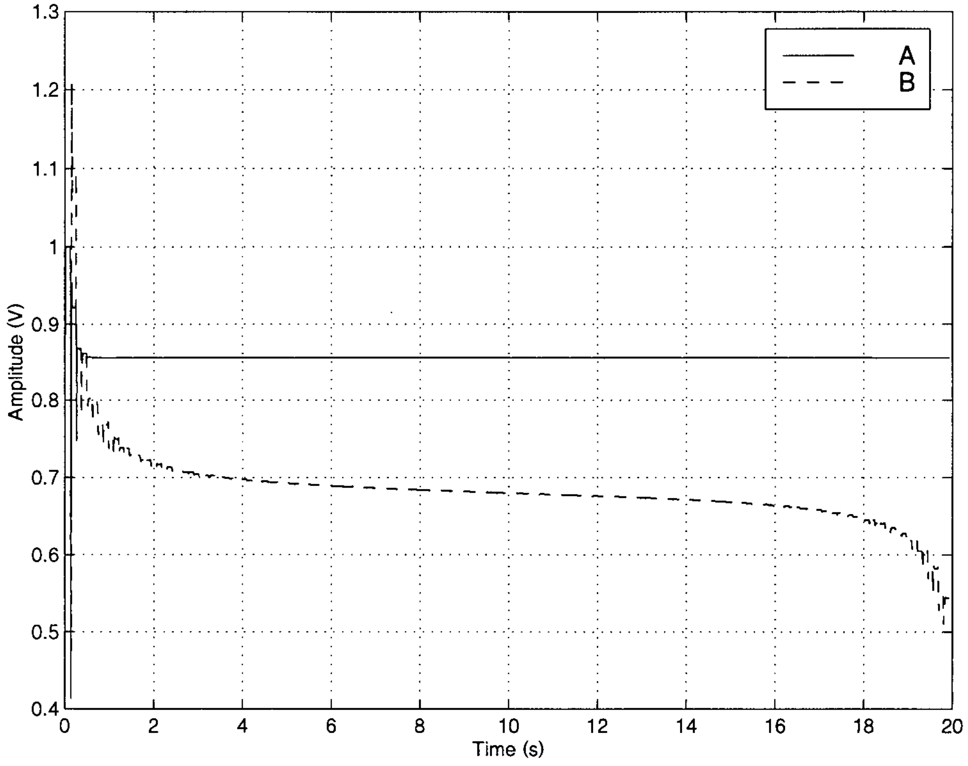


Figure 4.4 Circuits used to test the PQ load model. The circuit on the right yields a solution, whereas the circuit in the left is ill conditioned and the voltage at node 'B' collapses.



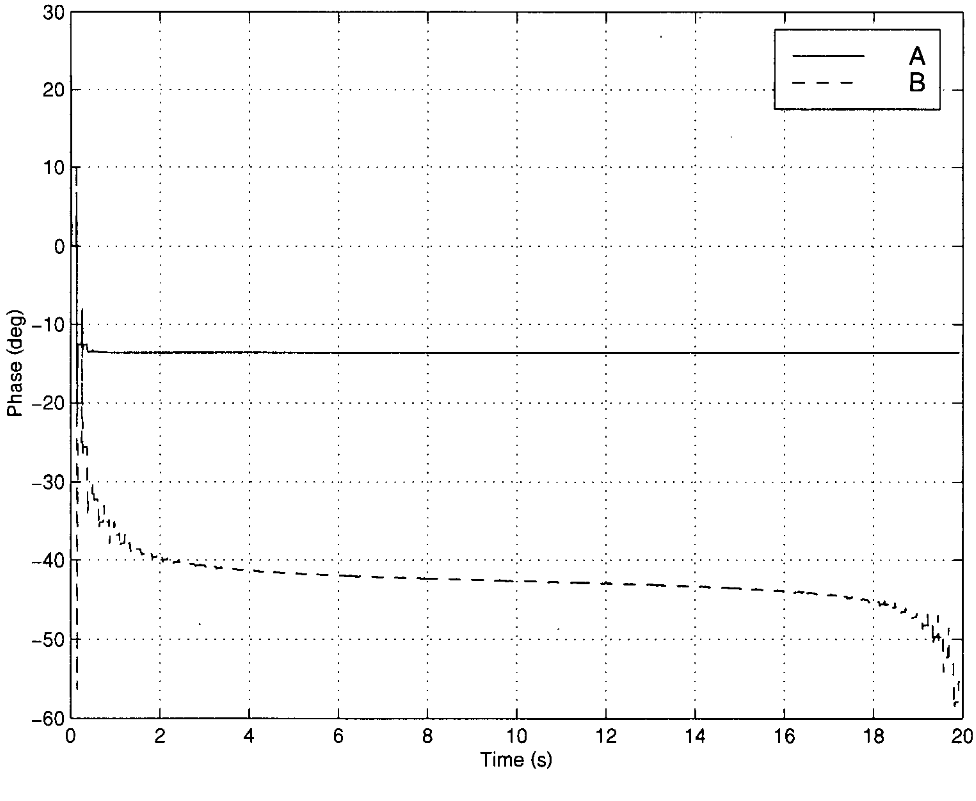


Figure 4.5 Magnitude and phase changes over time for the circuits of Figure 4.4.

4.2.5 Modeling PV Sources

PV sources are modelled with a sinusoidal voltage source in the tuned time-domain simulator. In this case, the magnitude of the voltage is known and the phase has to be determined. Since the current coming out of the sinusoidal voltage source is needed to compute the active power, the modified nodal analysis (MNA) model of such sources is used [27]. The use of the MNA voltage source augments the size of the circuit *Y* matrix by one row and column as explained in section 1.2.2. The steady state is obtained by changing the phase of the voltage source until the desired active power comes out of the voltage source. From formula (4.4)

$$\theta = \alpha - \beta \tag{4.6}$$

and the actual active power coming from the voltage source is given by

$$P_{actual} = 0.5 * V_p * I_p \cos(\alpha - \beta) \tag{4.7}$$

In equation (4.7), α is slowly changed so that P_{actual} gradually becomes equal to the required active power *P*. A proportional-integral controller (PI), as shown in Figure 4.6, is well suited for this purpose. Typical values of the proportional constant *Kp* range from 0.001 to 0.5, while the integrative constant *Ki*, using five terms in the summation, can go from 0.0001 to 0.05. The larger the values of *Kp* and *Ki* the faster the convergence to the steady state, but the risk of the controller going unstable increases. For most cases a *Kp* of 0.01 and a *Ki* of 0.001 seems to work satisfactorily.

Consider for example the circuit of Figure 4.7 taken from [33]. For *Kp*=0.11 and *Ki*=0.011 the steady state is obtained with a tolerance of 10^{-5} in 332 steps, while the magnitude and phases converge as depicted in Figure 4.8. If *Kp*=1 and *Ki*=0.1 the number of steps required to achieve steady state is 3328 time steps, and the magnitude and phase waveforms are those of Figure 4.9. Similarly for *Kp*=0.01 and *Ki*=0.001 it takes 2124 time steps to achieve the steady state, with the magnitude and phase changing as shown in Figure 4.10.

As observed from Figure 4.9, large *Kp* and *Ki* constant values results in faster transition near the final steady state, but the response is somehow noisy. On the other hand, small constant values result in a much smoother response as shown in Figure 4.10, but it takes a longer time to achieve the steady state. It is therefore of crucial importance to select appropriate values for these constants. A rule of thumb consists in using the total generated active power and the total amount of load active power to compute the PI controller constants as

$$\begin{aligned} Kp &= \frac{1}{10\left|Pg - Pl\right|}, \\ Ki &= \frac{Kp}{10} \end{aligned} \tag{4.8}$$

where *Pg* is the total specified active power generated at the PV buses, and *Pl* is the total specified load active power both at the PV and PQ buses. Also, notice that the constants are selected to accelerate convergence not to accurately model the dynamic behaviour of the PV generators.

Capacitor banks and other reactive power sources are also often modelled as PV generators. In this case the active power is set to zero and the capacitor bank is expected to maintain the desired bus voltage providing any necessary reactive power in the process. Reactive power limits can be taken into account by monitoring the reactive power generated by the PV source. If the limit is reached, the phase of the voltage source is computed using

$$\alpha = arctan(\frac{Q_{Limit}}{P}) + \beta, \tag{4.9}$$

and the voltage source magnitude is computed using

$$V_p = \frac{Q_{Limit}}{0.5 * I_p \sin(\alpha - \beta)}. \tag{4.10}$$

Equations (4.9) and (4.10) are functionally equivalent to the PQ load described in section 4.2.4.

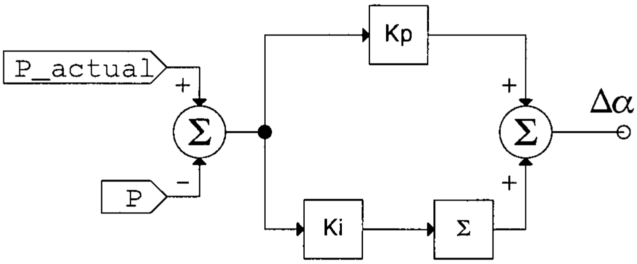


Figure 4.6 PI controller used to adjust the phase angle of the PV sources.

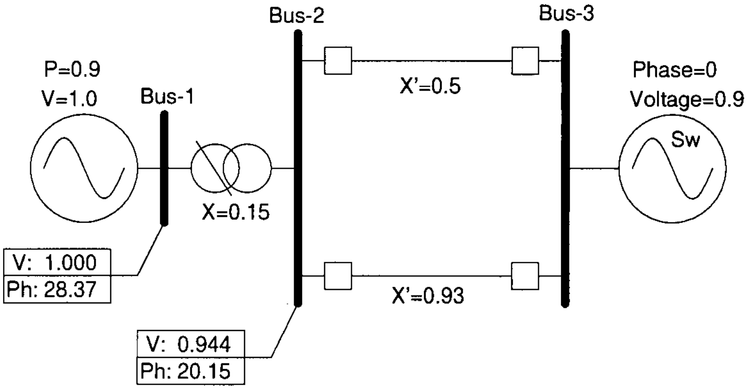


Figure 4.7 Circuit used to test the PV sources model.

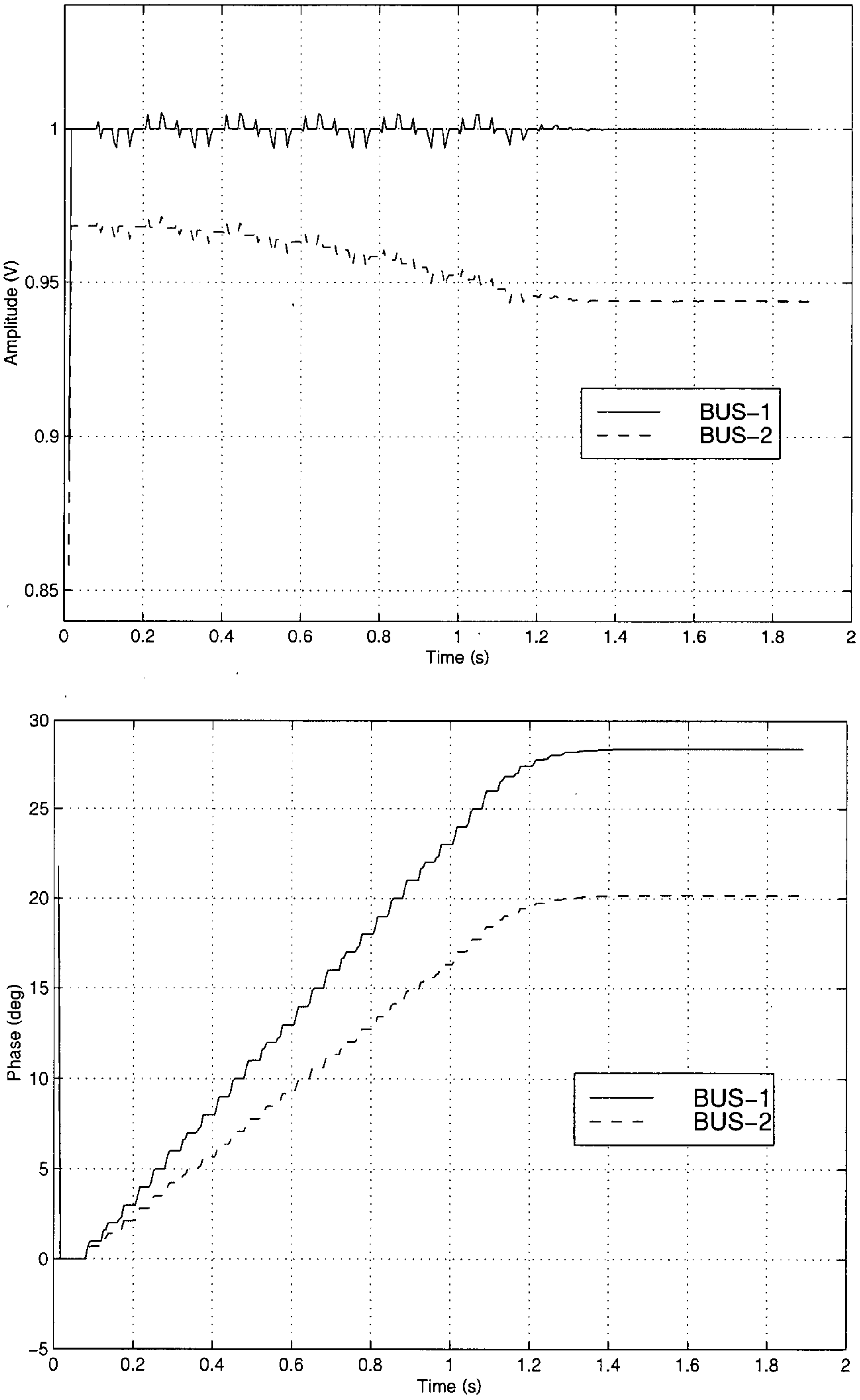


Figure 4.8 Voltages magnitude and phase for the circuit of Figure 4.7 when Kp=0.11 and Ki=0.011

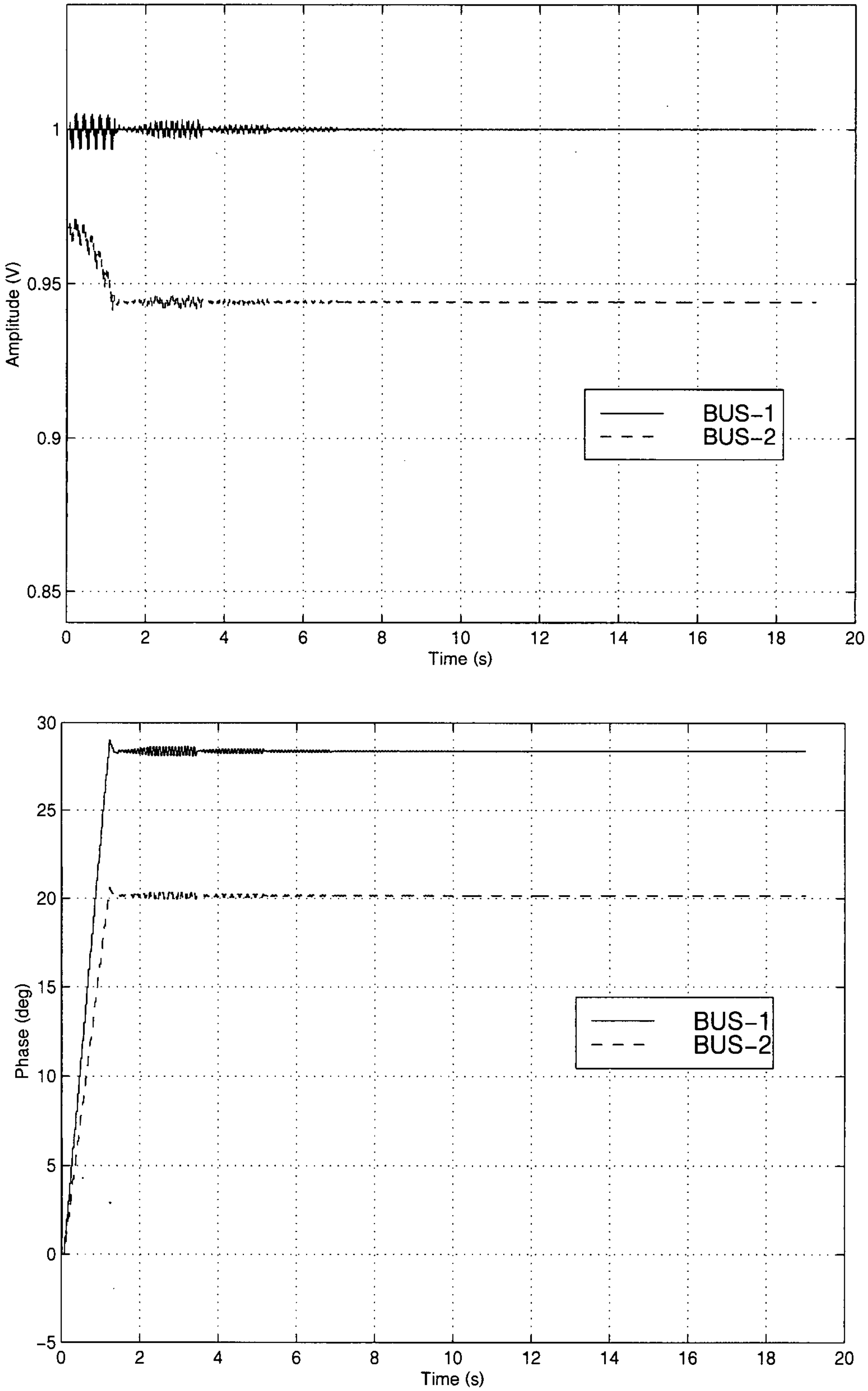


Figure 4.9 Voltages magnitude and phase for the circuit of Figure 4.7 when Kp=1.0 and Ki=0.1

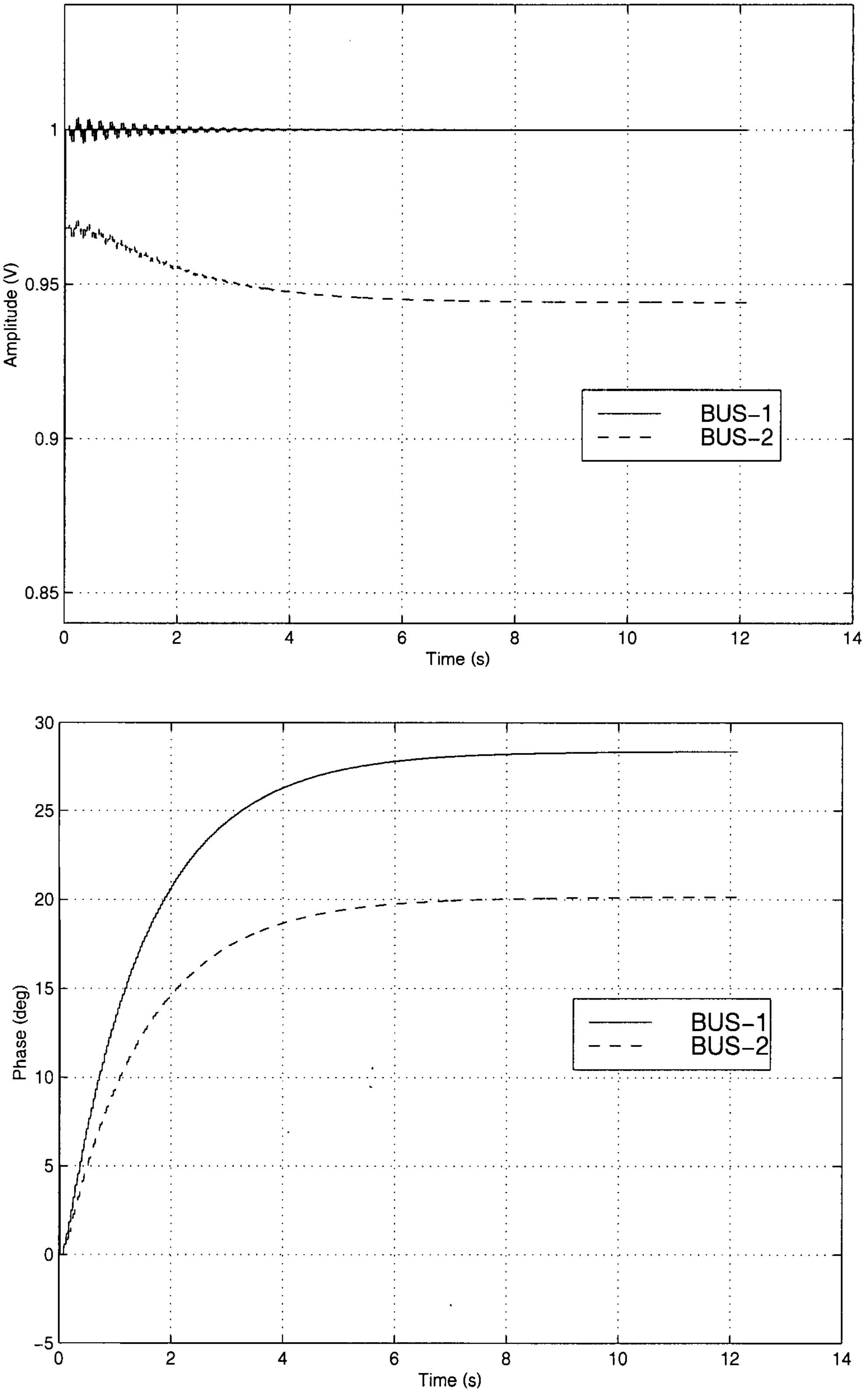


Figure 4.10 Voltages magnitude and phase for the circuit of Figure 4.7 when Kp=0.01 and Ki=0.001

4.3 Tuned Time-Domain Static Power flow Test Cases

To validate the functionality and accuracy of the tuned formulas and techniques described, the models of section 4.2 were added to the simulator program described in section 3.6 and applied to three test cases: the standard IEEE 14-bus and 57-bus test systems, and the CIGRE[17] 32-bus test system. The tests cases were solved both using tuned time-domain power flow and Newton-Raphson iteration power flow, and the voltages magnitude and phase resulting from both methods compared. For the tuned time-domain power flow the seventh order tuned formula was used for all the cases. The tolerance for all the cases and methods was set 10^{-5} (see section 3.6).

4.3.1 IEEE 14-Bus test system

The standard IEEE 14-bus test system of Figure 4.11 [36] consists of 14 buses; five generator buses of which one is the swing generator and the other four are PV sources; eleven PQ loads; fifteen transmission lines; five transformers, of which three are off-nominal tap transformers; and one shunt reactance. Kp and Ki for all the PV generators were set using equation (4.8). The final voltages magnitude and phase are shown in the circuit of Figure 4.11 in the box close to each bus, while the same voltages magnitude and phases are shown in Figure 4.12 as they change over time. The results obtained with the tuned time-domain program are identical to those obtained using the Newton-Raphson iterations program.

[17] Conseil International des Grands Réseaux Électriques.

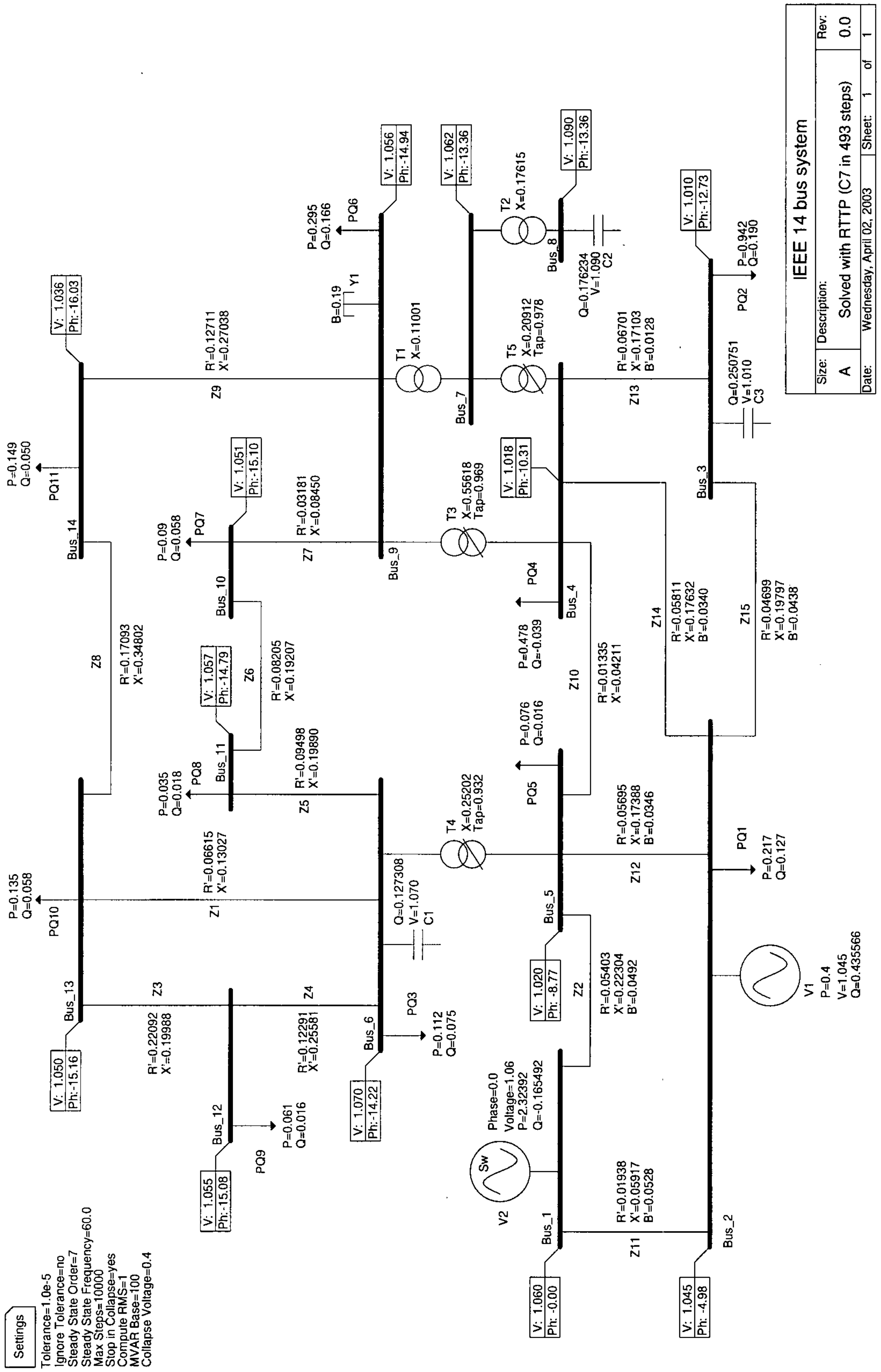


Figure 4.11 IEEE 14-bus test system.

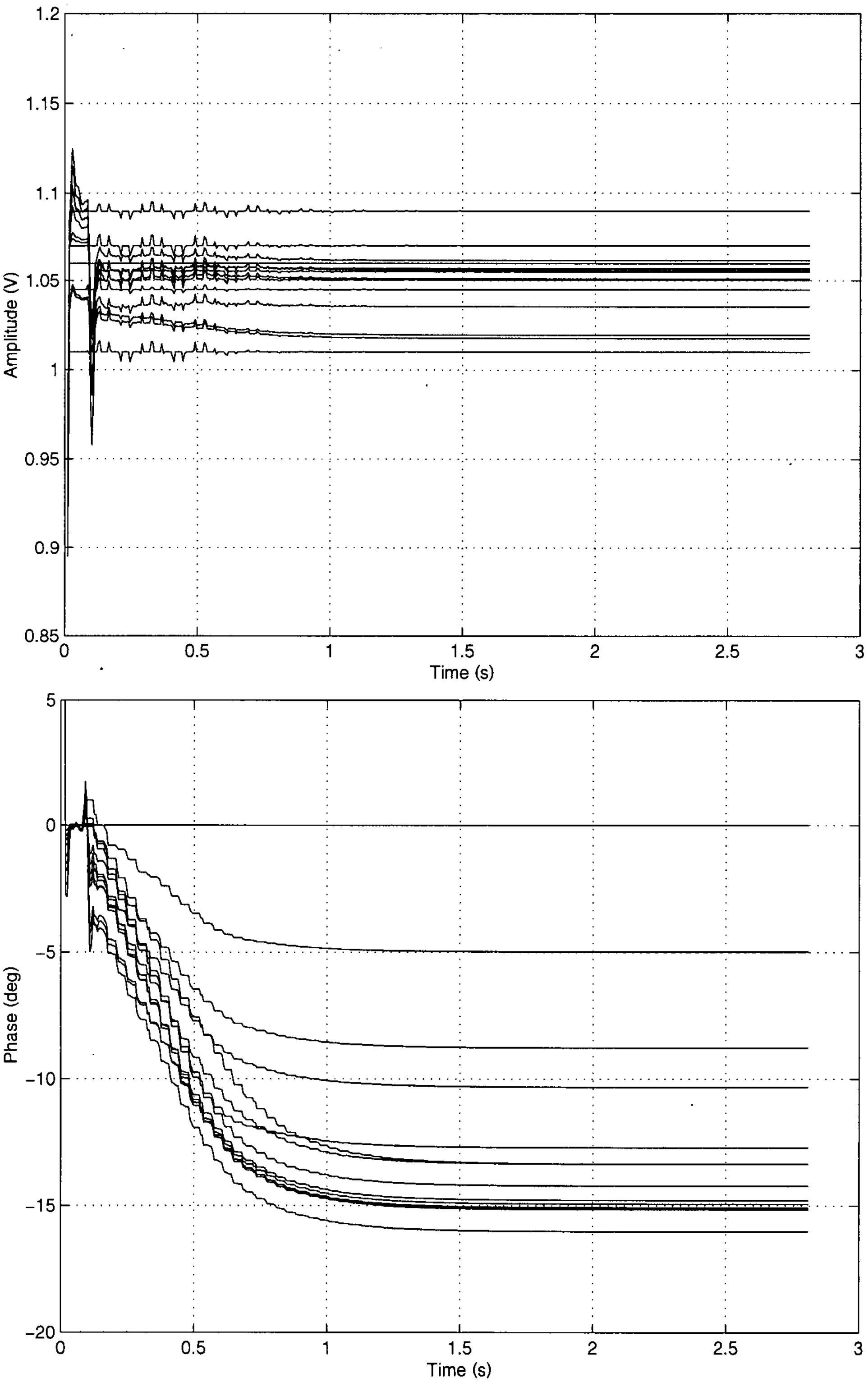


Figure 4.12 Voltages magnitude and phase change over time for the IEEE 14-bus test system solved using tuned time-domain power flow.

4.3.2 CIGRE 32-Bus test system

The static power flow of the 32-bus system shown in Figure 4.13 [37][18] was also obtained using the tuned time-domain static power flow. It consists of: the swing generator, four PV buses, 19 PQ loads, 25 lines, 15 tap transformers, and 12 shunts. Table 4.1 shows the buses magnitudes and phases obtained both using Newton-Raphson in three iterations and the tuned time-domain simulation, obtained in 525 time steps. Except for the last digit in some of the buse phases, the results obtained with both methods are identical.

Bus	Newton-Raphson		Tuned Time-Domain	
	|V|	Phase	|V|	Phase
N201	1.0426	-9.6963	1.0426	-9.6963
N202	1.0236	-11.3695	1.0236	-11.3695
N203	1.0247	-9.9861	1.0247	-9.9861
N204	1.0134	-11.3138	1.0134	-11.3138
N205	1.0332	-9.7571	1.0332	-9.7571
N206	1.0256	-10.1259	1.0256	-10.1259
N207	1.0254	-11.6906	1.0254	-11.6906
N101	1.0547	-5.7613	1.0547	-5.7613
N102	1.0286	-7.1895	1.0286	-7.1895
N103	1.0193	-7.0523	1.0193	-7.0523
N104	1.0296	-5.8108	1.0296	-5.8108
N105	1.0552	-5.8521	1.0552	-5.8521
N106	1.0305	-5.9568	1.0305	-5.9568
N107	1.0303	-7.5213	1.0303	-7.5213
M1	1.0000	4.1614	1.0000	4.1614
M2	1.0000	1.5591	1.0000	1.5592
N1	1.0502	-0.1696	1.0502	-0.1696
N10	1.0840	2.5467	1.0840	2.5468
N11	1.0881	0.8021	1.0881	0.8021
N13	1.0593	-0.3546	1.0593	-0.3546
N14	1.0814	0.8045	1.0814	0.8046
N16	1.0400	-1.6802	1.0400	-1.6802
N2	1.0485	-0.3843	1.0485	-0.3843
N3	1.0398	-1.6628	1.0398	-1.6628
N4	1.0442	-0.9636	1.0442	-0.9635
N5	1.0400	-1.4283	1.0400	-1.4283
N6	1.0443	-1.5498	1.0443	-1.5498
N7	1.0303	-2.0440	1.0303	-2.0440
N8	1.0552	-0.6752	1.0552	-0.6751
N9	1.0554	-0.6351	1.0554	-0.6351
N12	1.0977	-1.3056	1.0977	-1.3055
N15	1.0920	0.0000	1.0920	0.0000

Table 4.1 Node voltages for the system of Figure 4.13, solved using Newton-Raphson iterations and tuned time-domain power flow.

[18] Reference [37] includes this case in the common data format described in [34].

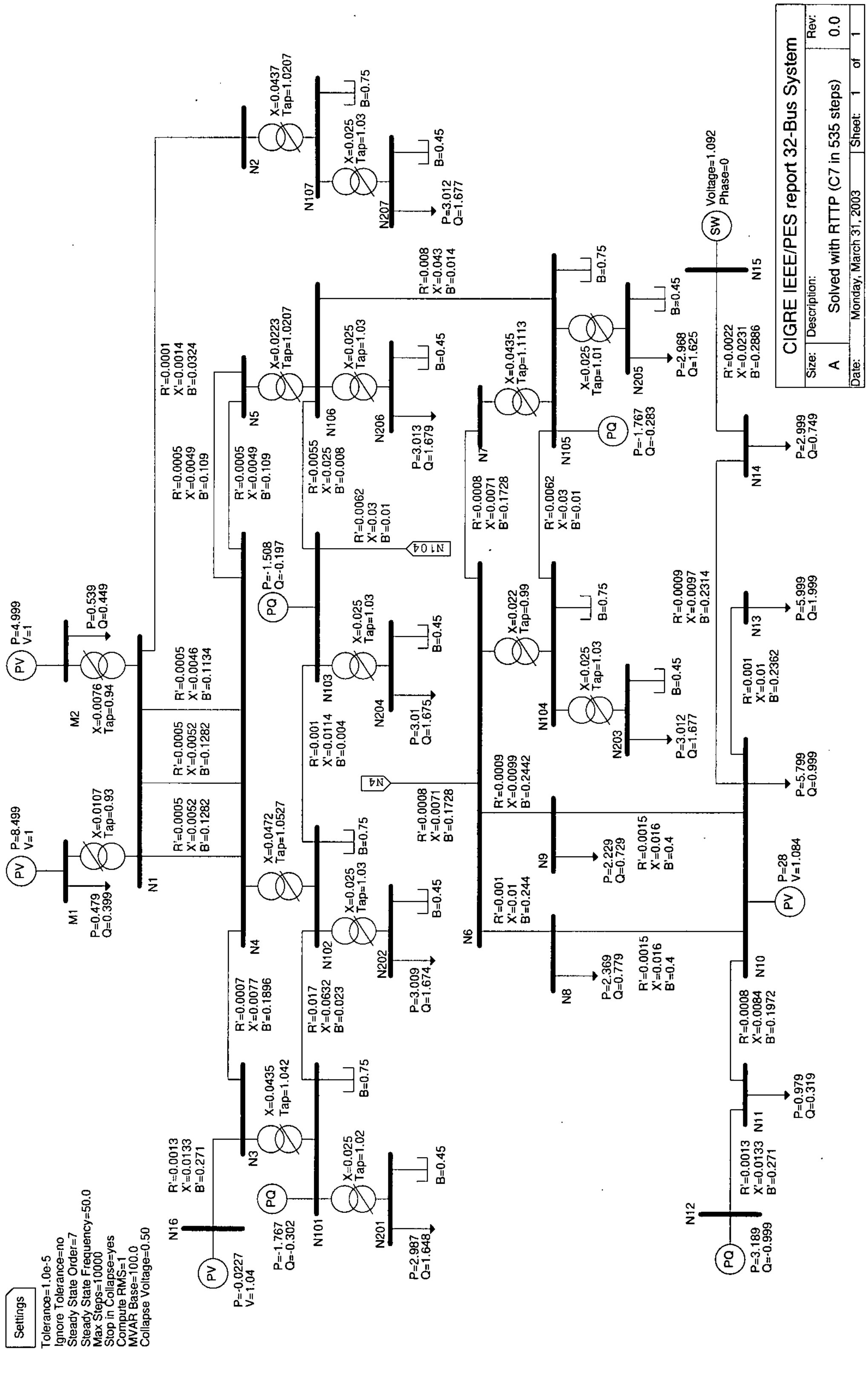


Figure 4.13 32-bus test system. The resulting voltages magnitudes and phase for this system are presented in Table 4.1.

4.3.3 IEEE 57-Bus test system

Table 4.2 shows the voltages magnitude and phase for the buses of the standard IEEE 57-bus system shown in Figure 4.14 and Figure 4.15 [36]. The system consists of: the swing generator, six PV generators, 42 PQ loads, 63 lines, 17 tap transformers, and 3 shunts. The results obtained both with Newton-Raphson in three iterations and the tuned time-domain static power flow are identical. The time-domain solution was obtained in 528 time steps.

Bus	Newton-Raphson		Tuned Time-Domain	
	|V|	Phase	|V|	Phase
BUS_1	1.0400	0.0000	1.0400	0.0000
BUS_2	1.0100	-1.1882	1.0100	-1.1882
BUS_3	0.9850	-5.9881	0.9850	-5.9881
BUS_4	0.9808	-7.3374	0.9808	-7.3374
BUS_5	0.9765	-8.5464	0.9765	-8.5464
BUS_6	0.9800	-8.6741	0.9800	-8.6741
BUS_7	0.9842	-7.6014	0.9842	-7.6014
BUS_8	1.0050	-4.4779	1.0050	-4.4779
BUS_9	0.9800	-9.5847	0.9800	-9.5847
BUS_10	0.9862	-11.4497	0.9862	-11.4497
BUS_11	0.9740	-10.1932	0.9740	-10.1932
BUS_12	1.0150	-10.4712	1.0150	-10.4712
BUS_13	0.9789	-9.8035	0.9789	-9.8035
BUS_14	0.9702	-9.3503	0.9702	-9.3503
BUS_15	0.9880	-7.1902	0.9880	-7.1902
BUS_16	1.0134	-8.8589	1.0134	-8.8589
BUS_17	1.0175	-5.3959	1.0175	-5.3959
BUS_18	1.0007	-11.7296	1.0007	-11.7296
BUS_19	0.9702	-13.2265	0.9702	-13.2265
BUS_20	0.9638	-13.4443	0.9638	-13.4443
BUS_21	1.0085	-12.9290	1.0085	-12.9290
BUS_22	1.0097	-12.8743	1.0097	-12.8743
BUS_23	1.0083	-12.9396	1.0083	-12.9396
BUS_24	0.9992	-13.2921	0.9992	-13.2921
BUS_25	0.9825	-18.1732	0.9825	-18.1732
BUS_26	0.9588	-12.9813	0.9588	-12.9813
BUS_27	0.9815	-11.5136	0.9815	-11.5136
BUS_28	0.9967	-10.4816	0.9967	-10.4816
BUS_29	1.0102	-9.7718	1.0102	-9.7718
BUS_30	0.9627	-18.7196	0.9627	-18.7196
BUS_31	0.9359	-19.3838	0.9359	-19.3838
BUS_32	0.9499	-18.5123	0.9499	-18.5123
BUS_33	0.9476	-18.5520	0.9476	-18.5520
BUS_34	0.9592	-14.1490	0.9592	-14.1490
BUS_35	0.9662	-13.9062	0.9662	-13.9062
BUS_36	0.9758	-13.6348	0.9758	-13.6348
BUS_37	0.9849	-13.4459	0.9849	-13.4459
BUS_38	1.0128	-12.7346	1.0128	-12.7346
BUS_39	0.9828	-13.4910	0.9828	-13.4910
BUS_40	0.9728	-13.6582	0.9728	-13.6582
BUS_41	0.9962	-14.0767	0.9962	-14.0767
BUS_42	0.9665	-15.5328	0.9665	-15.5328
BUS_43	1.0096	-11.3544	1.0096	-11.3544
BUS_44	1.0168	-11.8565	1.0168	-11.8565
BUS_45	1.0360	-9.2701	1.0360	-9.2701
BUS_46	1.0598	-11.1161	1.0598	-11.1161
BUS_47	1.0333	-12.5116	1.0333	-12.5116
BUS_48	1.0274	-12.6107	1.0274	-12.6107
BUS_49	1.0362	-12.9361	1.0362	-12.9361
BUS_50	1.0233	-13.4127	1.0233	-13.4127
BUS_51	1.0523	-12.5334	1.0523	-12.5334
BUS_52	0.9804	-11.4976	0.9804	-11.4976
BUS_53	0.9709	-12.2526	0.9709	-12.2526
BUS_54	0.9963	-11.7097	0.9963	-11.7097
BUS_55	1.0308	-10.8011	1.0308	-10.8011
BUS_56	0.9684	-16.0651	0.9684	-16.0651
BUS_57	0.9648	-16.5837	0.9648	-16.5837

Table 4.2 Static power flow voltages solution for the circuit of Figure 4.14 and Figure 4.15.

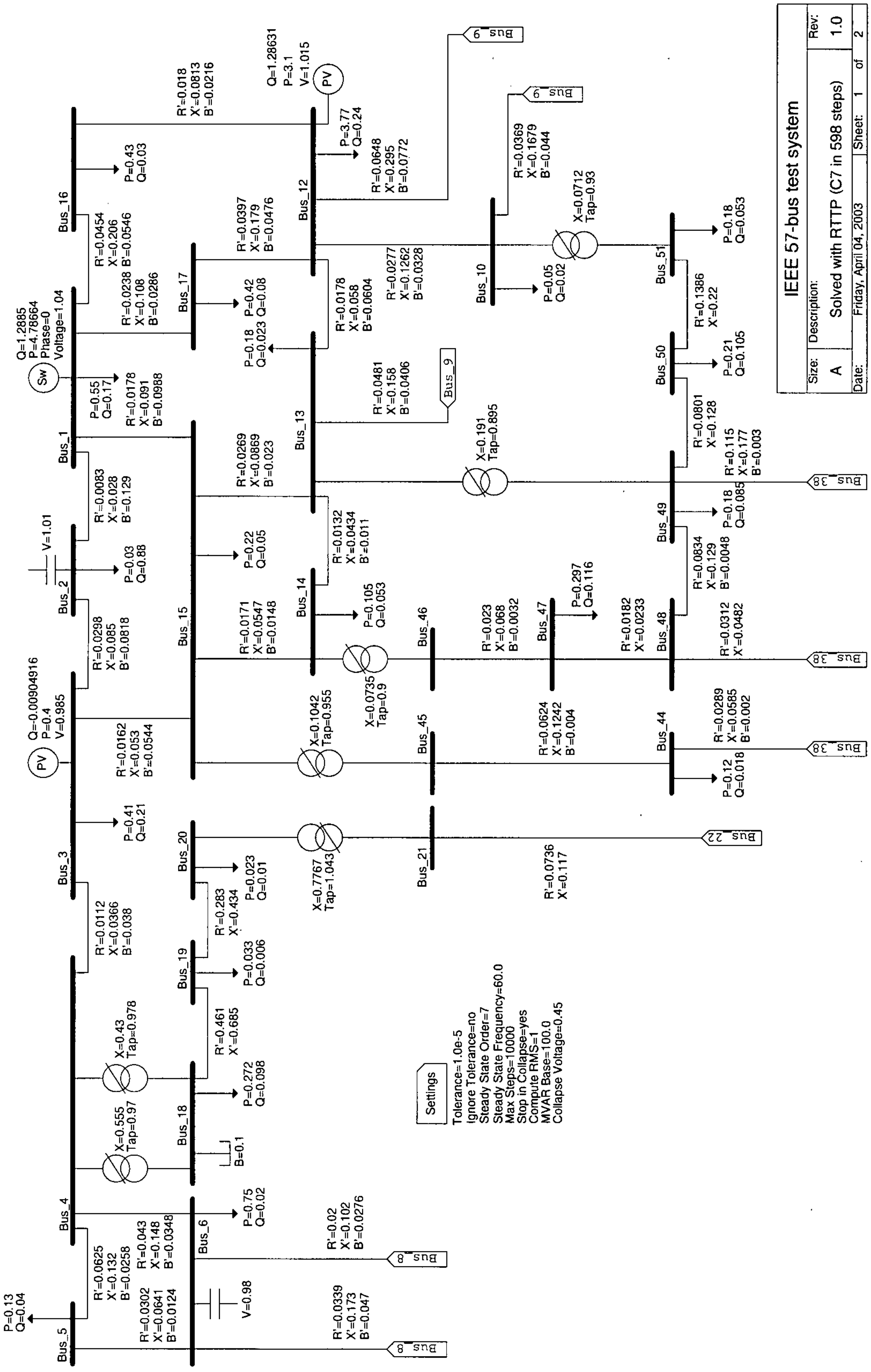


Figure 4.14 IEEE 57-bus test system, sheet 1 of 2.

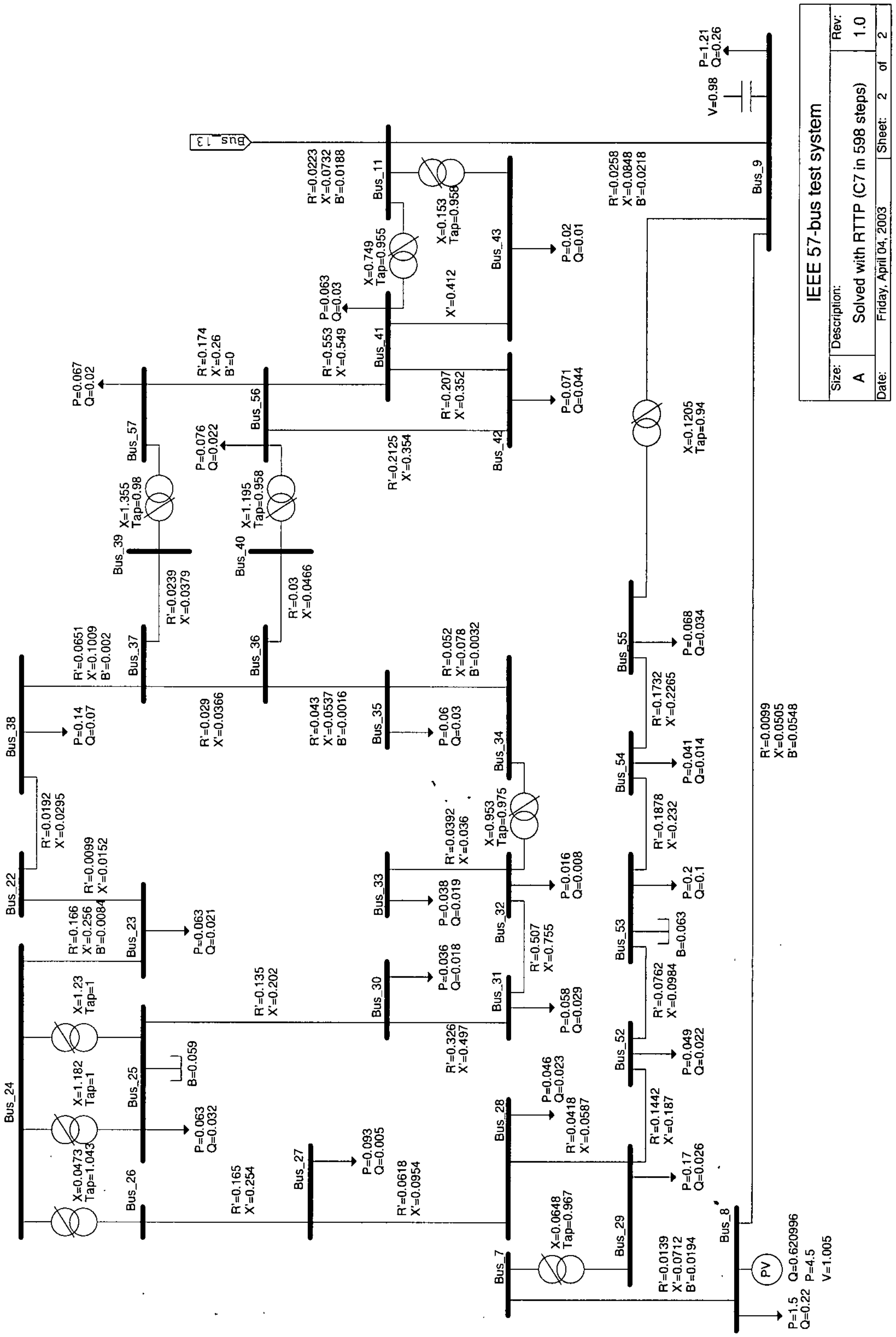


Figure 4.15 IEEE 57-bus test system, sheet 2 of 2.

4.4 Computational Cost of the Tuned Time-Domain Power flow

At this point, it is possible to compare the computational cost of the tuned time-domain simulation against the standard Newton-Raphson iterations power flow. As mentioned in section 1.2.4, the size of the Jacobian matrix when using Newton-Raphson iterations is given by

$$NR = 2 * PQ + PV, \tag{4.11}$$

where PQ is the number of PQ buses, and PV is the number of PV buses. Assuming that it takes IT iterations to reach the desired tolerance, the number of floating point operations required to solve the system using Newton-Raphson iterations is

$$SolNR \approx IT * \frac{2}{3} * (NR)^3. \tag{4.12}$$

On the other hand, the size of the Y matrix for the time domain simulation using the tuned formulas and MNA voltage sources for the PV elements and swing generator is given by

$$TD = PQ + 2 * (PV + 1) \tag{4.13}$$

while the number of floating point operation required to reach the steady state is given by

$$SolTTD \approx \frac{2}{3}(TD)^3 + 2q(TD)^2, \tag{4.14}$$

where q is the number of time steps required to reach the steady state. Therefore, the tuned time-domain simulation could result in a considerable smaller matrix to be LU decomposed if the number of PQ buses is much larger than the number of PV buses. Unfortunately, it is just the opposite if the PV buses outnumber the PQ buses. However, even if the number of PQ buses is larger than the number of PV buses, the tuned time-domain may still result in a larger number of operations in some cases. For example, for the system of Figure 4.13, which has 27 PQ buses and 4 PV buses, NR=58, while TD=37. The solution using Newton-

Raphson converges in *IT*=3 iterations. The time domain solution reaches the steady state in *q*=535 steps. Therefore the approximate number of floating point operations for each method is:

$$
\begin{aligned}
&SolNR \approx 3 * \frac{2}{3} * (58)^3 = 390224 \\
&SolTTD \approx \frac{2}{3}(37)^3 + 2 * 535 * (37)^2 = 1465743
\end{aligned}
\quad . \qquad (4.15)
$$

Therefore, the tuned time-domain solution, for the system of Figure 4.13, takes four times more floating-point operations. This is often the case for small cases, as presented in Table 4.3. Fortunately, the number of time steps required by the tuned method depends on the time constants of the system not on its size and the situation would change radically for a larger system. For instance, if *PQ*=9000 and *PV*=1000, and assuming *IT*=3 and *q*=1000 steps, results on a *SolNR*=6.859x10^{12} and *SolTTD*=1.130x10^{12}.

Case	TTD	NR
IEEE-14	0.02 s	<0.001 s
CIGRE-32	0.03 s	0.03 s
IEEE-57	0.08 s	0.15 s

Table 4.3 Power flow solution times using the tuned time-domain and Newton-Raphson methods.

4.5 Generation of PV Curves

Despite its computational cost compared with Newton-Raphson iterations, one clear advantage of the tuned time-domain over iterative methods is the way the voltages change from step to step, even when the voltage collapses. Unlike Newton-Raphson iterations which just fail to converge for an ill-conditioned system, the tuned time-domain simulation will show which voltages become unstable and collapse, as exemplified in Figure 4.5. One way of assessing the voltage stability of a power system is by means of a curve showing the relation between the active power P and the bus voltage V. This PV curve shows the distance between the current operating point and the point at which the system collapses.

Figure 4.5 suggests that it is possible to get the load PV curve profile of a system node beyond the point of voltage collapse, not only accurately but also economically. Figure 4.16 shows the load PV curve at node 'A' for the left circuit of Figure 4.4. The current source used to model the PQ load at node 'A' was slowly incremented, maintaining the phase angle constant, which results also in a constant power factor, and the voltage and active power were recorded after each increment. That is, instead of using equation (4.5) to compute the injected current at the PQ bus, the following formula was used:

$$\begin{aligned} I_p &= I_p + \Delta I \\ \beta &= \alpha - \theta \end{aligned} , \tag{4.16}$$

where ΔI is chosen to obtain a PV curve with the accuracy and detail required. Notice that by using this methodology there is no stable solution beyond the nose of the PV curve and the simulated power system will eventually collapse, but since the solution follows a trajectory towards the system instability point the states before reaching this point are displayed in the PV curve.

For the PV curve of Figure 4.16, ΔI was set to 0.005. Since the changes are very small from one solution point to the next, the steady state is reached quickly and with little error. The curve was obtained after 6608 time-step evaluations and consists of 944 points using the tuned discretization formula obtained from a seventh order interpolating polynomial (equation (3.9)). Notice that the curve clearly shows the point of voltage collapse and points thereafter.

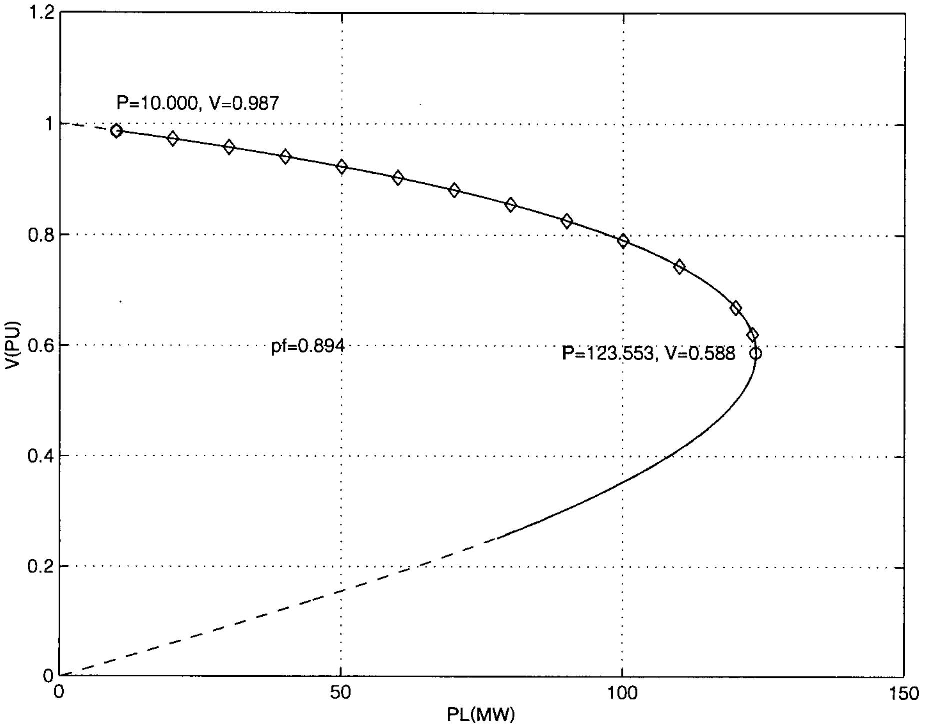


Figure 4.16 Load PV curve at bus A for the circuit of Figure 4.4. The solid line corresponds to the values obtained from the tuned time-domain simulation. The dashed line corresponds to a fifth order interpolating polynomial obtained from the solid line. The diamonds are the solutions obtained using Newton-Raphson iterations.

For comparison purposes, some solutions obtained with Newton-Raphson iterations are also plotted in Figure 4.16 (the diamonds), showing a perfect match with the results obtained with tuned time-domain power flow. When the curve is very detailed, as that of Figure 4.16, the point of voltage collapse can be found directly from the tabular data resulting from the tuned time-domain simulation. An alternative method consist in fitting the tabular data using a given order polynomial, derivate it, and find the tip point of the PV curve. The dashed line of Figure 4.16 comes from a fifth order polynomial obtained with the Matlab function 'polyfit'. Due to the large number of points used, as well as the availability of points around the tip of the PV curve, the fitting is excellent, resulting in a very accurate PV curve peak computation.

4.5.1 Effect of the Current Increment Size

A very small increment value ΔI in equation (4.16) results in a very detailed and accurate load PV curve similar to the one shown in Figure 4.16. That kind of accuracy is unlikely to be needed, therefore it would be wise to select an appropriate value of ΔI such that the resulting PV curve is still accurate enough, but with fewer points, thus resulting in faster computation times. Figure 4.17 shows the load PV curve at node 'A' for the left circuit of Figure 4.4, computed using three different values of ΔI: 0.005, 0.05, and 0.5. The curves obtained with ΔI =0.005 and 0.05 are very similar, while the curve obtained with ΔI =0.5, which only has 10 points, is noticeably different. As a general rule of thumb, curves with 100 to 200 points seem to be accurate and detailed enough for engineering purposes.

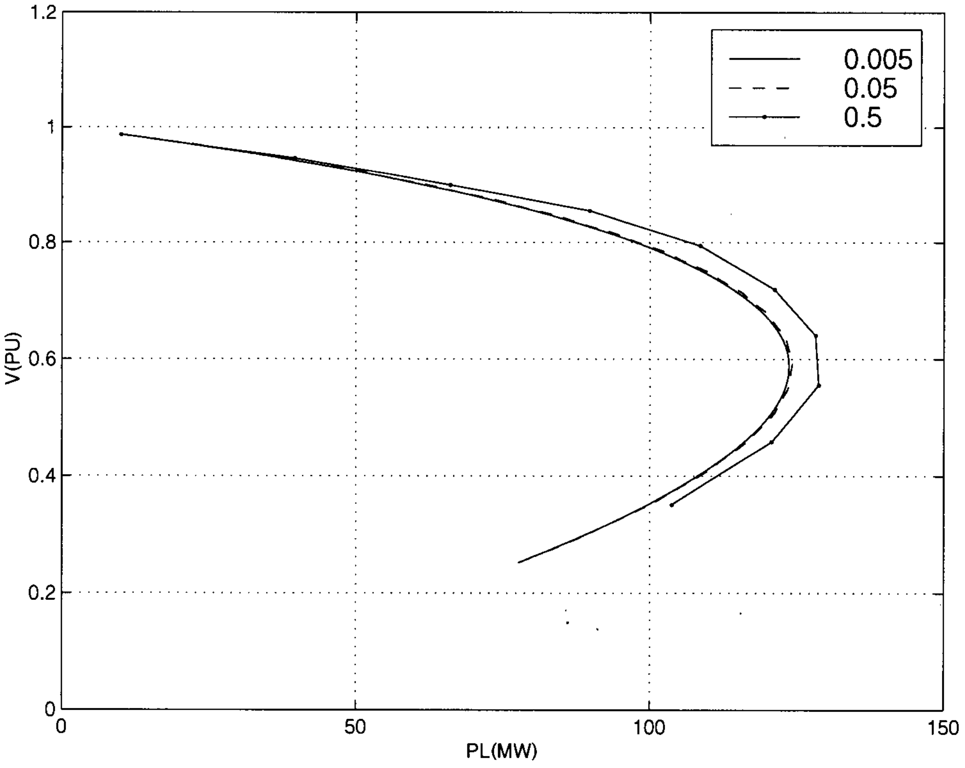


Figure 4.17 Effect of the ΔI size in the generated PV curves. A small value of ΔI results in a more accurate and detailed PV curve, but at a higher computational cost. The plot shows the curves generated with a ΔI of 0.5, 0.05, and 0.005.

4.5.2 Comparison with Published PV Curves

Figure 4.18 shows the load PV curve of node N207 for the 32-bus system of Figure 4.13. Dashed line (1) shows the PV curve as reported in [37][19] which was obtained using the technique described in [38], while the solid line (2) shows the PV curve as obtained using the tuned time-domain technique with ΔI=0.1, resulting in a curve with 88 points. Although similar in form, there is a visible difference on the resulting PV peaks, where the results of the tuned time-domain curve are more conservative. The difference results from the way the curve reported in [37] is obtained. According to [38], the load for which the PV curve is required, is replaced by an equivalent admittance of the form:

$$Y_s = \frac{P_s - jQ_s}{V_s^2}, \tag{4.17}$$

and the system is solved by what looks like standard Newton-Raphson iterations. The resulting admittance is then incremented by a small value and the system solved once again. The procedure is repeated a number of times, until there are enough points to draw the curve. This technique results in PV curves that do not keep the power factor constant, and therefore the difference with the results obtained using the tuned time-domain technique. Furthermore, the technique described in [38] is computationally more expensive than the tuned time-domain power flow technique describe in this report, since a number of Newton-Raphson iteration solutions would have to be computed to obtain a detailed curve.

[19] This document is available in PDF format from http://thunderbox.uwaterloo.ca/~claudio/vswg/Chapter4.pdf. The figure in reference is 4-3.24(a). To obtain the figure from the document, it was printed to a file using a postscript driver. The resulting file was edited, and the table of values from which the figure is plot was exported to a Matlab file from which the original figure was reconstructed.

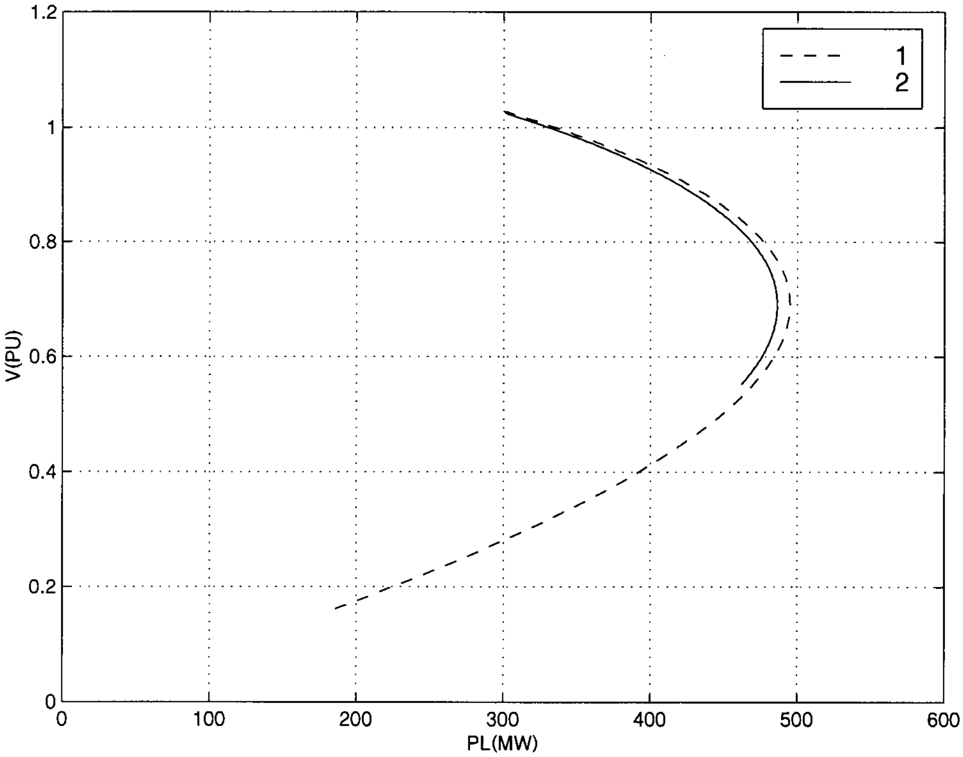


Figure 4.18 Load PV curve for bus N207 of the system of Figure 4.13. The dashed line is the curve reported in [37]. The solid line was obtained using the tuned time-domain power flow described in this report.

4.5.3 Computing the PV Curves for all System PQ Buses

The technique described at the beginning of section 4.5 to obtain the load PV curves can be applied to obtain all system load PV curves. To achieve this objective, the following changes were made to the tune time-domain program described in section 3.6:

1. A flag was added to each PQ load to indicate that the PV curve is required. If a load is sharing a bus with a PV generator, then the PV curve is not obtainable since the active power will remain constant despite the load.
2. Once the static flow is completed, all the sources values are stored. This includes the independent PQ current PV voltage sources, as well as the historic current sources of inductors, capacitors, RL, and RC series connections.
3. The PV curve is computed for each flagged load at a time. Before doing so, the static power flow steady state stored as in point 2 is restored. Once a number of points for the PV curve are obtained, the resulting table is saved to disk.

Experimental results show that the cost of each PV curve generation, which requires the computation of about 100 points, is approximately equal to twice the computational cost of the equivalent tuned time-domain static power flow method. For example, consider Figure 4.19 which shows all the PV curves for the system of Figure 4.11. The static power flow for this system was obtained in 0.02 seconds[20] while the eight PV curves were obtained in 0.35 seconds. Similarly for the 32-bus system of Figure 4.13, the static power flow was obtained in 0.04 seconds, and its 12 load PV curves were obtained in 0.68 seconds. Finally, for the system of Figure 4.14 and Figure 4.15, the static power flow was obtained in 0.08 seconds, and its 35 PV curves were obtained in 4.6 seconds.

[20] Pentium Celleron 1 GHz, running Microsoft Windows XP.

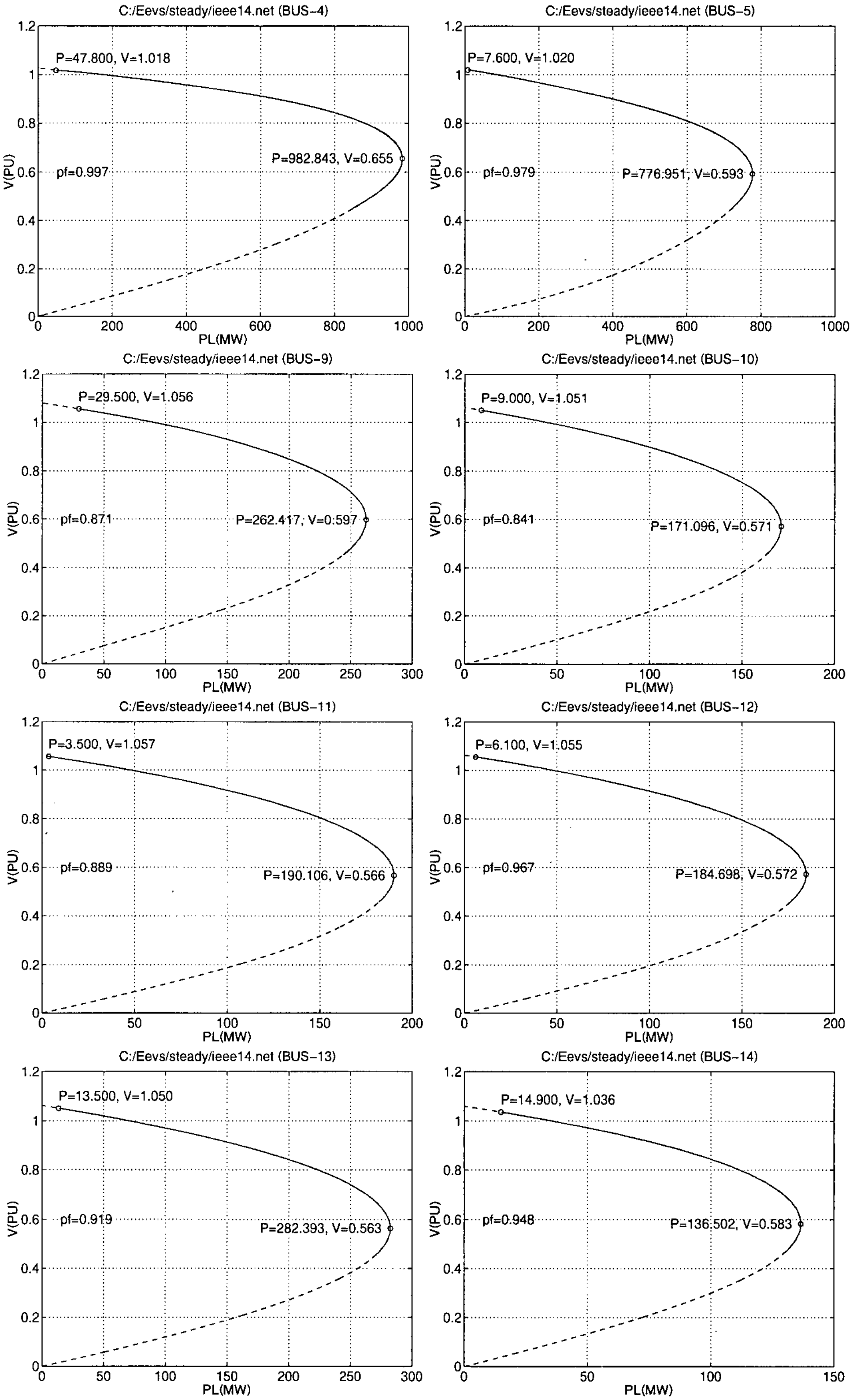


Figure 4.19 All load PV curves for the 14-bus system of Figure 4.11.

4.5.4 A Note on Load Voltage Stability Indices

Voltage stability indices are scalar numbers used to asses the proximity to voltage collapse of a power system. Many indices have been proposed as described in [37], but of particular interest for this research is the PV curve based index proposed in [38] which assesses the proximity of the current operating point to the tip of the curve:

$$P_{Lmgn} = \frac{P_{max\,n} - P_{no}}{P_{max\,n}}, \tag{4.18}$$

where P_{maxn} is the peak value of the PV curve at node n and P_{no} is the current operating active power at node n. Unfortunately, this index, as defined, is not taking into account the voltage collapse at different buses in the system. This condition occurs in some of the nodes of the system of Figure 4.14 and Figure 4.15. Consider, for example, the PV curve at bus 38 in Figure 4.20, where a voltage collapse is detected first at bus 31. Equation (4.18) should be modified, therefore, to show this behaviour as follows:

$$P_{Lmgn} = \frac{P_{stablen} - P_{no}}{P_{stablen}}, \tag{4.19}$$

where $P_{stablen}$ is the last value of the PV curve obtained before a voltage collapse in any of the system buses.

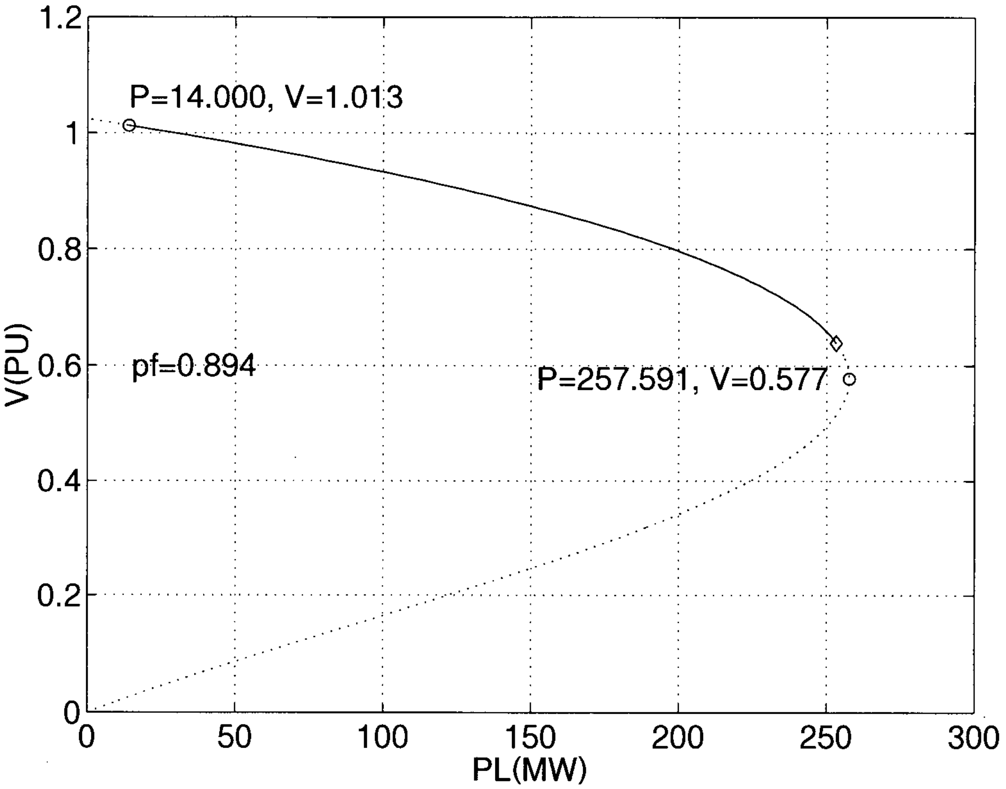


Figure 4.20 The load PV curve for bus 38 of the IEEE 57-bus system showing the point at which the voltage at bus 31 collapses first (diamond).

CHAPTER 5: Dynamic Power flow Using Tuned Time-Domain Formulas

5.1 Introduction

The selection of the correct power system analysis tools for the study of some particular system phenomena greatly affects the results obtained, as well as the time required to obtain such results. Traditional power flow methods are often used because of their efficiency and simplicity, but they are usually inadequate for the simulation of long time dynamic phenomena that often arises in power systems [37;39]. On the other hand, time-domain methods are better suited for this kind of simulations since the transition from one time solution to the next represents the dynamics of the simulated system. Furthermore, time domain methods are free of numerical problems around the point of system instabilities which allows them to closely follow the trajectory of the system solution even for ill-conditioned power flow problems where traditional Newton-Raphson iterations fail to converge. Unfortunately, the numerical integration of the differential-algebraic equations as well as the handling of non-linear elements often posses a high computational cost that makes traditional time-domain techniques very slow.

In the previous chapter, the tuned time-domain formulas were successfully used to solve the non-linear static power flow problem. Fortunately, the time-step required by this new technique is considerably larger than that of traditional time-domain simulations (e.g. EMTP [30]), and due to the exactness the method achieves for the tuned-in frequency it constitutes a viable alternative for the solution of the static power flow problem. Since the results are obtained using a time-domain simulation, it is possible to model and simulate the dynamic behaviour of the power system elements involved in the power flow problem, such as time and voltage dependencies.

Before applying the tuned time-domain formulas to the solution of the dynamic load-flow problem, the dynamic system time-scale must be well thought-out. That is, the time constants of the system under study should be considered, so that the system, once discretized with the tuned time-domain formulas, does not have much introduced error in the simulation process. Consider for example the magnitude and frequency response of the tuned formulas of Figure 3.2, from which four main observations can be derived:

1) No distortion is introduced by any of the formulas at zero frequency (DC).
2) No distortion is introduced by any of the formulas at some tuned frequency.
3) The distortion introduced at low frequencies near DC is very small. Furthermore, it is not the magnitude error, which is almost flat, but the phase shift which introduces the most distortion.
4) The distortion introduced around the tuned frequency is also very small.

Traditional time-domain simulations which use a fixed time-step, such as the EMTP, require a time step small enough to accurately represent the smallest time constant of the simulated system. This time step depends on the discretization technique used, for example, for the trapezoidal rule it is often considered that a time step of one tenth of the maximum time constant is enough to simulate the circuit with a maximum distortion of 3% [2]. This criterion can also be applied to the tuned difference equations described in this report. Consider for example Figure 3.2, from which it can be observed that for the formula derived from a seventh order interpolating polynomial the phase shift is less that 3% at about a normalized frequency of 0.025. If a tuned frequency of 60 Hz is used, the resulting time step, using the values of Table 3.1 and equation (3.8), would be approximately 5.71 ms, which corresponds to a smallest accurately discretized system time constant of 0.025*5.71 ms=0.23 s[21], enough to represent many dynamic power system behaviours [37] [33].

In this chapter, simple dynamic models for the tuned time-domain dynamic power flow simulation are described. They include under-load tap-changing transformers (ULTC), as well as time varying and voltage dependent PQ load. Also, the method used to handle outages in the tuned time-domain simulation program is described. Finally a system example is presented, which uses the developed models and techniques, with its corresponding time domain solution.

5.2 Dynamic Modelling of ULTC Transformers

Under-load tap-changing transformers (ULTC) are used to control the amount of active and reactive power delivered to the loads as well as to regulate the voltage at the buses they are connected to [33] [26]. ULTC transformers are usually made with 8, 16, or 32 steps, with a regulation range of ±10% of the rated tap value [40], but it is not uncommon to find transformers with a range of regulation of ±7.5% and ±5%.

Figure 5.1 shows the basic connection of an ULTC transformer. The tap changing contacts are placed near the center of the transformer to keep the coil physically and electrically symmetric in order to avoid electrical and mechanical stress. In addition to the

[21] If a formula derived from a fourth order polynomial were to be used, the smaller accurately discretized time constant would be 0.15*2.55 ms=382.5 μs! Unfortunately this formula is less stable than the one derived from a seventh order polynomial as shown in Figure 3.3.

step change size, the desired voltage range, the number of taps, the delay for the first tap movement and delay time of any tap change thereafter is usually specified [33].

The inclusion of the dynamic model of ULTC transformers in the tuned time-domain program of section 3.6 is similar to that described in section 4.2.2. The equivalent circuit of Figure 4.2 is adjusted according to the selected tap value, and is then included in the *Y* matrix of the system of equations (1.1). The *Y* matrix has to be LU decomposed every time the ULTC transformer position changes, before solving the system for the node voltages and voltage source currents.

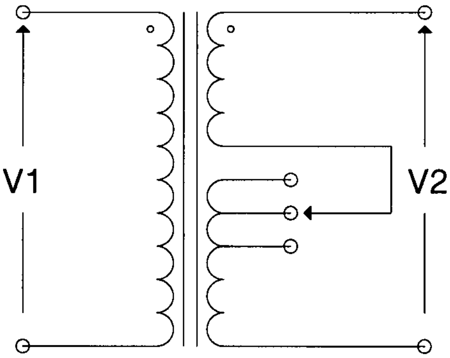


Figure 5.1 Basic connexion of a ULTC transformer.

Unlike standard Newton-Raphson iterations which produce a real number for the tap position that may or may not be one of the actual available physical values, the tuned time-domain method described in this report selects the closest tap position available to satisfy the transformer operation conditions. For example, consider Figure 5.2 which shows all the buses magnitudes and phases after solving the static power flow of the system of Figure 4.11. The tap changing transformers are required to maintain the voltage at buses 4 and 5 to 1.02 and 1.027 V (PU) respectively. In Figure 5.2 the action of the tap transformers is clearly visible as a step change on the system buses voltages magnitude and phase, until the desired bus voltages are obtained.

When the load PV curves are obtained using the model of the ULTC's instead of fixed-tap transformers some interesting results are obtained. For instance, Figure 5.3 shows the PV curves for nodes 4 and 5 of the system of Figure 4.11, both with fixed-tap and ULTC transformers. Although the initial effect of the ULTC is to increase the bus voltage as the load at such bus is increased, shortly after the effect reverses, resulting in a less stable system, as shown by the peak values of the PV curves in Figure 5.3. This undesirable effect on the stability of the system caused by the ULTC is also commented on by some authors, for example Kundur in [33][22], who made the observation for a similar case.

[22] Page 986: "The net effect of each tap movement of transformer T6 is to reduce bus 11 voltage rather than increase it."

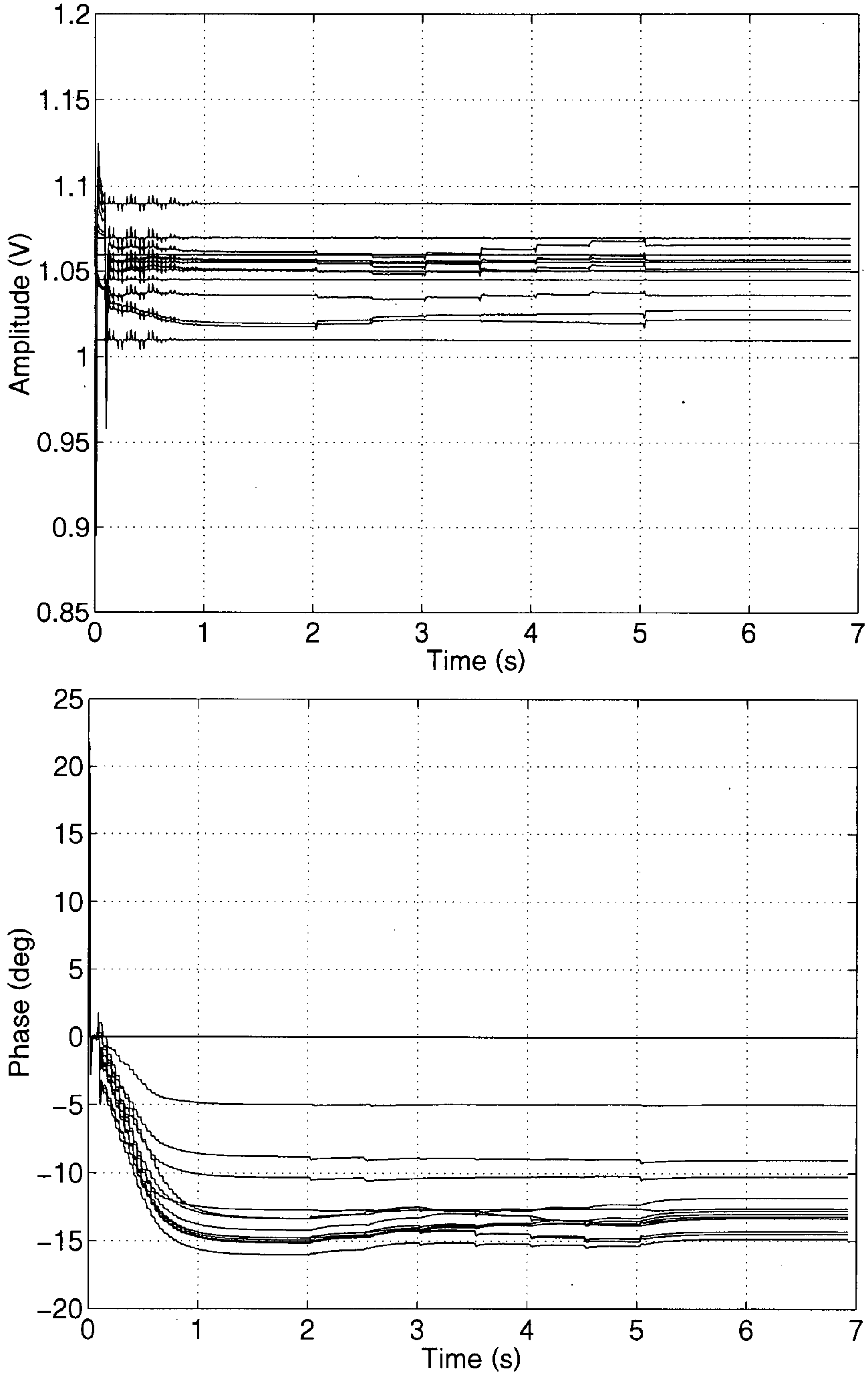


Figure 5.2 Effect of the ULTC transformers on the static power flow solution of the system of Figure 4.11.

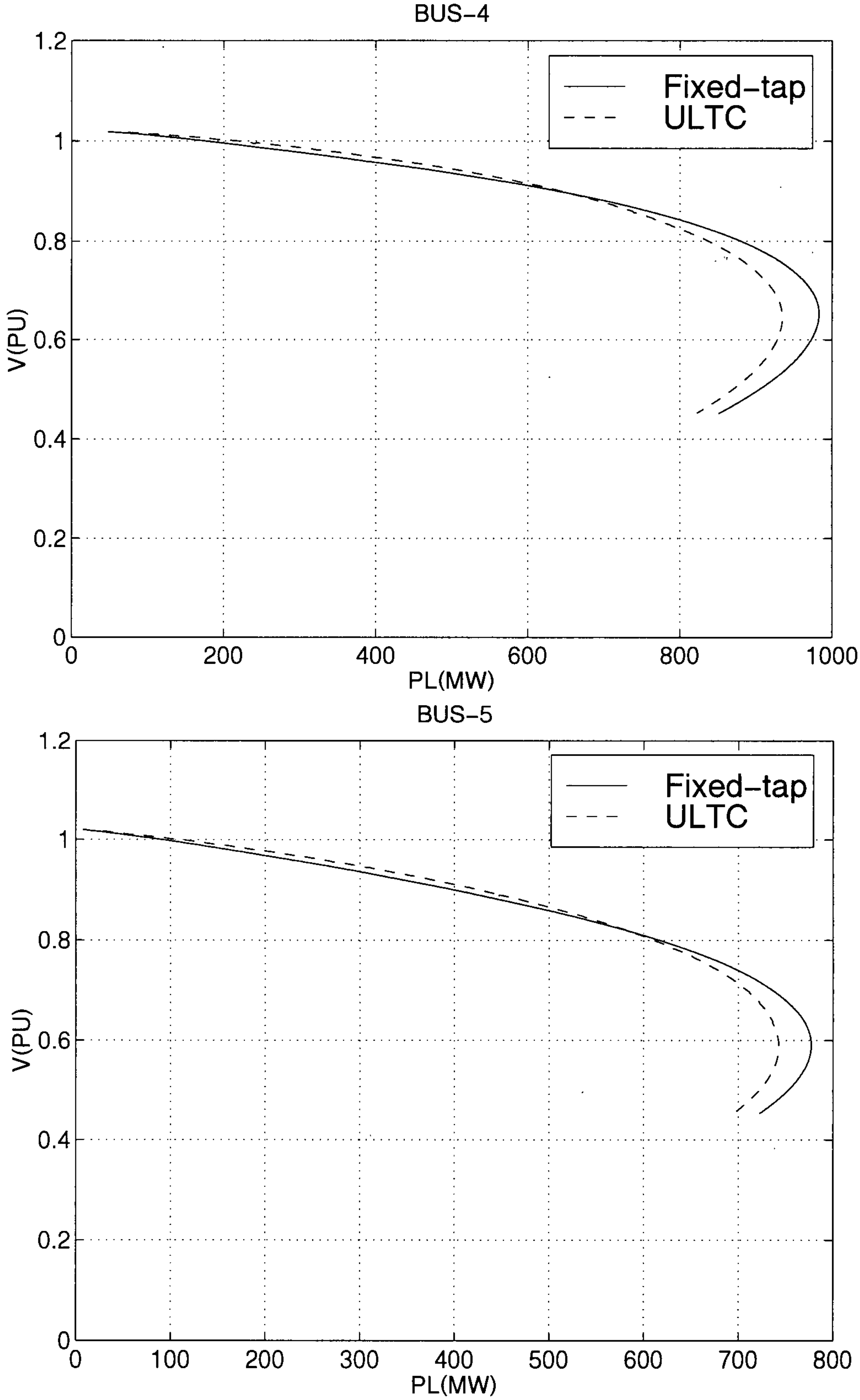


Figure 5.3 Effect of the ULTC transformers in the PV curves of buses 4 and 5 for the system of Figure 4.11.

5.3 Dynamic Modelling of Loads

The simplest way of modelling the dynamic behaviour of loads consists in the adjustment of its complex power value over time:

$$S(t) = P(t) + Q(t)j\,. \tag{5.1}$$

The implementation of equation (5.1) on the tuned time-domain power flow program described in section 3.6 is straight forward, since the PQ loads are modelled as current sources. Also, since any changes on the current sources are only reflected in the vector of independent sources of the system of equations (1.1), the computational cost associated with such changes is very small.

Furthermore, the variation of loads with voltage as described in [41] can be easily modelled as well. Assuming the system frequency to be constant, the load variation with voltage can be represented using an exponential model,

$$\begin{aligned} P &= P_0\left(\frac{V}{V_0}\right)^{pv} \\ Q &= Q_0\left(\frac{V}{V_0}\right)^{qv} \end{aligned}, \tag{5.2}$$

where P_0 and Q_0 are the nominal power values at nominal voltage V_0, and pv and qv are the coefficients selected according to the load type as shown in Table 5.1.

Load Type	*pv*	*qv*
Incandescent lights	1.6	0
Fluorescent lights	1.2	3.0
Heaters	2.0	0
Induction Motors, full load	0.1	0.6
Induction Motors, half load	0.2	1.6

Table 5.1 Some typical coefficients for the exponential load model of equation (5.2).

The effect of a dynamic load model in a simulation can be illustrated when the load connected at bus 5 in the system of Figure 4.11 is step-incremented by the values given in Table 5.2. After 10 seconds (when the steady state has been reached), the value of the load is increased every five seconds keeping the power factor constant and with the ULTC transformers activated so as to compensate for the voltage drop at the controlled buses. Figure 5.4 shows the voltages magnitude and phase for buses 4 and 5. Figure 5.4 clearly shows the effect of the load variation on the buses voltages, as well as the compensating effect of the ULTC transformers. Notice that the final load value is near the point of voltage collapse, as indicated by the nose peak of Figure 5.3.

Time (s)	10.0	15.0	20.0	25.0	30.0	35.0
P	2.5	3.5	4.5	5.5	6.5	7.5
Q	0.5263	0.7368	0.9474	1.1579	1.3684	1.5789

Table 5.2 Step values over time for the load connected at bus 5 in the system of Figure 4.11.

The load can be dynamically adjusted using any function. One function that is found in the literature is a lineal change over time of the load values [37;38]. Figure 5.5 shows the voltages magnitude and phase at buses 4 and 5 for the system of Figure 4.11 when the load at bus 5 is incremented linearly using the equation

$$\begin{gathered} m = \frac{P_{final} - P_{initial}}{t_{final} - t_{initial}} , \\ P(t) = m(t - t_{initial}) + P_{initial} \end{gathered} \tag{5.3}$$

where P_{final}=7.5, $t_{initial}$=10.0, t_{final}=35.0, and $P_{initial}$ is the resulting value of the static power flow solution. Similarly, $Q(t)$ was set to keep the power factor constant.

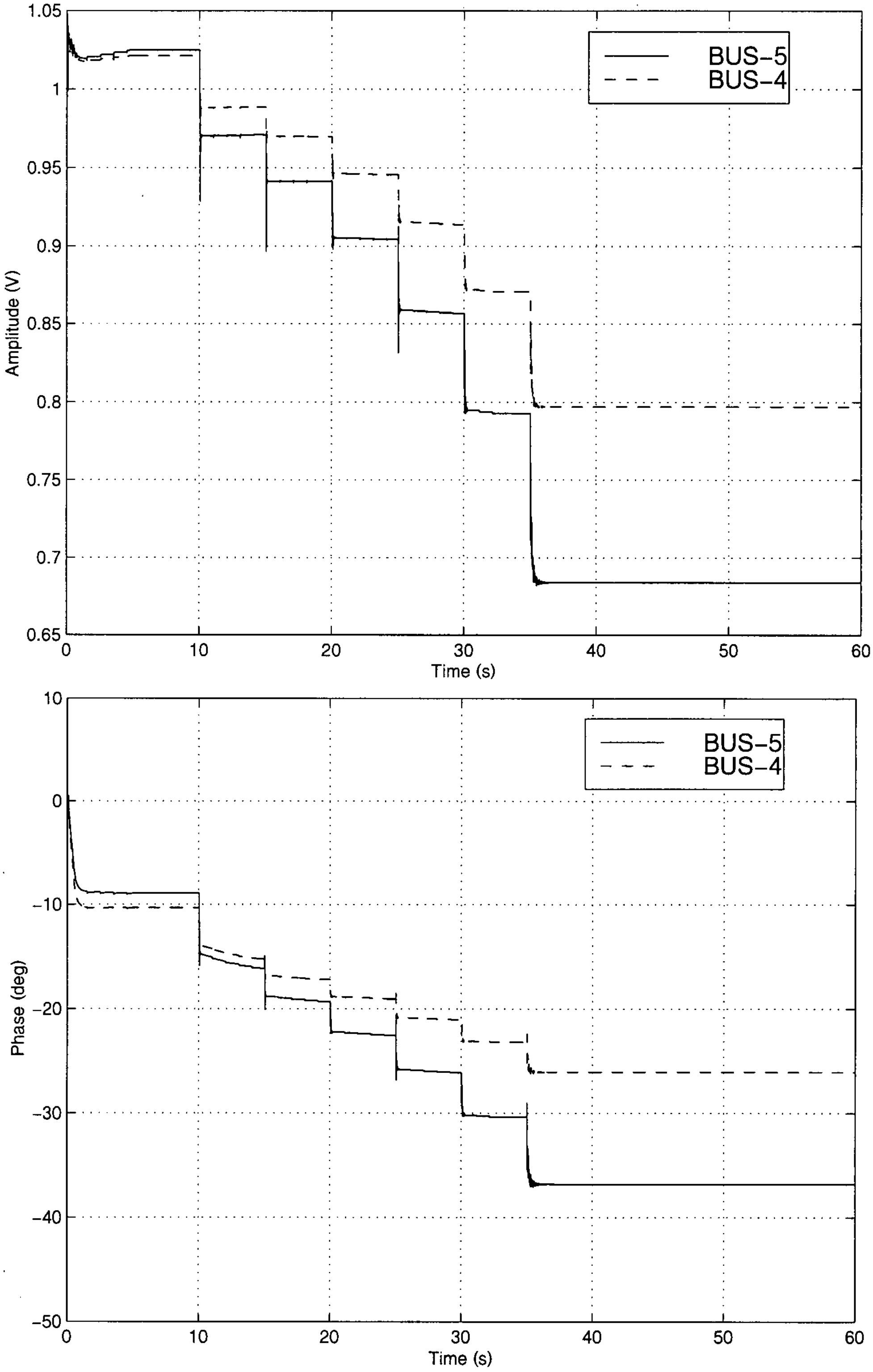


Figure 5.4 Magnitude and phase for the voltages at buses 4 and 5 for the system of Figure 4.11 when the load at bus 5 changes with the values given in Table 5.2.

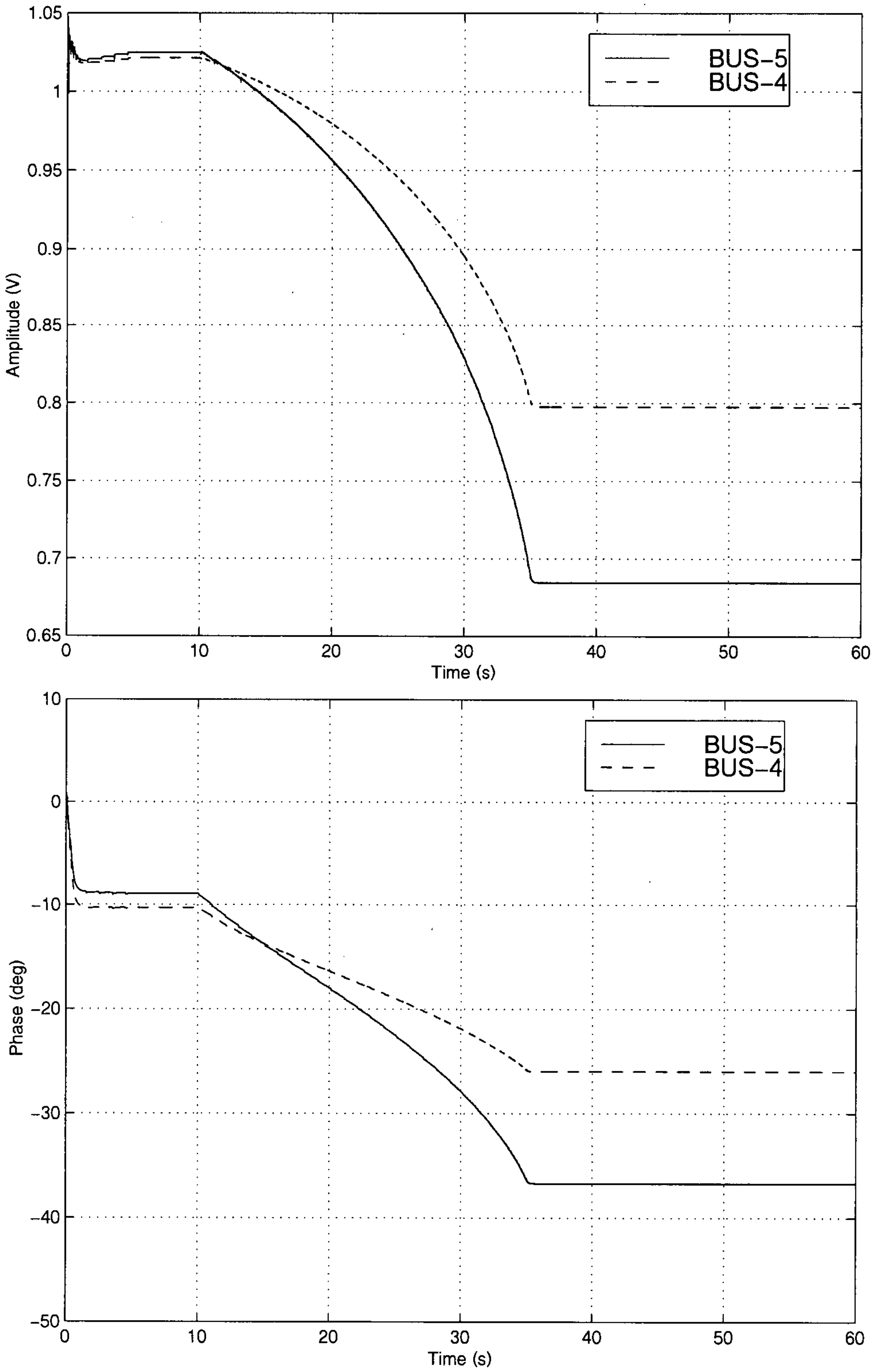


Figure 5.5 Magnitude and phase for the voltages at buses 4 and 5 for the system of Figure 4.11 when the load at bus 5 changes linearly with time.

5.4 Simulating Outages

Outages such as the loss of a transmission line or generator can be simulated using a simple switch model. Switches can be modelled using a resistance whose value is very high when the switch is open and very low when the switch is closed. In any case, the matrix of admittances Y of the system of equations (1.1) changes whenever the state of the switch changes, which results in a new LU decomposition of the corresponding system of equations, and the costs associated with it (section 4.4).

Resistor values of $10^{10}\,\Omega$ for an opened switch and $10^{-10}\,\Omega$ for a closed switch, which are a few orders of magnitude different from those normally encountered in power flow cases, seem to work satisfactorily. The operation of the switch is considered instantaneous, in the sense that the current across the switch is not expected to cross zero before toggling its state. This behaviour differs from the EMTP implementation of switches, since in the tuned time-domain simulation the time steps are much larger, and the point of zero current crossing probably will be far from any of the discrete time points anyhow.

Figure 5.6 shows the voltages magnitude and phase for buses 4 and 5 of the system of Figure 4.11. The fixed load at bus 5 was incremented to 0.76+0.16j so to emphasize the effect of the outage: after 40 seconds the lines connecting buses 4 and 5 were removed by means of two switches at both ends of the line. The resulting transient is clearly visible, as well as the effect of the ULTC transformers that managed to bring the buses voltage back to their desired values just after a few tap changes.

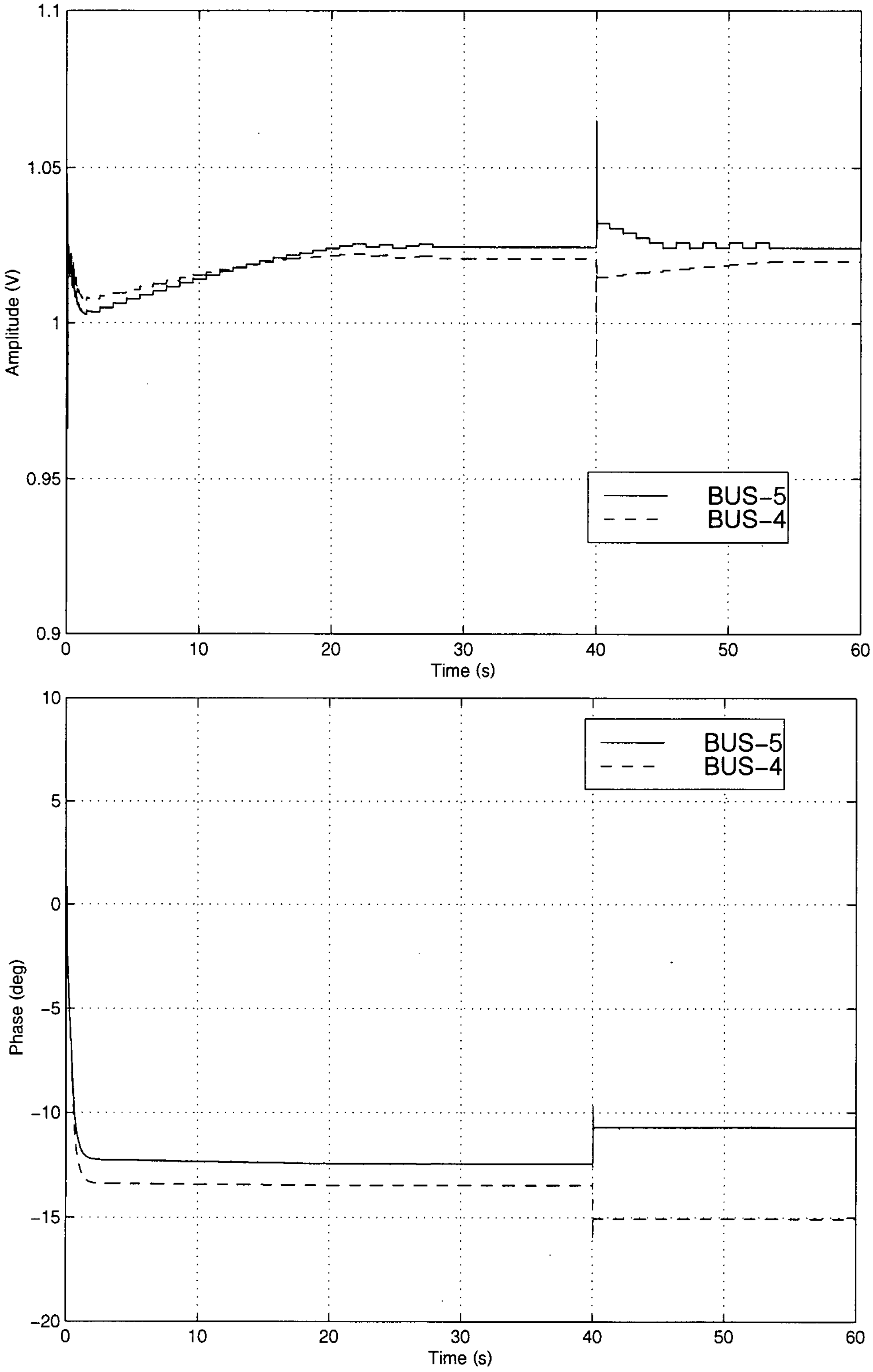


Figure 5.6 Magnitude and phase for the voltages at buses 4 and 5 for the system of Figure 4.11 when the line connecting both buses is removed.

5.5 Example of Dynamic Power flow Solution

Linearly changing loads, ULTC transformers, and outage simulation mechanism are required to simulate the dynamic power flow behaviour of the CIGRE 32-bus system of Figure 4.13, as presented in [37]. According to [37], the CIGRE 32-bus system was originally created to study the voltage collapse of the northern part of Belgium in 1982 and reported in [42]. The series of events that drove the system to collapse are described as follows:

1. At t=30 s, the loads at buses N201 through N207 started to increase steadily at a rate of 30% in 7200 s. All other loads remain constant. Although the ULTC transformers in the actual system were controlled by the operator, the tap of the transformers connected to loads N201 to N207 was allowed to change automatically in the simulation.
2. The transmission line connecting buses N3 and N16 was tripped off at t=5000 s.
3. At t=7230 s the loads at buses N201 to N207 stop increasing.
4. At 7400 s, generator M2 trips off.

For the tuned time-domain dynamic power flow simulation, the ULTC transformers are assumed to have 32 steps with a range of 10%, with a first tap movement delay of 30 s and 30 s thereafter. It is also assumed that the linearly increasing PQ loads keep their power factor constant as described in section 5.3. For the outage modelling of the line between nodes N3 and N16, switches were placed at both ends of the line which trip off at the required outage time of 5000 s.

The tuned time-domain dynamic power flow simulation was run for a total simulation time of 7500 seconds. For a time step of approximately 5.71 ms, corresponding to the tuned formula derived from a seventh order polynomial, the number of time steps required to complete the simulation is around one million. In order to save disk space and speed-up plotting, the output of the simulation was saved every 700th time step, or around every fourth second. The simulation was completed in about 68 seconds in a 1 GHz Pentium Celeron running Windows XP.

Figure 5.7 shows the voltages magnitude and phase for buses N201, N202, and N204 that result from the tuned time-domain simulation. As the loads increase over time, the ULTC transformers adjust their tap position so as to keep the voltage at the load buses more or less constant. At t=5000 s the line connecting busses N3 and N16 trips off, which results in the large transients shown in the plots. The system recovers from this outage, but the ULTC transformers quickly reach their tap limit at t=5165 s. From that point on, the buses voltage magnitude decrease steadily until the voltage quickly collapses at t=7160 s when the buses voltage become erratic.

The results of Figure 5.7 are very similar to those presented in [37][23] but the predicted time for the collapse is different since the aforementioned reference states that the voltage collapse is due to the tripping of generator M2 at t=7400 s. This is caused by the way loads were represented in reference [37]: "constant impedances with internal taps (no limits) to avoid power flow convergence problems", while in the tuned time-domain dynamic power flow simulation they were represented as pure PQ loads. To obtain better results, the loads can be represented as suggested in [42], the original source for the CIGRE 32-bus case, as a mix of fixed and voltage dependent loads given by

[23] Page 4-46, Figure 4.3-25.

$$
\begin{aligned}
P &= P_0(0.6+0.4V^2) \\
Q &= Q_0(0.5+0.5V^2)
\end{aligned}, \qquad (5.4)
$$

where the quadratic part results from equation (5.2), with *pv=qv=2* and $V_o=1$. Figure 5.8 shows the magnitude and phase for the voltages at buses N201, N202, and N204 when using equation (5.4). Since the loads do not increase as fast as before due to their voltage dependence, the system remains stable after the loads stop increasing at t=7230 s. It is now the loss of the generator at bus M1 at t=7400 s which precipitates the voltage collapse as shown in the figure.

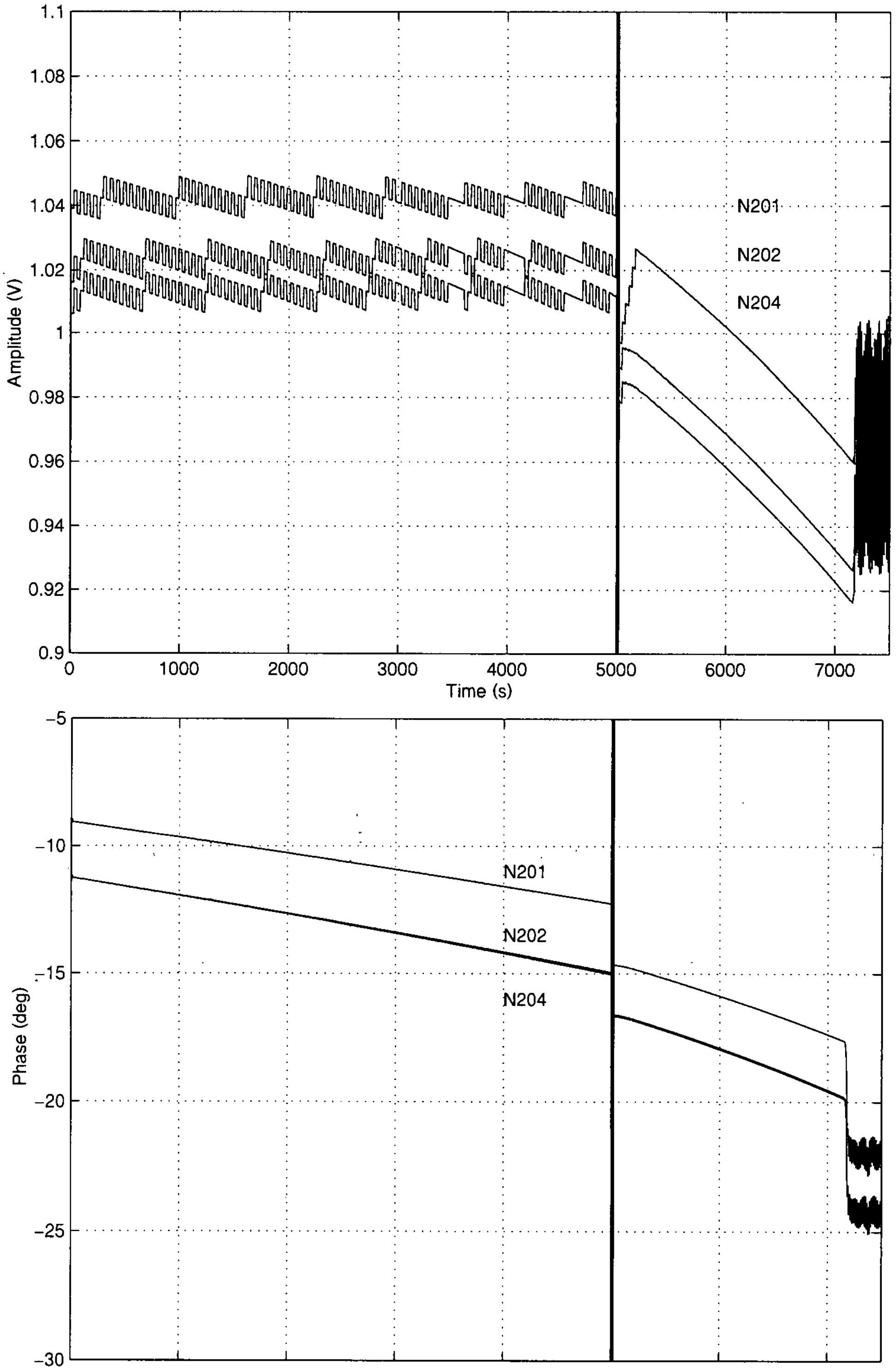


Figure 5.7 Dynamic voltage power flow solution at buses N201, N202, and N204 for the CIGRE 32-bus system of Figure 4.13 using pure PQ loads.

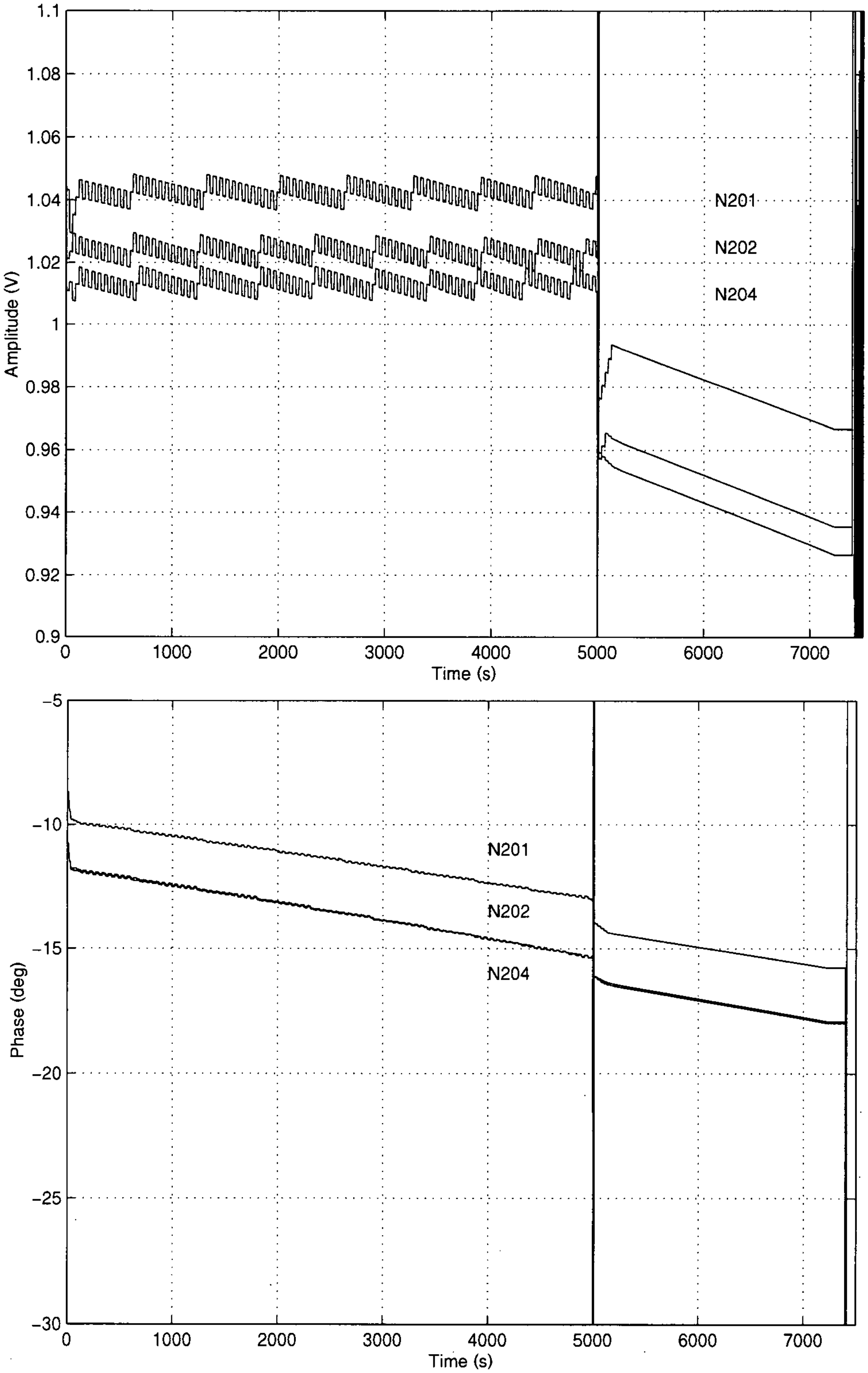


Figure 5.8 Dynamic voltage power flow solution at buses N201, N202, and N204 for the CIGRE 32-bus system of Figure 4.13 using voltage dependant loads.

CHAPTER 6: Conclusions and Recommendations for Future Work.

6.1 Summary and Conclusions

In Chapter 1 the description of some of the basic techniques used in the simulation of electric power circuits was presented. The use of nodal analysis and MNA for the formulation of the circuit equations was presented, and the use of difference equations to convert the resulting system of differential equations into a set of algebraic equations was described. Also in Chapter 1 important properties of the difference equations, such as A-stability and distortion were presented. Finally, phasor analysis for the solution of linear AC circuits and Newton-Raphson iterations for the solution of non-linear power flow problems were described.

The development of adjustable difference equations, suitable for electric circuit simulators, was presented in Chapter 2. These new difference equations were derived from interpolating polynomials of a given order, and their coefficients were written as a function of an adjusting parameter k. By setting this parameter k to zero, the resulting difference equation becomes the BDF of the same order of the used interpolating polynomial. On the other hand, if the adjusting parameter is set to one, the well known Backward Euler formula is obtained. By setting the parameter k to values between zero and one, the adjustable difference equations can be attuned to meet some stability and accuracy criteria. For example, the adjustable difference equation resulting from a third order interpolating polynomial can be made A-stable with a k of about 0.14, and the resulting formula will exhibit accuracy and stability properties very similar to those of the well known trapezoidal formula.

When interpolating polynomials of order higher than four are used, the resulting adjustable difference equations can be tuned to a particular frequency, so that there is no distortion at such frequency. This results in difference equations that produce exact results for linear circuits working at one particular frequency, such as for example those arising in electric power circuits. Chapter 3 describes the derivation of these tuned formulas. The resulting tuned formulas use fixed time steps, considerably larger than those required by traditional formulas such as the Trapezoidal and Backward Euler rules for the particular case of simulating of one frequency circuits and result in faster and more accurate simulations. Additionally, these tuned formulas show very good stability properties, in particular the tuned formula derived from a seventh order interpolating polynomial, which is almost A-stable.

Also in Chapter 3, linear electric power circuits are simulated using the tuned formulas in the time-domain, and the results compared with the solution of phasor analysis, where a perfect match is obtained. The computational cost of the tuned time-domain simulation is compared against phasor analysis, and it was concluded that the linear tuned time-domain simulation will result in faster solutions when the number of nodes in the circuit being simulated is larger than the number of steps required for the time-domain transients to damp.

In Chapter 4 the solution of the static power flow problem using the tuned time-domain methods developed in Chapter 3 was achieved. Before doing so, suitable models for elements commonly found in static power flow cases were developed. These include transmission lines, fixed tap transformers, shunts, and in particular PQ loads and PV sources. PQ loads were modelled using a sinusoidal current source whose magnitude and phase constantly adapt so as to match the required complex power load. Similarly, PV sources were modelled with a sinusoidal voltage source for which the magnitude is known, and the phase is adjusted using a simple PI controller to match the required active power.

Also in Chapter 4, three standard static power flow problems were solved using the tuned time-domain simulator. These include the IEEE 14-bus, the CIGRE 32-bus, and the IEEE 57-bus cases, where the results match accurately those obtained using standard Newton-Raphson iterations. The computational cost of the tuned time-domain static power flow is then compared to standard Newton-Raphson iterations. It was determined that the tuned time-domain static power flow would be faster reaching a solution than the standard Newton-Raphson iterations for systems with many more PQ buses than PV buses.

Finally, in Chapter 4, the stability properties of the tuned time-domain static power flow, which is capable of producing usable results on and around the point of voltage collapse, is used to generate PV curves. An adjustable current source is used to represent a varying PQ load with constant power factor. The resulting voltage, for a given active power, is then obtained from which a load PV curve can be plotted. The resulting PV curves are validated against the results obtained using Newton-Raphson iterations and compared with published load PV curves where the results show good agreement.

In Chapter 5 the dynamic power flow solution using the tuned time-domain formulas is obtained. This is possible because the tuned time-domain formulas are not only exact for the tuned frequency but also for low frequencies near DC. Models for the ULTC transformers, time varying loads and switches used to simulate outages were developed.

Finally the dynamic power flow simulation of the CIGRE 32-bus system is presented and the results compared with published data, where once again good agreement is obtained with the results obtained using the tuned time-domain method.

6.2 Recommendations for Future Work

The results obtained in this research, due to their fundamental nature, in particular the tuned difference equations for the solution of differential equations, may have many applications on the solution of many other oscillatory problems found in nature. Concentrating into the scope of this report, this research can be further expanded by combining it with the research of other members and former members of the power group at UBC:

1. Marti in [43;44] extended the concepts of Kron's diakoptics [45] to multi-area Thévenin equivalents (MATE) in which each subsystem can be solved with a different solution technique while maintaining a simultaneous solution of the entire network. Linares in [46] developed highly efficient computer algorithms to implement MATE in real-time simulations. Hollman in [47] mapped MATE algorithms to a PC-cluster architecture with multiple processing nodes. Using MATE's methodology in which a large circuit is torn into smaller more manageable circuits, the solution of static and dynamic power flow problems using the tuned time-domain method can be greatly speed up.
2. Louie in [48] researched the modeling of electric power circuit loads. His research can be used to greatly improve the results obtained with the tuned time-domain method by replacing the simple load models presented in this research with more accurate models.

3. Moreira also in [44], studied the simulation of electric circuits using latency. In his research an electric circuit is split into sections with remarkable different time constants, where each section is solved independently with a different time-step and latter combined to obtain the correct overall solution. By using this solution methodology the tuned time-domain method can be used to simulate the slow changing part of the circuit, while EMTP-like methods would be used to solve the fast changing part of the circuit.
4. Finally, Lukic in [31] has demonstrated the feasibility of using time-domain simulations for the solution of many traditional power flow problems. The developed tuned formulas are also directly applicable to her research.

Bibliography

[1] Dommel, H. W., "Digital computer solution of electromagnetic transients in single and multiphase networks," *IEEE Transactions on Power Apparatus and Systems*, vol. pas-88, no. 4, pp. 388-399, 1969.

[2] Marti, J. R. and Lin, J., "Suppression of numerical oscillations in the EMTP," *IEEE Transactions on Power Systems.*, vol. 4 pp. 739-745, June 1989.

[3] Ascher, U. M. and Petzold, L. R., *Computer methods for ordinary differential equations and differential-algebraic equations* Philadelphia: Society for Industrial and Applied Mathematics, 1998.

[4] Curttis, C. F. and Hirschfelder, J. O., "Integration of stiff equations," *Proceedings U.S.National Academy of Science*, vol. 38 pp. 235-243, 1952.

[5] Dahlquist G.G, "A special stability problem for linear multistep methods," *BIT*, vol. 3 pp. 27-43, 1963.

[6] Gear, C. W., *Numerical initial value problems in ordinary differential equations* Englewood Cliffs, N.J.: Prentice-Hall, 1971.

[7] Wildund, O., "A note on unconditionally stable linear multistep methods," *BIT*, vol. 7 pp. 65-70, 1967.

[8] Spijker, M. N., "Stiffness in numerical initial-value problems," *Journal Of Computational And Applied Mathematics*, vol. 72, no. 2, pp. 393-406, Aug. 1996.

[9] Verwer, J. G., Blom, J. G., Van Loon, M., and Spee, E. J., "A comparison of stiff ode solvers for atmospheric chemistry problems," *Atmospheric Environment*, vol. 30, no. 1, pp. 49-58, Jan. 1996.

[10] Hairer, E. and Wanner, G., "On the instability of the BDF formulas," *SIAM Journal on Numerical Analysis.*, no. 20., pp. 1206-1209, 1983.

[11] Fornberg, B., "Calculation of Weights in Finite Difference Formulas," *SIAM Review*, vol. 40, no. 3, pp. 685-691, Sept. 1998.

[12] Fredebeul, C., "A-BDF: A Generalization of the Backward Differentiation Formulae," *SIAM Journal on Numerical Analysis*, vol. 35, no. 5, pp. 1917-1938, Oct. 1999.

[13] Hosseini, S. M. and Hojjati, G., "Matrix free MEBDF method for the solution of stiff systems of ODEs," *Mathematical And Computer Modeling*, vol. 29, no. 4, pp. 67-77, Feb. 1999.

[14] Wu, X. Y., "A sixth-order A-stable explicit one-step method for stiff systems," *Computers & Mathematics With Applications*, vol. 35, no. 9, pp. 59-64, May 1998.

[15] Bettis, D. G., "Stabilization of finite difference methods of numerical integration," *Celestial Mechanics*, vol. 2, no. 3, pp. 282, 1970.

[16] Shampine, L. F. and Reichelt, M. W., "The MATLAB ODE suite," *SIAM Journal on Scientific Computing*, vol. 18, no. 1, pp. 1, 1997.

[17] Trnka, O., Hartman, M., and Svoboda, K., "An alternative semi-implicit Euler method for the integration of highly stiff nonlinear differential equations," *Computers & Chemical Engineering*, vol. 21, no. 3, pp. 277, 1997.

[18] Cremascoli, P. and Gubian, P. Mathematical behavior of integration methods with non continuous first derivative devices in circuit simulation. 2, 1225-1228. 1993. Proceedings of 36th Midwest Symposium on Circuits and Systems.

[19] Gubian, P. and Zanella, M. Stability properties of integration methods in SPICE transient analysis. 2701-2704. 1991. IEEE International Symposium on Circuits and Systems.

[20] Alvarado, F. L., Lasseter, R. H., and Sanchez, J. J., "Testing of trapezoidal integration with damping for the solution of power transient problems," *IEEE Transactions on Power Apparatus and Systems.*, no. 102., pp. 3783-3790, Dec. 1983.

[21] Smith, J. M., *Mathematical modeling and digital simulation for engineers and scientists* New York: Wiley, 1977.

[22] Gear, C. W. and Osterby, O., "Solving ordinary differential equations with discontinuities," *ACM Transactions on Mathematical Software.*, vol. 10, no. 1, pp. 23-44, Mar. 1984.

[23] Calvino-Fraga, J. Implementation of a Real Time Simulator for Protective Relay Testing. 6-15-1999. The University of British Columbia. Master Thesis Report.

[24] Aprille, T. J., Jr. and Trick, T. N., "Steady-state analysis of nonlinear circuits with periodic inputs," *Proceedings of the IEEE*, vol. 60, no. 1, pp. 108, 1972.

[25] Perkins, B. K., Marti, J. R., and Dommel, H. W., "Nonlinear elements in the EMTP: steady-state initialization," *IEEE Transactions on Power Systems*, vol. 10, no. 2, pp. 593, 1995.

[26] Bergen, A. R. and Vittal, V., *Power systems analysis*, 2nd ed. ed. Upper Saddle River, NJ: Prentice Hall, 2000.

[27] Vlach, J. and Singhal, K., *Computer methods for circuit analysis and design* New York: Van Nostrand Reinhold, 1983.

[28] Chung-Wen, H., Ruehli, E., and Brennan, P. A., "The modified nodal approach to network analysis," *IEEE Transactions on Circuits and Systems*, vol. CAS-22, no. 6, pp. 504-509, June 1975.

[29] Chua, L. O. and Lin, P., *Computer Aided Analysis of Electronic circuits: Algorithms & computational techniques.* Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1975.

[30] Dommel, H. W., *EMTP Theory Book.*, Second ed. Vancouver, BC: Microtran Power System Analysis Corporation, 1996.

[31] Lukic, M., *Feasibility of security assessment in power systems using full time-domain solutions in the EMTP* Vancouver: University of British Columbia. Master Thesis Report, 2000.

[32] Burden, R. L., Reynolds, A. C., and Faires, J. D., *Numerical analysis*, 2d ed. Boston, Mass.: Prindle, Weber & Schmidt, 1981.

[33] Kundur, P., *Power system stability and control* New York: McGraw-Hill, 1994.

[34] "Common format for exchange of solved power flow data," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-92, no. 6, pp. 1916-1925, Nov. 2000.

[35] Grainger, J. J. and Stevenson, W. D., *Power system analysis* New York: McGraw-Hill, 1994.

[36] Freris, L. L. and Sasson, A. M., "Investigation of the load-flow problem," *Proc.IEE*, vol. 115, no. 10, pp. 1459-1470, Aug. 1968.

[37] Voltage Stability Assessment: Concepts, Practices and Tools. Canizares, C. Editor. IEEE/PES Power System Stability Subcommittee Special Publication. 2002.

[38] Nagao, T., Tanaka, K., and Takenaka, K., "Development of static and simulation programs for voltage stability studies of bulk power system," *IEEE Transactions on Power Systems*, vol. 12, no. 1, pp. 273-281, Feb. 1997.

[39] Van Cutsem, T., Jacquemart, Y., Marquet, J. N., and Pruvot, P., "A comprehensive analysis of mid-term voltage stability," *IEEE Transactions on Power Systems*, vol. 10, no. 3, pp. 1173-1182, Aug. 1995.

[40] Westinghouse, E. C., *Electrical transmission and distribution reference book*, [4th completely rewritten ed.] East Pittsburgh, 1950.

[41] Berg, G. J., "Power-system load representation," *Proceedings of the Institution of Electrical Engineers*, vol. 120, no. 3, pp. 344-348, Mar. 1973.

[42] Indices Predicting Voltage Collapse Including Dynamic Phenomena. Hatziargyriou, N. D. and Van Cutsem, T. technical report TF 38-02-11, CIGRE. 1994.

[43] Marti, J. R. Multi-area thevenin equivalent. A circuit concept. Internal memo to H.W.Dommel and L.R.Linares.Department of Electrical Engineering, The University of British Columbia. 1993.

[44] Marti, J. R., Linares, L. R., Hollman, J. A., and Moreira, F. A. OVNI: Integrated software/hardware solution for real-time simulation of large power systems. Conference Proceedings of the 14th power system computation conference PSCC02 . 2002. 6-24-2002.

[45] Kron, G., "Tensorial analysis of integrated transmission systems, part III - the primitive division," *AIEE Transactions*, vol. 71 pp. 814-821, 1952.

[46] Linares-Rojas, L. R., *OVNI (Object Virtual Network Integrator)* Vancouver: University of British Columbia. Ph. D. Thesis Report, 2000.

[47] Marti, J. R., Hollman, J. A., and Calvino-Fraga, J. Implementation of a real-time distributed network simulator with PC-cluster. 223-227. 2000. Trois-Rivieres, Que., Canada, IEEE Comput. Soc. Proceedings International Conference on Parallel Computing in Electrical Engineering. PARELEC 2000.

[48] Louie, K. W., *Aggregation of voltage and frequency dependent electrical loads* Vancouver: University of British Columbia. Ph. D. Thesis Report, 1999.