Signal Processing for Distribution Network Monitoring

Ph.D. Thesis

Kåre Jean Jensen

LYNGBY 1999

IMM-PHD-1999-60

IMM

Signal Processing for Distribution Network Monitoring

Ph.D. Thesis

INDUSTRIAL RESEARCH EDUCATION PROGRAMME EF618

NESA A/S

DEPARTEMENT OF MATHEMATICAL MODELING,
TECHNICAL UNIVERSITY OF DENMARK

DEPARTEMENT OF AUTOMATION,
TECHNICAL UNIVERSITY OF DENMARK

DEFU

ACADEMY OF TECHNICAL SCIENCES

Funded by Sjællandssamarbejdet, NESA A/S,
and The Danish Agency for Trade and Industry

NESA A/S
Strandvejen 102
2900 Hellerup
Denmark

Technical University of Denmark
2800 Lyngby
Denmark

ISBN 87-89038-11-8

In-house publishing, NESA A/S, Strandvejen 102, DK-2900 Hellerup

Contents

Preface

This thesis is the final documentation for the Ph.D. project called "*Signal Processing for Distribution Network Monitoring*". The project has been accomplished under the Industrial Research Education Programme as a cooperation between the power distribution utility, NESA A/S, Research & Development Department, and Department of Mathematical Modeling (IMM), Technical University of Denmark (DTU) in the period from May 1996 to April 1999. The project is part of a research activity at NESA called DISMO which is an acronym for **Dis**tribution Network **Mo**nitoring, and is a project in monitoring medium voltage power distribution networks. This Ph.D. project directly follows the Ph.D. project "*Centralized Monitoring of 10 kV Cable Based Radial Distribution Networks*" by Steen M. Munk [Munk, 1995].

This project has been administered by The Academy of Technical Sciences (ATV), and had two institutions connected as third party: Department of Automation (IAU), DTU and Department of Development and Research in Power Distribution, DEFU.

The project steering committee was: Steen M. Munk (NESA), John Aasted Sørensen (IMM), Henrik Weldingh (DEFU), and Morten Lind (IAU).

Structure of Thesis

The project consisted of both a theoretical and a practical part. The theoretical part involved mathematical and numerical simulation models of power system signals and a proposition for a ground fault localization algorithm. The practical part of the project consisted of acquisition of normal operation signals and a ground fault experiment. The normal operation signals was broad-band acquisitions of voltage and current on a network supplying both industrial and domestic customers. The ground fault experiment was performed on a specially designed full scale laboratory with both medium voltage network and distribution transformers.

The first part of the thesis is an introduction to problem of distribution network monitoring and an overview of the normal operation signals. The mathematical and numerical models are discussed next and this theoretical part of the thesis is concluded by a chapter describing the proposed ground fault localization algorithm. Following is a chapter describing the ground fault experiments and finally a conclusion. The various models and program source code is documented in the appendices together with details on the ground fault experiments.

Following list is a brief overview of the contents of the chapters and appendices of this thesis.

Chapter 1 is an introduction to the problem of monitoring distribution networks, and introduces the concept of *centralized monitoring*.

Chapter 2 provides examples of the activity normally seen on the distribution network. The DISMO-PC and the DISMO-toolbox is introduced.

Chapter 3 reviews the necessary mathematical network analysis.

Chapter 4 describes the tools and models used for numerical simulation of distribution network signals.

Chapter 5 derives the algorithm proposed by the author for a ground fault localization system.

Chapter 6 describes the large scale ground fault experiments performed in autumn 1998. Examples of the acquired data is given.

Chapter 7 provides the conclusions of this project and suggestions for future work.

Appendix A discusses a deconvolution algorithm and a digital integrator for the DISMO-PC signals.

Appendix B derives a transfer function for a Π-section cable model.

Appendix C documents the ATP simulation models used in this thesis.

Appendix D provides an overview of the Matlab functions written during this project and discusses programming specific details of the DISMO-toolbox.

Appendix E describes the input/output format of the C++ utility programs written during this project.

Appendix F provides details on the ground fault experiments during the Autumn 1998.

Typography and Naming Conventions

The polar components of a function in the complex plane are often called the magnitude and the phase of the function. In power systems the term phase has another meaning, so in this thesis the polar components will be called the *magnitude* and the *argument*. The term *phase* will exclusively be used for the three components of the electrical power system voltage and current — the three phase system.

In different countries the phases of the three phase system have different names. This thesis will follow the Danish convention and call the three phases: phase R, phase S, and phase T.

The various names and variables have been typeset differently from the rest of the text. The following list shows the typographic conventions for the various text and the mathematical symbols.

Text symbols	
trademark	TRADEMARK
program name	ProgName
file type	`EXT`
file name	`/dir/name.ext`
geographical location	*Geographical Location*
ATP routine	*`ATP ROUTINE`*
ATP variable	`ATPVAR`
Mathematical symbols	
scalar	x
vector	$\mathbf{v}$
matrix	$\mathbf{M}$
convolution operator	$*$

Acknowledgments

During this project help has been provided from many people not directly connected to the project. This is highly appreciated and in this connection I would like to express my thanks to people at both NESA, ABB and DTU. In particular Janne Kaiberg and Bjarne Bendtsen from The Systems Operation Department at *Glentegården*, NESA, have been very helpful regarding the maintenance of the data acquisition system that was installed in the 10 kV transformer station at *Glentegården*.

The Department of Electric Power Engineering, ELTEK, DTU, has provided assistance on several occasions throughout this project, both with their expertise and their facilities. This has been a great help and is greatly acknowledged.

During autumn 1998 a large scale experiment on medium voltage ground faults was performed, and I would like to express my special thanks to Lars Nordin (ABB Corporate Research), Bernth Kjettrup (NESA),

Kaj Hoffmann Nielsen (NESA), and Ragnar Kristjánsson (NESA). Without these people the experiments would have been impossible to perform. Much of the equipment that was used during the experiments, was supplied by ABB Corporate Research, which also provided large resources to support the experiments.

I would also like to express my thanks to Professor Jeng-Nenq Hwang and Ph.D. student Dongxiang Xu at the Electrical Engineering Department at University of Washington who made it possible for me to visit University of Washington, Seattle, USA for external research.

Last but not least I would like to thank Steen M. Munk, John Aasted Sørensen, and Jesper Raaberg for proof reading and valuable comments during the process of writing this thesis and to Hugh Matthews of WEEKLY PRODUCTIONS for correcting my bad English.

December 1999

Kåre Jean Jensen

Abstract

A new representation of a distribution network is presented, where the network is modeled by a set of impulse responses referring to a number of equidistant locations along the network. This allows for using standard signal processing tools for estimation instead of simulation tools, some of which are computationally very demanding. This was published in a paper at the International Conference on Acoustics, Speech, and Signal Processing 1998 (ICASSP'98), Seattle, USA [Jensen *et al.*, 1998].

Using this representation of a distribution network, a method of estimating the location of a ground fault in a branched compensated radial distribution network is proposed. The method uses only measurements of voltage and current at the primary substation, and network data which is generally available at the utility data base. It is assumed that a valid model of the current in the fault, and a circuit model of the network exist. The method is verified successfully on simulated data. Experimental data acquired during the project sustains the potential of the method. It was not possible to perform an actual localization test on the experimental data within the time frame of the project. The results however, indicate that this will be possible with an improvement of the network models.

During the project a number of broad-band signals (10 kHz) were acquired at a distribution network in normal operation. Each of the signals had a duration of 45 minutes and were acquired at five different times during the day. This network supplies both industrial and domestic customers and no changes was made to the system during the measurements. The signals were analyzed and some initial classifications of transients were performed.

Full scale experiments of ground faults on medium voltage distribution networks were performed at a specially designed laboratory. The network at the laboratory consists of approximately 4.5 km of cable and 2.5 km overhead line. The ground faults were emulated by connecting a 10 kV phase to ground using a high precision controllable switch. Voltage and current signals were acquired at the feeding point of the network using both measurement transformers, probes and Rogowski coils.

Resumé

En ny representation for et distributions netværk præsenteres, hvor netværket modelleres ved et sæt impulsresponser, der refererer til et antal ækvidistante placeringer langs netværket. Dette giver mulighed for at anvende standard signalbehandlingsværktøjer til estimering istedet for simuleringsværktøjer, hvoraf nogle er beregningsmæssigt meget krævende. Dette er publiceret i en artikel ved International Conference on Acoustics, Speech, and Signal Processing 1998 (ICASSP'98) i Seattle, USA [Jensen *et al.*, 1998].

Ved hjælp af denne repræsentation af et distributionsnetværk, fremsættes en metode til lokalisering af en jordfejl i et forgrenet, kompenseret radialt distributionsnetværk. Metoden benytter sig kun af målinger af strøm og spænding ved hovedstationen, og netværksdata som normalt er tilgængelig for distributionsselskabet. Det antages at en gyldig model af strømmen i fejlstedet, og en kredsløbsmodel af netværket eksisterer. Metoden er verificeret på simulerede data med succes. Eksperimentelle data understøtter metodens potentiale, men et egentligt lokalisations eksempel var det ikke muligt at udføre indenfor projektets tidsramme. Resultaterne indikerer dog at dette vil være muligt med forbedringer af netværksmodellen.

Under projektet blev der foretaget dataopsamling af et antal bredbåndssignaler (10 kHz) på et distributionsnet i normal drift. Signalerne har en længde af 45 minutter hver og blev opsamlet på fem forskellige tidspunkter af døgnet. Dette distributionsnet forsyner både industri og beboelse og ingen ændringer blev foretaget af nettet under målingerne. Signalerne er analyseret og en indledende klassifikation af transienter blev udført.

Der blev udført forsøg med jordfejl på et mellemspændings-distributionsnet på et specielt designet laboratorium. Nettet på laboratoriet består af omkring 4.5 km kabel og 2.5 km luftledninger. Jordfejlene blev emuleret ved at forbinde en 10 kV fase til jord gennem en kontrollerbar højpræsitionskontakt, og spændinger og strømme blev opsamlet ved den forsynende transformer ved hjælp af både måletransformere, prober og Rogowski spoler.

Chapter 1

Introduction

1.1 Background

Power delivery has up until now been a monopoly business. This situation is changing these years where a deregulation process gradually moves the choice of supplier from an area-determined matter to the free decision of the customer. In other words the customer will in future be free to choose the supplier of electrical power. This is analogous to the situation on the telecommunication marked.

In this new marked of competition it is essential to optimize resource consumption and in order to do this, detailed monitoring of the power delivery system is essential.

Power delivery networks consists basically of two components — the *transmission network* and the *distribution network*. The transmission network transports the energy from the power generation units to the local area of the customers, and the distribution network is the link between the transmission network and the customer. These two network types are complex structures of network elements such as overhead lines, underground cables, transformers, switch gear, etc. Even though they consist of the same type of elements, they have very different properties. Throughout this thesis the different voltage levels, low voltage, medium voltage, and high voltage, will be denoted LV, MV, and HV respectively and the definition in [Lakervi and Holmes, 1989] is adopted where LV is below 1 kV, MV is in the range 1 kV–36 kV, and HV is above 36 kV.

The transmission network operates at HV level, it transports the energy over long distances and has relatively few nodes. The network is carefully monitored at all nodes in the network because an outage in this network affects a large number of customers, as it is the backbone of the power delivery system.

The distribution network operates at LV and MV level and connects each customer to the transmission network. The distribution network has a relatively large number of nodes and few customers are supplied through each node compared to the transmission network. This means that the cost of monitoring equipment as a price per customer, is much higher in the distribution network than in the transmission network, if monitoring equipment was going to be installed at all nodes. In [Munk, 1995] the following was concluded on a monitoring system based on measurements at all transformer stations: »It was estimated that a full implementation of a DSO-project[1] would require an investment of DDK 200 mill. (1992 prices) in NESA's area of operation alone; 500 000 customers supplied from 60 substations and 6 000 transformer stations. Such an investment would seem hard to motivate«.

1.2 The NESA Distribution Network

The distribution network in NESA's area of operation supplies approximately 500 000 customers at 0.4 kV level. These customers are supplied from approximately 5800 10 kV/0.4 kV transformer stations, and these are again supplied from the primary substation by approximately 600 feeders. The 10 kV/0.4 kV distribution transformer stations are called the

[1]The DSO-project, 1991–1994, was a project on control and monitoring of distribution networks accomplished as a cooperation between several Danish power distribution utilities.

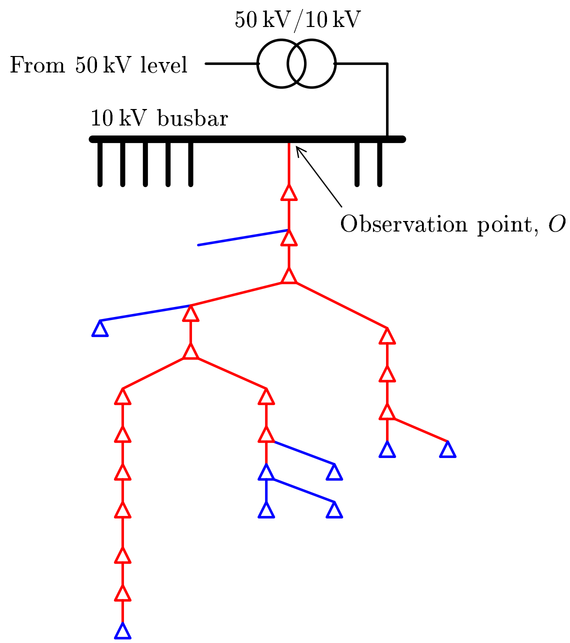


Figure 1.1: Schematic outline of A12 at *Glentegården*

secondary substations. The total length of NESA's 10 kV distribution network is 4013 km. Figure 1.1 shows an example of a typical feeder at NESA. The triangle symbols designates 10 kV/0.4 kV distribution transformer and the busbar is located at the primary transformer station. The blue sections of the network mark the borders to the neighbor feeders. The network is grounded through a Petersen coil (see Section 4.3) at the primary substation. Note that the figure is only an outline that describes the interconnection of the elements. The distance between the distribution transformers is not reflected by the figure.

1.3 The Centralized Monitoring Concept

As described in Section 1.1 a detailed monitoring system in the distribution network represents a very large investment. This leads to the idea of *centralized monitoring*: instead of a large number of data acquisition points with a low band width and a low level of signal processing, data is acquired from few central located high bandwidth observation points. The idea is that the information from the missing observation points can be replaced by utilizing signal processing tools on the high bandwidth data [Munk and Sørensen, 1997]. In Figure 1.1 the observation point O for the feeder is located by the primary substation busbar.

1.4 Ground Fault Localization

The aim of this project is to develop a general monitoring system that is able to provide information about all events on the distribution network such as, faults, decentral power generation (windmills), sudden changes in the load, etc.

After the initial study of normal operation signals it was decided, however, to concentrate on the problem of localizing a ground fault in a cable network, since that is the single most frequent cause of outage in the distribution network.

A *ground fault* is in this thesis defined as a (possibly) high impedance connection between one MV phase and ground. That is, only single phase to ground faults are considered. A ground fault on the cable network is typically caused by worn out insulation. As the network is grounded through a Petersen coil this will not immediately cause an outage. The cable will heat up and eventually develop a short circuit. This might take seconds and it might take hours. If the ground fault can be found before this occurs, the faulted section of the feeder can be disconnected without interruption of the supply to the customers, and an outage have been prevented.

Today, localization of ground faults is a highly manual trial-and-error

process, where a person at the system operation directs another person in the field which operates the non-automated circuit breakers. This can be a time consuming process which requires experience with the exact network in question.

This makes it interesting to automate this process, because the faster the fault can be found, the more outages can be prevented. The ground fault localization problem is the topic of Chapter 5.

Chapter 2

Normal Operation Feeder Activity

In order to design a monitoring system for the distribution network, it is important to have some knowledge about the activity that can be expected during normal operation. Therefore a set of voltage and current sensors were installed at a 50 kV/10 kV transformer station at *Glentegården*. *Glentegården* is a main transformer station located at *Buddinge* just North of Copenhagen. This 50 kV/10 kV transformer station has previously been used in the DISMO project to provide experimental data [Munk, 1995]. One specific feeder, A12, was selected as the research object because it has a well balanced distribution of industrial and domestic customers, and may be regarded as representative for the distribution network in general.

2.1 The Combi-Sensors

Three combi-sensors were installed at *Glentegården* at the 10 kV feeder A12 by the transformer station busbar. The sensors are from ABB in Finland and are high bandwidth, compact, combined voltage and cur-

Figure 2.1: The combined voltage and current sensor, the combi-sensor, from ABB.

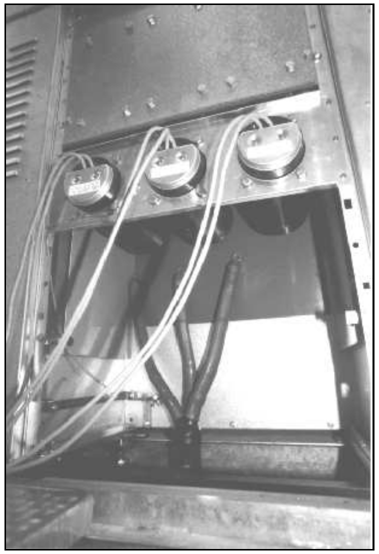


Figure 2.2: The combi-sensors installed at *Glentegården*.

rent sensors. The combi-sensor is shown in Figure 2.1 and described in [Mähönen *et al.*, 1996]. Figure 2.2 shows the sensors mounted at *Glentegården* just prior to the connection of the cables to A12.

The voltage sensor is based on a resistive voltage divider and the current sensor is a Rogowski coil. The Rogowski coil gives a voltage that is proportional to the derivative of the current through the sensor. This means that integration of the signal is required in order to derive a signal that represents the current. This operation is not performed in the sensor itself so it has to be implemented as a part of the acquisition system or the data processing system.

An ideal integrator has the Laplace transform $\frac{1}{s}$ which means that it has infinite magnitude response at DC. An attempt was made to design an

analog approximation to the integrator, but the large amplification at low frequencies combined with a requirement of an integration effect in the decade below 50 Hz complicated the design. Instead a digital integrator was implemented as an infinite impulse response (IIR) filter. The design of the IIR filter and consideration of two other possible designs are discussed in Appendix A.2.

All implementations of an integrator will include some kind of approximation compared to the ideal integrator, and different purposes may require different approximations. An advantage of the digital integrator in that respect, is that the choice of approximation does not have to be made at the time of the acquisition. The Rogowski coil signal can be acquired and stored directly, and the integration of the signal can be applied when the signal it is processed.

2.2 The DISMO-PC

The acquisition system for the combi-sensors is called the DISMO-PC. It is a PC running DOS with a 16 bit A/D-board installed. The A/D-board is capable of simultaneous sampling six channels at 20 kHz. The software used to control the A/D-converter is from DATA TRANSLATION and is called GlobalLab. This software has both a graphic interface and a macro capability so it is possible to run batch jobs. Figure 2.3 shows the DISMO-PC on a shelf above the walkway in the 10 kV transformer station at *Glentegården*. Figure 2.4 is an enlargement that shows the screen on one shelf and above that, the mini tower cabinet of the computer.

A modem and remote control software provides easy access to the DISMO-PC, so it is not necessary to actually be present in the transformer station in order to activate a data acquisition. This is a practical arrangement as a transformer station is a restricted area for which special access rights are required.

A program that is able to start a data acquisition at a specific time has been written. The program is called timer and is described in Appendix E.4 on page 168. A list of data acquisition requests is given in a

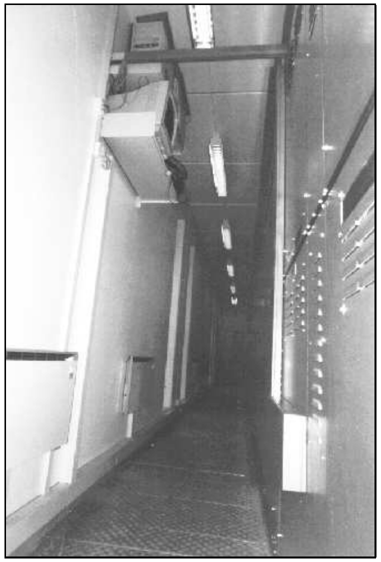


Figure 2.3: The DISMO-PC on the shelf above the walkway in the transformer station.

Figure 2.4: A closer look at the DISMO-PC.

file which is read by timer. When the system time reaches the given time, the specified acquisition is started. In this way it is easy to make a large number of acquisitions at a well defined time of the day (or night).

The modem connection can also be used to transfer small data acquisitions. At a sampling frequency of 20 kHz and with a three phase voltage and current acquisition, a signal corresponding to approximately 5 min. can be transferred on a normal telephone connection during a night.

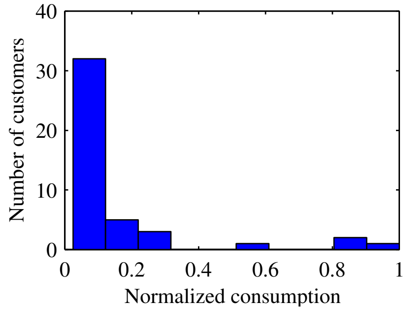


Figure 2.5: Histogram of the 44 largest customers at A12.

2.3 Customers at the Feeder

A database at NESA was searched for information on the customers at A12. This data base contains information on how much electricity each customer uses on a long term basis. Care has been taken to normalize data in this section so that no confidential information is exposed. A histogram of the annual consumption of the 40 largest customers is shown in Figure 2.5. The horizontal axis is normalized with the consumption of the largest customer.

For the large customers the database has information on the load variation during the day and night. The average load variation during 24 hours of the seven largest customers is shown in Figure 2.6. The data is given for each half hour and it has been normalized so that it has a mean value of one. The period with the largest load is, not surprisingly, from 7^{00} to 15^{00} — i.e. normal working hours. The level of the load during the day is approximately 3 times the load during the night.

Figure 2.7 shows the *rate of change* of the load variation in Figure 2.6, computed as the difference between one sample and the next on the horizontal axis. As the load variation is given on a half hourly basis, this plot only gives a very rough view of the changes as a function of time. It is

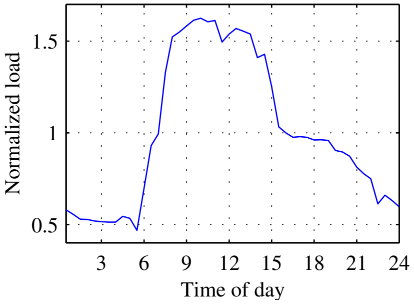


Figure 2.6: Normalized load variation over 24 hours.

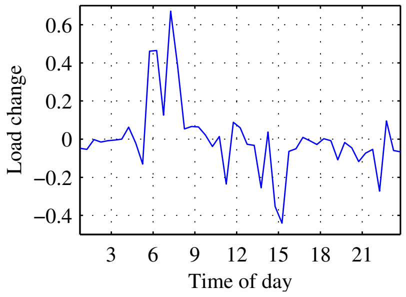


Figure 2.7: Rate of change of the load

seen from the figure that the period with the largest activity (the largest changes in the load) is between 5^{00} and 8^{00}.

2.4 Normal Operation Data

The information on the activity on A12 was used to make 5 long term data acquisitions in periods of both high and low activity (3^{00}, 7^{00}, 10^{00}, 12^{00}, and 18^{00}). The DISMO-PC was used for the acquisitions which each had a duration of 45 minutes. With six signals (three phase voltage and current), 16 bit precision, and a sampling frequency of 20 kHz, each acquisition results in approximately 630 MB of data.

The data was analyzed with the DISMO-toolbox which is specially designed to analyze voltages and currents from power distribution systems. The individual parts of the DISMO-toolbox was developed during four Masters projects, [Høg, 1994], [Jespersen, 1994], [Nielsen, 1995], and [Madsen, 1996].

The toolbox can be used to estimate the instantaneous amplitude and frequency of the fundamental sinusoidal power component. In addition a residual signal is computed as the input signal minus the estimated

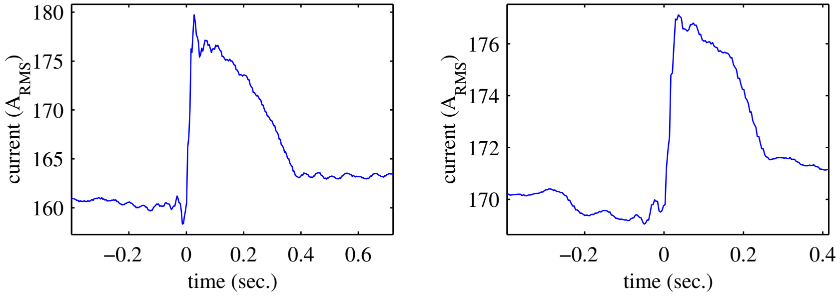


Figure 2.8: Detected transients assumed to originate from the start of a motor.

fundamental component. This residual signal can then be used as input to a Wavelet transformation for a time-frequency analysis.

The DISMO-toolbox is implemented as a Matlab toolbox and is a central development and demonstration framework for the DISMO project. The DISMO-toolbox is described in Appendix D.8 on page 148.

In [Gunnarsson, 1998] all five 45 min. data acquisitions was analyzed using the DISMO-toolbox. This resulted in more than 200 detected transients having a rate of change of more than 1.5 A/ms. In [Munk, 1995] the starting and stopping of a large three phase motor is treated in detail and approximately 75 % of the detected transients falls into this category, [Gunnarsson, 1998].

Two of these transients are shown in Figure 2.8 which are the amplitude estimate computed by the DISMO-toolbox. The characteristic of this class of transients is that the current raises very fast when the motor is turned on. As the motor is running up to speed, the current falls at a lower rate to a steady state level. This sequence has a time duration in the order of a tenth of a second.

Figure 2.9 shows two transients of unknown origin but with a characteristic shape. This is also a common class of transients.

More than 90 out of the 200 detected transients are from the acquisition made at 7^{00} hours, so this supports the conclusion made on the rate

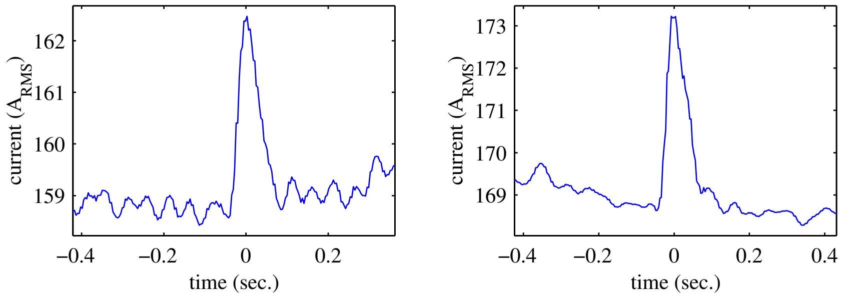


Figure 2.9: Detected transients of unknown origin.

of change of the load in Figure 2.7.

In terms of the height of the transients, the figures are representative as most transients are about 5–10 A in height and a few are 20 A. Note the steady state level of the current which is 160–170 A. A change in the amplitude of 5 A over several periods would hardly be noticeable at this level in the sinusoidal current signal. With the DISMO-toolbox these transients are easily detected.

Chapter 3

Mathematical Network Analysis

3.1 Symmetrical Components

The symmetrical components of a three phase power system are a well treated subject in the literature. It is a method of decomposing the coupled three phase system into three uncoupled one phase systems. These three systems are called the positive, the inverse and the zero sequence and are named V_1, V_2, and V_0 respectively in the case of voltages. The transformation is defined as [Lakervi and Holmes, 1989]

$$\begin{bmatrix} V_1 \\ V_2 \\ V_0 \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 1 & a^2 & a \\ 1 & a & a^2 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} V_R \\ V_S \\ V_T \end{bmatrix}, \quad a = \mathrm{e}^{-j\frac{2\pi}{3}} \tag{3.1}$$

This is a frequency domain representation of the transformation, and it also most often used when considering sinusoidal excitation. A transformation to the time domain would require two all-pass filters with the transfer function $\mathrm{e}^{-j\frac{2\pi}{3}}$ and $\mathrm{e}^{-j\frac{4\pi}{3}}$. By applying these two filters to the

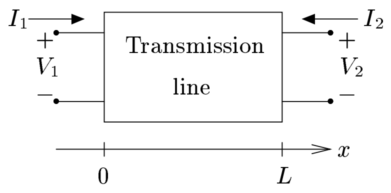


Figure 3.1: Single phase ideal transmission line with distributed parameters.

relevant phase components the symmetrical sequences could be derived in the time domain. For the zero sequence the transformation is simply the sum of the phase components directly, as given by Equation 3.1.

3.2 Distributed Parameter Transmission Line

In this section, the admittance matrix of an ideal two wire transmission line will be derived. Not all details will be given explicitly as it is regarded to be out of the scope of this thesis.

Figure 3.2 shows the definition of voltage and current at the two ports of the transmission line. The parameters for the transmission line are defined in Table 3.1 under *Primary parameters*.

The voltage and current on the transmission line are given by the wave equation,

$$\frac{\partial^2 V}{\partial x^2} - \gamma^2 V = 0 \quad \text{and} \quad \frac{\partial^2 I}{\partial x^2} - \gamma^2 I = 0 \tag{3.2}$$

and at any given location, x, the relation between the voltage and the current is given by the characteristic impedance as

$$V_x = Z_0 I_x \tag{3.3}$$

The propagation constant γ and the characteristic impedance Z_0 are defined in Table 3.1 under *Modal parameters*.

Primary parameters	
Series resistance per meter	r
Series inductance per meter	l
Shunt conductance per meter	g
Shunt capacitance per meter	c
Length of transmission line	L
Phase parameters	
Series impedance	$z = r + j\omega l$
Shunt admittance	$y = g + j\omega c$
Total series impedance	$Z = zL$
Total shunt admittance	$Y = yL$
Modal parameters	
Wave propagation constant	$\gamma = \sqrt{zy}$
Characteristic impedance	$Z_0 = \sqrt{\frac{z}{y}}$

Table 3.1: Definition of transmission line parameters.

If the voltage and current for $x = 0$ is V_1 and I_1 respectively, as shown in Figure 3.2, the solution to the wave equation at any location x is given by

$$\begin{aligned} V_x &= \frac{1}{2}(V_1 + Z_0 I_1)e^{-\gamma x} + \frac{1}{2}(V_1 - Z_0 I_1)e^{\gamma x} \\ I_x &= \frac{1}{2}(\frac{V_1}{Z_0} + I_1)e^{-\gamma x} + \frac{1}{2}(\frac{V_1}{Z_0} - I_1)e^{\gamma x} \end{aligned} \tag{3.4}$$

or expressed in terms of hyperbolic functions

$$\begin{aligned} V_x &= V_1 \cosh(\gamma x) - Z_0 I_1 \sinh(\gamma x) \\ I_x &= -\frac{V_1}{Z_0} \sinh(\gamma x) + I_1 \cosh(\gamma x) \end{aligned} \tag{3.5}$$

For $x = L$, this equation gives the relation between the voltage and current of the two ends of the cable in Figure 3.2.

$$
\begin{aligned}
V_2 &= V_1 \cosh(\gamma L) - Z_0 I_1 \sinh(\gamma L) \\
I_2 &= \frac{V_1}{Z_0} \sinh(\gamma L) - I_1 \cosh(\gamma L)
\end{aligned} \tag{3.6}
$$

Rearranging this equation, the currents can be expressed in terms of the voltages as

$$
\begin{aligned}
I_1 &= V_1 \frac{1}{Z_0} \frac{1}{\tanh(\gamma L)} - V_2 \frac{1}{Z_0} \frac{1}{\sinh(\gamma L)} \\
I_2 &= -V_1 \frac{1}{Z_0 \sinh(\gamma L)} + V_2 \frac{1}{Z_0 \tanh(\gamma L)}
\end{aligned} \tag{3.7}
$$

This equation expresses the voltages and currents in terms of the modal quantities Z_0 and γ. To introduce the phase quantities Z and Y we need to write the hyperbolic tangent in an alternative way as

$$
\frac{1}{\tanh(x)} = \tanh\left(\frac{x}{2}\right) + \frac{1}{\sinh(x)} \tag{3.8}
$$

and the characteristic impedance in two ways as

$$
Z_0 = \frac{Z}{\gamma L} = \frac{\gamma L}{Y} \tag{3.9}
$$

Using Equation 3.9 following equalities can be derived

$$
\begin{aligned}
\frac{1}{Z_0 \sinh(\gamma L)} &= \frac{1}{Z} \frac{\gamma L}{\sinh(\gamma L)} \\
\frac{\tanh(\frac{\gamma L}{2})}{Z_0} &= \frac{Y}{2} \frac{\tanh(\frac{\gamma L}{2})}{(\frac{\gamma L}{2})}
\end{aligned} \tag{3.10}
$$

The admittance matrix can now be derived from Equation 3.7 by inserting Equation 3.9 and 3.10.

$$
\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{Z}\frac{\gamma L}{\sinh(\gamma L)} + \frac{Y}{2}\frac{\tanh(\frac{\gamma L}{2})}{\frac{\gamma L}{2}} & -\frac{1}{Z}\frac{\gamma L}{\sinh(\gamma L)} \\ -\frac{1}{Z}\frac{\gamma L}{\sinh(\gamma L)} & \frac{1}{Z}\frac{\gamma L}{\sinh(\gamma L)} + \frac{Y}{2}\frac{\tanh(\frac{\gamma L}{2})}{\frac{\gamma L}{2}} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \tag{3.11}
$$

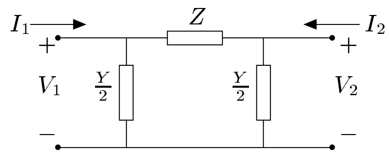


Figure 3.2: Model of single phase transmission line with lumped elements.

This is the admittance matrix for an ideal two wire transmission line with *constant* distributed parameters. The term *constant* parameter refers to the primary cable parameters r, l, g, and c. In general they cannot be assumed to be independent of frequency (see e.g. Figure 4.1 and 4.2 on page 24) so the results are in general only valid for the frequency of which the primary parameters were computed. They may, however, be varying slowly so in some limited frequency range Equation 3.11 may be a good approximation. Furthermore some frequency dependence is included through the parameters Z, Y, Z_0, and γ.

3.3 Lumped Element Cable Model

The admittance matrix in Equation 3.11 is well suited for frequency domain computations. If this matrix were to be transformed to the time domain, the solution would incorporate Bessel functions which are difficult to handle. A standard approximation to the ideal transmission line is the Π-equivalent shown in Figure 3.2. The lumped elements Z and Y are defined in Table 3.1. The admittance matrix that gives the relation between current and voltage in this model can be written as

$$
\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{Z} + \frac{Y}{2} & -\frac{1}{Z} \\ -\frac{1}{Z} & \frac{1}{Z} + \frac{Y}{2} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \tag{3.12}
$$

If this matrix is compared to Equation 3.11 it is clear that for a given angular frequency ω and corresponding Z, Y, and γ the Π-equivalent is valid if

$$Z_\Pi = Z\frac{\sinh(\gamma L)}{\gamma L} \quad \text{and} \quad Y_\Pi = Y\frac{\tanh(\frac{\gamma L}{2})}{\frac{\gamma L}{2}} \tag{3.13}$$

where Z_Π and Y_Π are Z and Y in Equation 3.12.

For low frequencies or short cable lengths, i.e. $\gamma L \approx 0$, the lumped elements of the Π-equivalent are given directly by Table 3.1. The terms of this condition are the subject of Section 4.7.1 on page 40.

Chapter 4

Numerical Simulations of Network Signals

This chapter discusses the basic properties of the models used for numerical simulation of data on power distribution networks in this project. The software used to perform these simulations is a version of ATP for Linux dated August 8$^{\text{th}}$ 1998. ATP, Alternative Transients Program, is a royalty-free version of the EMTP, Electro Magnetic Transient Program. The Linux version is distributed through the password protected internet site, `http://atp.pwr.eng.osaka-u.ac.jp/~jaug/index2-e.htm` in Japan. The manual for ATP (and EMTP) is called the *Rulebook* and is given in reference [Leu, 1987]. It contains a detailed description of input and output file formats. The mathematical background of ATP/EMTP is called the *Theorybook* and is given in reference [Dommel, 1981].

Modeling of electrical networks in general are not at all a simple task. If the frequency range under consideration is beyond the fundamental power frequency we deal with models and parameters which are varying with frequency. A Ph.D. project at NESA running from September 1998 to August 2001 focuses exclusively on modeling power systems up into the MHz-range. This project is accomplished by Ragnar Kristjánsson in a cooperation between NESA A/S, Department of Electrical Power Engineering, DTU, and Department of Applied Electronics, DTU and has the title "*Power Quality Modelling*". The project is administered by ATV and has the project number EF-744.

This chapter describes how each network element is modeled in ATP. The network models basically consist of three different elements: cables (and overhead lines), transformers, and generators. The cables will be treated in Section 4.1 and the transformers in Section 4.2.

These models are used for two types of simulations:

Impulse response generation. In this simulation type the source is a single phase impulse source and the network is static.

Ground fault simulation. In this simulation type the source is a three phase symmetrical sinusoid, and the network is changed during the simulation.

Section 4.5 describes the impulse source and Section 4.6 describes the ground fault simulations. Section 4.7 discusses the precision and the limitations of the models.

4.1 Cable Model

The challenge of modeling cables over a large frequency range in the time domain is that real world cables are distributed *and* frequency dependent parameter elements. So even if the ideal transmission line is used as described in Chapter 3, the parameters, r, l, and c, themselves are frequency dependent. ATP gives several possibilities of modeling cables which all have different strengths and drawbacks. The models implemented in ATP include distributed parameter line, lumped element Π-section, transformation matrix methods, and an ARMA model implementation.

4.1.1 Choice of Cable Model

The distributed parameter line implements the cable as a lossless line with the loss modeled as three lumped elements, one quarter of the loss at the ends of the line and half of the loss at the middle.

The lumped Π-section includes the loss, but is generally only valid for one single frequency.

The transformation matrix method is implemented in several ATP supporting routines such as *SEMLYEN SETUP* and *JMARTI SETUP*. This approach attempts to decouple a polyphase line into single phase lines by transforming the problem from the phase domain to the modal domain. The transformation matrix is assumed to be constant with respect to frequency but in general it is not, and this method is therefore always an approximation. For overhead lines, however, the approximation is good. See the Theorybook [Dommel, 1981] Section 4.1.5.3.

The *NODA SETUP* implements a cable model as an autoregressive moving average (ARMA) model. The method is described in details in reference [Noda *et al.*, 1996]. In filter theory an ARMA model is called an infinite impulse response (IIR) filter. It implements a transfer function with both poles and zeros very efficiently in the time domain.

The choice of model must be based on the specific application in question. In this context modeling of the distributed loss is considered an important factor, and as described in Section 5.2 on page 48, the network model should have nodes corresponding to a relatively large number of equidistant locations along the physical network. On this background the cascaded Π-sections were chosen as the cable model, since the loss is included in the model and the equidistant nodes is automatically implemented. The *NODA SETUP* is intended for modeling a cable section as one element. It has not been tested if the *NODA SETUP* model can be cascaded to produce the equidistant nodes.

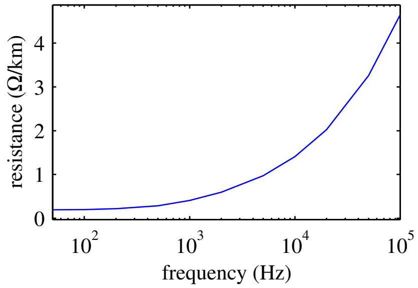


Figure 4.1: Positive sequence resistance for a Cu 95 mm^2 APB cable.

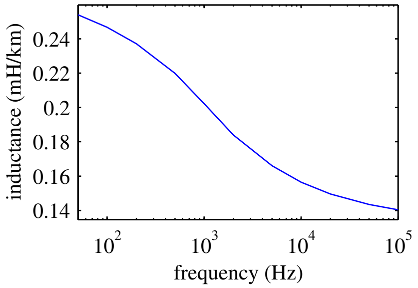


Figure 4.2: Positive sequence inductance for a Cu 95 mm^2 APB cable.

4.1.2 Parameters for the Cable Model

The cascaded Π-section model of ATP needs an impedance matrix that represent the particular cable type in question. For this purpose the supporting ATP routine *CABLE CONSTANTS* is used. *CABLE CONSTANTS* computes impedance matrices directly from cable material parameters and the geometric dimensions of the cable cross section. These impedance matrices can be used directly in the ATP model.

The impedance matrices are computed at one single frequency, meaning that in general the resulting Π-section is only valid at that specific frequency, even though frequency dependence is included in the model through the inductance and the capacitance. Figure 4.1 and 4.2 shows the positive sequence resistance and inductance for frequencies from 50 Hz to 100 kHz computed by *CABLE CONSTANTS*. The cable type is a 95 mm^2 copper APB cable. The capacitance is constant 0.29 μF. This frequency dependence is a function of porperties like *skin effect* and *proximity effect* [Allan, 1991].

CABLE CONSTANTS assumes that the core of the cable has a circular cross section. Some cables have sectionalized cores so in this case an approximation has to be made by keeping the thickness of the insulation, the cross sectional area of the core, the armor, and the pipe, and changing

diameters of the cores and the pipe. Alternatively the supporting ATP routine, *CABLE PARAMETERS*, can be used which makes no assumptions regarding the core cross section shape.

As the number of cable types is large and the ATP input files are tedious to write, a Matlab function has been written to generate a complete set of impedance matrices for a specified set of frequencies for both APB and PEX cables with both copper and aluminum cores. A single overhead line configuration is included. This function is described in Appendix D.7 on page 141.

4.1.3 Distributed Ground Resistance

ATP does not automatically include a distributed ground impedance although *CABLE CONSTANTS* does take the ground resistivity into account when the impedance matrices are computed. Under symmetrical conditions there will be no current flowing through ground apart from the current flowing through the network capacity. A ground fault, however, is a circuit path through ground and in this situation it is important how the ground is modeled. All references to ground in the ATP model is a reference to the same node, even though different ground points may be several kilometers apart in the real network. This node is called TERRA and is a common reference point for all voltages in the model. When a number of Π-sections all have some capacity to ground, all these capacitors are connected together with one terminal of the generator. If a ground fault is simulated on this model, it means that the fault location and the generator are connected together through a lumped element, whereas the real ground must be distributed by nature. On the other hand, a detailed model of the ground might be impossible to achieve, since the electrical parameters are highly dependent on soil properties and may vary over a wide range, and data would be needed which is not readily available. A compromise between these two situations is to add one extra branch, a *ground branch*, to the Π-section to represent the distributed ground impedance. In this way the network remains unchanged for symmetrical loads and only the ground fault current is affected.

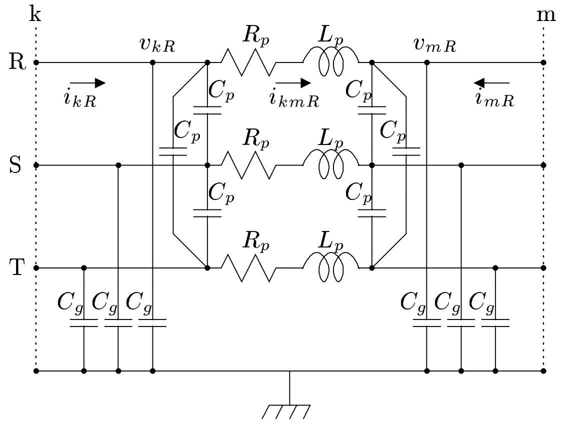


Figure 4.3: Π-section with no ground resistance

The rest of Section 4.1 is devoted to this modification of the standard three phase Π-section to approximate the distributed ground impedance.

4.1.4 Three Phase Π-section

In order to add a ground branch to the three phase Π-section without changing the three phases, it must be investigated how ATP interprets the impedance matrices. The topology is shown in Figure 4.3. According to the Rulebook, the relation between the three phase current vectors $\mathbf{i}_k$ and $\mathbf{i}_m$ and the voltage vectors $\mathbf{v}_k$ and $\mathbf{v}_m$ are given by Equation 4.1.

$$\begin{aligned} \mathbf{i}_k &= \frac{1}{2}\mathbf{C}\,\frac{\partial \mathbf{v}_k}{\partial t} + \mathbf{i}_{km} \\ \mathbf{i}_m &= \frac{1}{2}\mathbf{C}\,\frac{\partial \mathbf{v}_m}{\partial t} - \mathbf{i}_{km} \\ \mathbf{v}_k - \mathbf{v}_m &= \mathbf{L}\,\frac{\partial \mathbf{i}_{km}}{\partial t} + \mathbf{R}\,\mathbf{i}_{km} \end{aligned} \tag{4.1}$$

where

$$\begin{aligned} \mathbf{i}_k &= \begin{bmatrix} i_{kR} \\ i_{kS} \\ i_{kT} \end{bmatrix}, \quad \mathbf{i}_m = \begin{bmatrix} i_{mR} \\ i_{mS} \\ i_{mT} \end{bmatrix}, \quad \mathbf{i}_{km} = \begin{bmatrix} i_{kmR} \\ i_{kmS} \\ i_{kmT} \end{bmatrix} \\ \mathbf{v}_k &= \begin{bmatrix} v_{kR} \\ v_{kS} \\ v_{kT} \end{bmatrix}, \quad \mathbf{v}_m = \begin{bmatrix} v_{mR} \\ v_{mS} \\ v_{mT} \end{bmatrix} \end{aligned} \tag{4.2}$$

The currents i_{kR}, i_{mR} and i_{kmR} and the voltages v_{kR} and v_{mR} are shown in Figure 4.3. The 3-by-3 matrices, $\mathbf{R}$, $\mathbf{L}$, and $\mathbf{C}$, are computed by *CABLE CONSTANTS*.

The current i_{kR} going into the Π-section, can be expressed as a sum of the currents going to phase S and T through the capacitors C_p, the current going to ground through C_g and the current i_{km} as given by

$$\begin{aligned} i_{kR} &= C_g\,\frac{\partial v_{kR}}{\partial t} + C_p\,\frac{\partial(v_{kR} - v_{kS})}{\partial t} + C_p\,\frac{\partial(v_{kR} - v_{kT})}{\partial t} + i_{kmR} \\ &= (C_g + 2\,C_p)\,\frac{\partial v_{kR}}{\partial t} - C_p\,\frac{\partial v_{kS}}{\partial t} - C_p\,\frac{\partial v_{kT}}{\partial t} + i_{kmR} \end{aligned} \tag{4.3}$$

If this is repeated for the currents i_{kS} and i_{kT} and the expressions are compared to Equation 4.1, the capacitance matrix $\mathbf{C}$ can be identified as

$$\mathbf{C} = 2\begin{bmatrix} C_g + 2\,C_p & -C_p & -C_p \\ -C_p & C_g + 2\,C_p & -C_p \\ -C_p & -C_p & C_g + 2\,C_p \end{bmatrix} \tag{4.4}$$

From Equation 4.1 the voltage from node m to node k for phase R can be written as

$$\begin{aligned} v_{kR} - v_{mR} &= L_p\,\frac{\partial i_{kmR}}{\partial t} + L_m\,\frac{\partial i_{kmS}}{\partial t} + L_m\,\frac{\partial i_{kmT}}{\partial t} \\ &\quad + R_p\,i_{kmR} + R_m\,i_{kmS} + R_m\,i_{kmT} \end{aligned} \tag{4.5}$$

where L_m and R_m represents the mutual coupling between the cores of the cable, which are all assumed to be equal for reasons of symmetry. If this

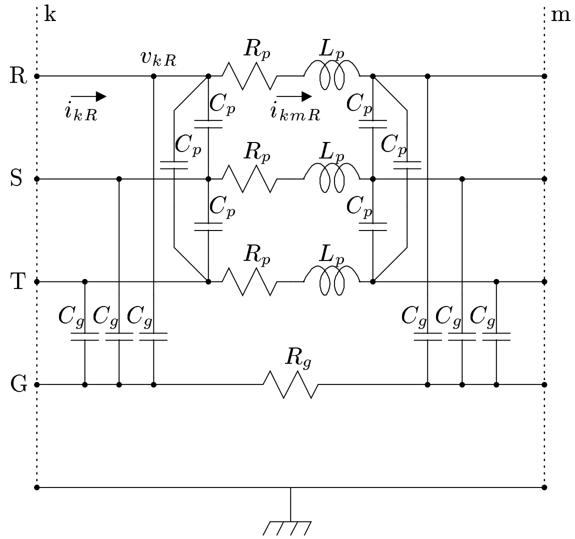


Figure 4.4: Π-section with ground resistance

is repeated for the voltage across phase S and T, the use of Equation 4.1 leads to identification of impedance matrices $\mathbf{L}$ and $\mathbf{R}$ as

$$\mathbf{L} = \begin{bmatrix} L_p & L_m & L_m \\ L_m & L_p & L_m \\ L_m & L_m & L_p \end{bmatrix} \qquad \mathbf{R} = \begin{bmatrix} R_p & R_m & R_m \\ R_m & R_p & R_m \\ R_m & R_m & R_p \end{bmatrix} \tag{4.6}$$

4.1.5 Π-section with Ground Resistance

If we include a ground branch to the Π-section as shown in Figure 4.4, the phases can be disconnected from the `TERRA` node.

Now the current and voltage vectors are given by

$$\begin{aligned} \mathbf{i}_k &= \begin{bmatrix} i_{kR} \\ i_{kS} \\ i_{kT} \\ i_{kG} \end{bmatrix}, \ \mathbf{i}_m = \begin{bmatrix} i_{mR} \\ i_{mS} \\ i_{mT} \\ i_{mG} \end{bmatrix}, \ \mathbf{i}_{km} = \begin{bmatrix} i_{kmR} \\ i_{kmS} \\ i_{kmT} \\ i_{kmG} \end{bmatrix} \\ \mathbf{v}_k &= \begin{bmatrix} v_{kR} \\ v_{kS} \\ v_{kT} \\ v_{kG} \end{bmatrix}, \mathbf{v}_m = \begin{bmatrix} v_{mR} \\ v_{mS} \\ v_{mT} \\ v_{mG} \end{bmatrix} \end{aligned} \tag{4.7}$$

Similar to Equation 4.3, the current i_{kR} can be expressed as a sum of the currents going to phase S, T, and G and i_{km} as

$$\begin{aligned} i_{kR} =& C_g \frac{\partial(v_{kR} - v_{kG})}{\partial t} + C_p \frac{\partial(v_{kR} - v_{kS})}{\partial t} + \\ & C_p \frac{\partial(v_{kR} - v_{kT})}{\partial t} + i_{kmR} \\ =& (C_g + 2\,C_p) \frac{\partial v_{kR}}{\partial t} - C_p \frac{\partial v_{kS}}{\partial t} - \\ & C_p \frac{\partial v_{kT}}{\partial t} - C_g \frac{\partial v_{kG}}{\partial t} + i_{kmR} \end{aligned} \tag{4.8}$$

The currents i_{kS} and i_{kT} can be derived similarly.

The current i_{kG} is the sum of the currents going to phases R, S, and T through the capacitors C_g and the current i_{kmG}.

$$\begin{aligned} i_{kG} &= C_g \frac{\partial(v_{kG} - v_{kR})}{\partial t} + C_g \frac{\partial(v_{kG} - v_{kS})}{\partial t} + C_g \frac{\partial(v_{kG} - v_{kT})}{\partial t} + i_{kmG} \\ &= -C_g \frac{\partial v_{kR}}{\partial t} - C_g \frac{\partial v_{kS}}{\partial t} - C_g \frac{\partial v_{kT}}{\partial t} + 3\,C_g \frac{\partial v_{kG}}{\partial t} + i_{kmG} \end{aligned} \tag{4.9}$$

If these expressions are compared to Equation 4.1 the matrix $\mathbf{C}'$ can be identified as

$$\mathbf{C}' = 2\begin{bmatrix} C_g + 2\,C_p & -C_p & -C_p & -C_g \\ -C_p & C_g + 2\,C_p & -C_p & -C_g \\ -C_p & -C_p & C_g + 2\,C_p & -C_g \\ -C_g & -C_g & -C_g & 3\,C_g \end{bmatrix} \tag{4.10}$$

From Equation 4.4 it is seen that the capacitance matrix $\mathbf{C}$ is included as the upper left 3-by-3 part of $\mathbf{C}'$. The capacitance matrix which includes the distributed ground resistance can therefore be expressed in terms of the capacitance matrix with no ground resistance as

$$\mathbf{C}' = \begin{bmatrix} & & & -2\,C_g \\ & \mathbf{C} & & -2\,C_g \\ & & & -2\,C_g \\ -2\,C_g & -2\,C_g & -2\,C_g & 6\,C_g \end{bmatrix} \tag{4.11}$$

If the mutual coupling between the cores and the ground are assumed to be zero, the fourth column and the fourth row of the inductance and the resistance matrix will contain only zero elements, except for the fourth diagonal element of $\mathbf{R}$ which contains the ground resistance R_g. The inductance matrix $\mathbf{L}'$ and the resistance matrix $\mathbf{R}'$ including ground resistance can thus be expressed in terms of $\mathbf{L}$ and $\mathbf{R}$ in Equation 4.6 as

$$\mathbf{L}' = \begin{bmatrix} & & & 0 \\ & \mathbf{L} & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{R}' = \begin{bmatrix} & & & 0 \\ & \mathbf{R} & & 0 \\ & & & 0 \\ 0 & 0 & 0 & R_g \end{bmatrix} \tag{4.12}$$

From Equation 4.11 and 4.12 it is seen that only C_g and R_g need to be found. Assuming that we know the capacitance matrix $\mathbf{C}$ in Equation 4.4 (can be computed with the *CABLE CONSTANTS* subroutine of ATP), C_g can be found as

$$C_g = \frac{c_{11} + 2c_{21}}{2} \tag{4.13}$$

where $\mathbf{C} = [c_{ij}]$. The ground resistance R_g has to be found by other means.

Open circuit test		Short circuit test	
P_o	Excitation loss	P_s	Short circuit loss
I_o	Excitation current	I_s	Short circuit current
U_o	Excitation voltage	U_s	Short circuit voltage

Table 4.1: Transformer test variables

4.2 Transformer Models

This section describes how a transformer can be modeled by an impedance matrix computed by the supporting ATP routine *BCTRAN*. The routine uses data from a standard transformer test as input. The transformer model is not included in the final network model as it does not seem to have much influence on the impulse response and ground fault simulations. As it will be described later in Section 6.6.4 some differences remain between the model and the experimental data, and it can not be excluded that a transformer model might have some significance. This section is therefore included for future reference.

A transformer test includes measurements of voltage, current and active power on one winding with the other winding either open or short circuited. A *winding* in this context means the three coils that make up one side of a three phase transformer. Only two-winding transformers will be considered here.

All voltages, U and currents, I in this section are RMS values, and can refer to either the primary or the secondary side of the transformer. All quantities in an expression, however, will refer to the same side of the transformer.

Table 4.1 lists the variables associated with a transformer test. These tests are generally performed for both a positive sequence and a zero sequence voltage. If the transformer has a delta connected winding, the zero sequence voltage will be short circuited. This means that the either the open circuit test will become a short circuit test, or the generator will be short circuited, depending on which side of the transformer has the delta. The zero sequence test is therefore not performed on a transformer that has a delta connected winding. *BCTRAN*, however, requires the data for both tests and according to the Rulebook, the positive sequence data can be copied to the zero sequence data. The zero sequence data will have no influence on the output. Only delta/star (Dy) connected transformers exist in the distribution networks treated in this thesis, so this is the only transformer type that will be considered here.

The nominal value of the apparent power, or the *power base*, of a three phase transformer can be written as

$$S_N = 3U_{N,l}I_N \tag{4.14}$$

Subscript N denotes nominal value and subscript l and p denotes line voltage and phase voltage respectively. The line voltage U_l is the voltage between one phase and ground, and phase voltage U_p is the voltage between two phases. A voltage is assumed to be phase voltage if nothing else is given. The relation between phase and line voltage under symmetrical conditions is

$$U_p = \sqrt{3}U_l \tag{4.15}$$

In terms of nominal phase voltage, the nominal apparent power is written as

$$S_N = \sqrt{3}U_N I_N \tag{4.16}$$

Rearranging this expression, the nominal current can be written as

$$I_N = \frac{S_N}{\sqrt{3}U_N} \tag{4.17}$$

Table 4.2 lists the variables needed to run the ATP supporting routine *BCTRAN*. IEXPOS is the normalized excitation current, so it is the current drawn by the transformer in the open circuit test at nominal voltage, and using Equation 4.17 it can be written as

$$\texttt{IEXPOS} = \frac{I_o}{I_N} \cdot 100\,\% = \frac{\sqrt{3}U_N I_o}{S_N} \cdot 100\,\% \tag{4.18}$$

ATP variable	Description	Unit
IEXPOS	Normalized excitation currentfor open circuit test	%
SPOS	Power base	kVA
LEXPOS	Normalized open circuit loss	kW
P12	Normalized short circuit loss	kW
ZPOS12	Normalized short circuit impedance	%

Table 4.2: *BCTRAN* input variables

LEXPOS is the active power at nominal voltage so it is defined as

$$\texttt{LEXPOS} = P_o \frac{U_N}{U_o} \tag{4.19}$$

P12 is the short circuit active power at nominal current.

$$\texttt{P12} = P_s \frac{I_N}{I_s} = \frac{P_s S_N}{\sqrt{3}U_N I_s} \tag{4.20}$$

ZPOS12 is the short circuit impedance normalized by the impedance at nominal conditions and is given as

$$\begin{aligned} \texttt{ZPOS12} &= \frac{Z_s}{Z_N} 100\,\% = \frac{U_s I_N}{I_s U_N} 100\,\% \\ &= \frac{U_s S_N}{\sqrt{3} I_s U_N^2} 100\,\% \end{aligned} \tag{4.21}$$

The open circuit test is normally performed with the generator and the measuring equipment on the low voltage side on the transformer, and the short circuit test with the generator and the measuring equipment on the high voltage side on the transformer. This means that U_N in Equation 4.18 and 4.19 is the rated voltage on the low voltage side, and in Equation 4.20 and 4.21 it is the rated voltage on the high voltage side.

Table 4.3 gives an example of test data for a 10/0.4 kV distribution transformer. The current of the open circuit test is not given directly, but

Variable	Description	Value	Unit
$Q_{m,o}$	Open circuit active power	0.648	kW
$P_{m,o}$	Open circuit reactive power	1.74	kVA
$P_{m,s}$	Short circuit active power	1.55	%
$S_{m,s}$	Short circuit apparent power	4.58	%

Table 4.3: Test data for an old 10/0.4 kV transformer. Note that the short circuit test data are relative to the power base.

the active and reactive power is given. This gives the following relation.

$$S_{m,o} = \sqrt{P_{m,o}^2 + Q_{m,o}^2} = \sqrt{3}U_N I_{m,o} \tag{4.22}$$

This is used in Equation 4.18 and `IEXPOS` is

$$\begin{aligned} \texttt{IEXPOS} &= \frac{\sqrt{P_{m,o}^2 + Q_{m,o}^2}}{S_N} \cdot 100\,\% \\ &= \frac{\sqrt{0.648\,\mathrm{kW} \cdot 1.74\,\mathrm{kVar}}}{400\,\mathrm{kVA}} \cdot 100\,\% = 0.464\,\% \end{aligned} \tag{4.23}$$

The active power is given directly so

$$\texttt{LEXPOS} = P_{m,o} = 0.648\,\mathrm{kW} \tag{4.24}$$

The active power of the short circuit test is given in percent of the power base so

$$\texttt{P12} = \frac{P_{m,s} S_N}{100\,\%} = \frac{400\,\mathrm{kVA} \cdot 1.55\,\%}{100\,\%} = 6.20\,\mathrm{kW} \tag{4.25}$$

The short circuit impedance `ZPOS12` in Equation 4.21 is equal to $S_{m,s}$ in Table 4.3. This can be seen by setting the short circuit current I_s equal to the nominal current I_N in the second equality Equation 4.21.

$$\texttt{ZPOS12} = \frac{U_s I_N}{I_N U_N} 100\,\% = \frac{S_s}{S_N} 100\,\% = S_{m,s} = 4.58\,\% \tag{4.26}$$

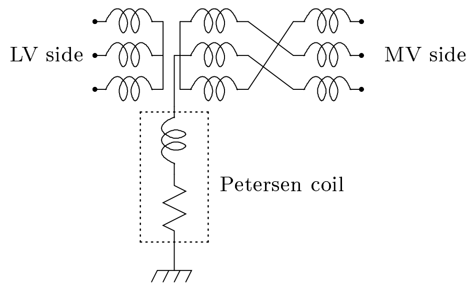


Figure 4.5: Petersen coil and Zy grounding transformer.

4.3 Petersen Coil

A normal MV network in Denmark is grounded through a Petersen coil. It is placed at the primary substation and it has the effect of suppressing the current in ground faults and is therefore often referred to as an *arc suppression coil*. The Petersen coil is connected to the star point on the MV network. As the MV network is delta coupled, a `Zy` grounding transformer is used to connect the Petersen coil to the network. The `Zy` transformer has two separate coils on each leg of the transformer [Lakervi and Holmes, 1989]. These two coils are connected to two adjacent phases as shown in the upper part of Figure 4.5. The Petersen coil is modeled by a series connection of a coil and a resistor as shown in the figure. The LV side of the grounding transformer is usually used as power supply for the transformer station.

The size of the coil is found as the value that gives a 50 Hz current through the coil of the same size as the current flowing through the network capacitance. This is the normal way of dimensioning the Petersen coil. If the total network capacitance is C_T then the value of the Petersen coil L_P is found as

$$L_P = \frac{1}{\omega^2 C_T}, \quad \omega = 2\pi 50\,\mathrm{Hz} \tag{4.27}$$

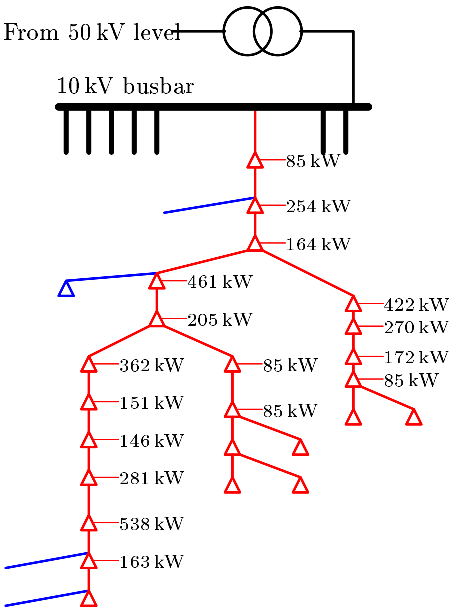


Figure 4.6: Model of A12 at *Glentegården*.

4.4 Network Model

Figure 4.6 shows a model of the A12 feeder at *Glentegården*. This model is used in Section 5 on page 47 to illustrate the ground fault localization algorithm, so an outline of the model will be given here. Details and ATP input files are given in Appendix C on page 113.

The model includes 42 individual cable segments and 8 different types of cable. The data on the cable network is taken directly from NESA's database. The total network is modeled using 212 Π-sections with ground resistance, each Π-section representing 40 m of cable. Furthermore, 20 loads and a Petersen coil is included in the model.

The triangle symbols in the figure represent loaded distribution transformers. The transformer itself is not included in the model, so the load is transferred to the MV network as a three phase symmetrical delta of impedances. Each impedance is a parallel connection of a resistor and an inductor.

The supplying 50 kV/10 kV transformer and the voltage source is modeled by a three phase star connected 10 kV source, a source impedance, and an ideal 1:1 transformer. The ideal transformer has the purpose of making the MV network floating, allowing the voltage on one phase to be zero during a ground fault.

The network model is generated by the makenet program described in Appendix E.1. Part of output from the program is a summary of the symmetrical components of the network model:

Terminal output

```
MakeNet version 3.5
Totals are:
  length of network      : 8480m.
  number of Pi sections  : 212
  symmetrical resistance : 2.22 ohm
  symmetrical inductance : 2.23 mH
  symmetrical capacitance: 2.36 uF
  zero resistance        : 11.4 ohm
  zero inductance        : 41.7 mH
  zero capacitance       : 1.05 uF
```

Using Equation 4.27 and the total zero sequence capacitance computed by makenet, the inductance of the Petersen coil becomes

$$L_P = \frac{1}{(100\pi)^2 \cdot 1.05\,\mu\mathrm{F}} = 9650\,\mathrm{mH} \qquad (4.28)$$

4.5 Impulse Response Generation

The purpose of this type of simulation is to get a representation of the network transfer function. As ATP is a time domain simulation tool so it seems to be most obvious to do the impulse response simulation in the time domain, although the ***FREQUENCY SCAN*** subroutine of ATP might be

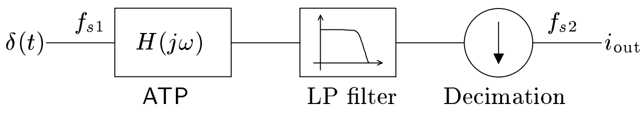


Figure 4.7: Signal processing scheme for impulse response generation

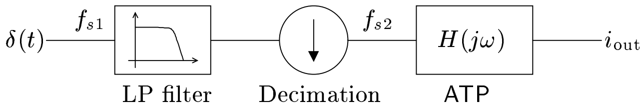


Figure 4.8: Signal processing scheme for impulse response generation

used as well. The advantage of this approach is that high performance signal processing tools like Matlab can be used for the frequency domain transformation.

The impulse source is modeled as a single phase DC current source which is zero at all times except at $t = 0$. As described in Section 4.7, the ATP step frequency chosen has to be 10–20 times larger than the bandwidth of the model for the ATP integration routines to converge. If the model bandwidth is larger than the required sampling frequency of the final signal, the ATP output must be low-pass filtered and decimated. This process is illustrated by the flow graph in Figure 4.7 where f_{s1} is the ATP step frequency and f_{s2} is the sampling frequency of the final signal.

If the network model contains only linear elements, the ATP simulation can be regarded as a linear system as indicated with $H(j\omega)$ in the ATP block in Figure 4.7. With this assumption the ATP block and the filtering/decimation block can be interchanged. This is illustrated in Figure 4.8 and the consequence of this scheme is that the source is the impulse response of a low-pass filter instead of a stepped DC source. The simulation now runs at the lower step frequency f_{s2}. If the bandwidth of the network model is large, this reduction in step frequency can reduce the time consumption for the simulation from days to hours. In practice

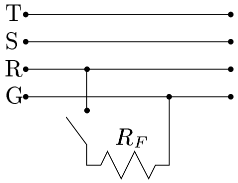


Figure 4.9: Ground fault model.

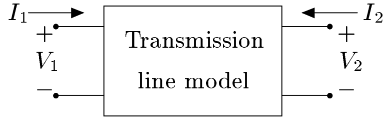


Figure 4.10: Transmission line two port.

the decimation in Figure 4.8 is omitted and the bandwidth of the low-pass filter must satisfy the ATP integration routines.

A Matlab function to generate the filter impulse response is described in Appendix D.6 and inclusion of the source in the ATP model is described in Appendix C.2.

4.6 Ground Fault Simulations

The model used for the ground fault simulations is a series connection of a switch and a resistor, R_F. This is connected between one phase and ground at the terminals of one of the Π-sections in the network model. This ground is not the global reference node `TERRA` but the ground node contained in the Π-section as described in Section 4.1.5. This is shown in Figure 4.9. The generator for the ground fault simulation is a three phase steady state sinusoidal source and the network model is described in Section 4.4.

4.7 Precision of ATP Simulations

This section addresses three key issues regarding the precision of the ATP simulations: the frequency range for which the model is valid, the bandwidth of the model, and the required minimum step frequency for the ATP integration routines. The primary part of the network model described here consists of cables so the first two issues deals with the Π-section. The last issue is general considerations for the ATP simulation.

4.7.1 Valid Frequency Range

To get an idea of the frequency range for which the cable model is valid, consider the two port in Figure 4.10. The admittance matrix $\mathbf{T}$ that gives the relation between the voltages and currents for this two port is given by

$$\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \mathbf{T} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \tag{4.29}$$

In Chapter 3 an admittance matrix for both a distributed parameter model and a lumped element Π-section is derived, repeated here for convenience. The admittance matrix for the distributed parameter model is given by

$$\mathbf{T}_d = \begin{bmatrix} \frac{1}{Z}\frac{\gamma L}{\sinh(\gamma L)} + \frac{Y}{2}\frac{\tanh(\frac{\gamma L}{2})}{\frac{\gamma L}{2}} & -\frac{1}{Z}\frac{\gamma L}{\sinh(\gamma L)} \\ -\frac{1}{Z}\frac{\gamma L}{\sinh(\gamma L)} & \frac{1}{Z}\frac{\gamma L}{\sinh(\gamma L)} + \frac{Y}{2}\frac{\tanh(\frac{\gamma L}{2})}{\frac{\gamma L}{2}} \end{bmatrix} \tag{4.30}$$

where Z is the series impedance, Y is the shunt admittance, γ is the propagation constant and L is the length of the cable that the model represents. In terms of the distributed cable parameters r, l, g, and c, the propagation constant is defined by

$$\gamma = \sqrt{(r + j\omega l)(g + j\omega c)} \tag{4.31}$$

The admittance matrix for the lumped element Π-section is given by

$$\mathbf{T}_\Pi = \begin{bmatrix} \frac{1}{Z} + \frac{Y}{2} & -\frac{1}{Z} \\ -\frac{1}{Z} & \frac{1}{Z} + \frac{Y}{2} \end{bmatrix} \tag{4.32}$$

It is obvious that these two matrices are equal if the hyperbolic sine and tangent terms in $\mathbf{T}_d$ are equal to 1. To get an estimate of the frequency

range for which this is true, let ε be some small number defined by

$$\varepsilon > \frac{\tanh(x)}{x} - 1 \qquad \text{and} \qquad \varepsilon > 1 - \frac{x}{\sinh(x)} \tag{4.33}$$

Using a Taylor expansion of the hyperbolic tangent, neglecting terms with powers of 5 and up and assuming that x is small, the first inequality of Equation 4.33 can be written as

$$\begin{aligned} \varepsilon > \frac{\tanh(x)}{x} - 1 &= \frac{x + \frac{x^3}{3!} + \frac{x^5}{5!} + \cdots}{x(1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \cdots)} - 1 \\ &\approx -\frac{2x^2}{6 + 3x^2} \end{aligned} \tag{4.34}$$

If the loss is neglected the propagation constant γ in Equation 4.31 becomes purely imaginary, and the square of γ real and negative.

$$\gamma^2 = -\omega^2 lc, \qquad \omega = 2\pi f \tag{4.35}$$

Rearranging Equation 4.34, substituting x with $\frac{\gamma L}{2}$ and γ^2 from 4.35, the frequency limit corresponding to the first inequality of Equation 4.33 can be approximated by

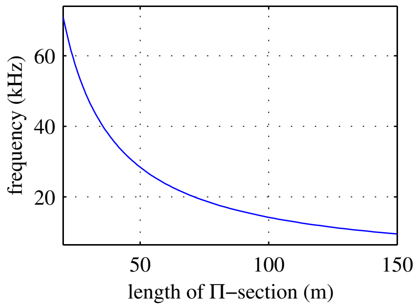
$$f < \frac{1}{\pi L}\sqrt{\frac{6\varepsilon}{(2 + 3\varepsilon)lc}} \approx \frac{1}{\pi L}\sqrt{\frac{3\varepsilon}{lc}}, \quad \varepsilon \ll 1 \tag{4.36}$$

Similarly the second inequality of Equation 4.33 can be approximated by

$$\varepsilon > 1 - \frac{x}{\sinh(x)} = 1 - \frac{x}{x + \frac{x^3}{3!} + \cdots} \approx \frac{x^2}{6 + x^2} \tag{4.37}$$

Again by rearranging Equation 4.37, substituting x with γL, and γ^2 from Equation 4.35, the frequency limit corresponding to the second inequality of Equation 4.33 can be approximated by

$$f < \frac{1}{2\pi L}\sqrt{\frac{6\varepsilon}{(1 - \varepsilon)lc}} \approx \frac{1}{2\pi L}\sqrt{\frac{6\varepsilon}{lc}}, \quad \varepsilon \ll 1 \tag{4.38}$$



Parameters for a 95 mm^2 Cu APB cable		
	ε	0.001
resistance	r	$0.32\,\mathrm{m\Omega/m}$
inductance	l	$0.26\,\mu\mathrm{H/m}$
capacitance	c	$0.29\,\mathrm{nF/m}$

Figure 4.11: Frequency range for a Π-section as a function of section length.

The overall frequency limit is the frequency range where both inequalities in Equation 4.36 and 4.38 are true, so the upper limit on the frequency range for which the Π-section is valid is given by Equation 4.38 as

$$f_{\mathrm{lim}} = \frac{1}{2\pi L}\sqrt{\frac{6\varepsilon}{lc}}, \quad \varepsilon \ll 1 \tag{4.39}$$

For a given cable and a given length of Π-section, Equation 4.39 will give the highest frequency for which the Π-section is valid. Figure 4.11 shows this limiting frequency as a function of the length of the Π-section, L, for a $95\,\mathrm{mm}^2$ Cu APB cable.

4.7.2 Model Bandwidth

If the model contains non-linear elements or broad spectered sources, the bandwidth of the model may have to be considered to guarantee valid simulation results in general. If the model contains Π-sections they will have a low-pass filtering effect, so an analysis of the Π-section can give a maximum limit on the model bandwidth. This is the subject of this section.

If a cable or an overhead line is modeled by a large number of Π-sections, a single section may be modeled as shown in Figure 4.12. It is

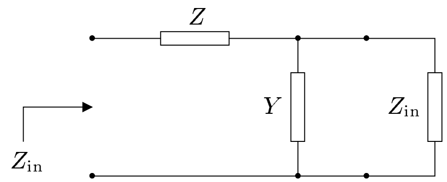


Figure 4.12: Model of Π-section

assumed that there is a large number of sections to the right of this model, so that the load impedance is the same as the input impedance Z_{in}, as shown in the figure. The transfer function of the model in Figure 4.12 is given by

$$H(j\omega) = \frac{Z_{\mathrm{in}}}{Z + Z_{\mathrm{in}}(1 + YZ)} \tag{4.40}$$

where Y and Z are defined as in Table 3.1 on page 17 and

$$Z_{\mathrm{in}} = \frac{Z}{2} + \sqrt{\frac{Z^2}{4} + \frac{Z}{Y}} \tag{4.41}$$

Equation 4.40 is a very sharp low-pass function with a cutoff frequency dependent on the four parameters: resistance r, inductance l, capacitance c and length of Π-section L. Figure 4.13 shows the cutoff frequency of Equation 4.40 for a $95\,\mathrm{mm}^2$ Cu APB cable as a function of L. Figure 4.13 can be used to ensure the validity of the ATP simulations in terms of step frequency for a specified cable type and length of Π-section.

All mathematical details and a treatment of the precision of the Π-section is given in Appendix B on page 109.

4.7.3 Step Frequency

As described in the Rulebook, ATP solves partial differential equations in the time domain by numeric integration using the trapezoidal method.

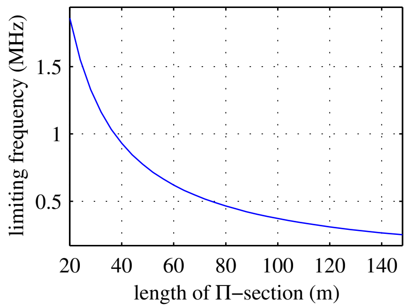


Figure 4.13: Cutoff frequency of Equation 4.40 for a 95 mm^2 Cu APB cable as a function of L.

The size of the time step, or equivalently the step frequency, is essential to the validity of the simulations since it controls the convergence of the integration. To illustrate this, the left side of Figure 4.14 shows two sinusoids. One is plotted with 8 steps per period and the other is the true sine function. When the 8 step function is integrated with the trapezoidal method the result is the area beneath the function and therefore the area between the functions can be regarded as an error measure. The error for the first half period is plotted relative to the true area in the left side of Figure 4.14 as a function the number of steps per period. From the figure it is seen that in order to get an error less than 1 % all voltages and currents on the ATP model must have at least 20 steps per period for the highest frequency. This means that the ATP step frequency must be 20 times larger than the bandwidth of the model for the integration routines to converge. For example, assume that a simulation must provide information about the system at a frequency range from 0 Hz to 5 kHz. First ensure that no signal on the model contains frequency components higher than 5 kHz. To avoid aliasing, the sampling frequency of the final signal must be 10 kHz. This is the Nyquist frequency. To get an error less than 1 % in the ATP simulations, the step frequency must be 20 times

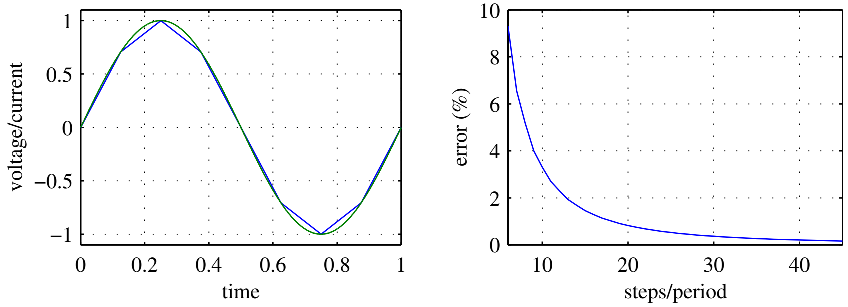


Figure 4.14: Error of the ATP integration routine

larger than the required signal bandwidth at 5 kHz, which is 100 kHz.

4.7.4 Summary

To summarize the discussion of precision, assume that we have a model employing a 95 mm^2 Cu APB cable and we need a signal bandwidth of 25 kHz. The series inductance is 0.26 μF and the shunt capacitance is 0.29 nF. Equation 4.39 can be used to find the largest acceptable length of the Π-section. With an ε of 10^{-3} corresponding to a signal to noise ratio (SNR) of 60 dB we find a length of 56.8 m as the maximal length of the Π-section. Rounding off to 50 m and using this as entry into Figure 4.7.2 we find that the bandwidth of the model is approximately 0.75 MHz. Using 20 steps per period at highest frequency gives an ATP step frequency of 15 MHz, which is equal to a time step of $6.7{\cdot}10^{-8}$ seconds. So, a cable model consisting of Π-sections each representing 50 m of cable, simulated in ATP using a time step size of 0.06 μs will give signals that are valid in a frequency range from 0–25 kHz.

Note that the step frequency derived in this section will result in good precision of the ATP simulation, even in the presence of non-linear elements. It may, however, be too pessimistic and less restrictive conditions may be present in the individual case.

Chapter 5

Ground Fault Localization

In this chapter the problem of localizing a ground fault in a radial compensated distribution network is addressed. The work described here is a major part of the contribution of this project.

5.1 Existing Methods

Very few reliable methods exist for localization of faults in branched, compensated, power distribution networks. The existing methods all suffer from drawbacks in this context as they assume conditions that are not present in a compensated branched network.

In [Zhu *et al.*, 1997] a method is described where a distance to the fault is computed by iterative solution of a set of equations. The basis is voltage and current measurements at the primary substation before and after the fault and a statistical model of the load. The load model is described by a set of parameters which depend on the power factor of the load and how the load reacts on voltage changes. The algorithm only gives a distance to the fault, so in branched networks this may result in several location estimates. To determine the true fault location, it is assumed that the fault trips the circuit breakers and causes an outage.

When the breakers are automatically reclosed, the tripping signals can be observed and the faulted section can be identified. Together with the fault distance, this gives an estimate of the fault location. An attempt to implement this algorithm is described in [Gunnarsson, 1998]. Here tests on simulated data show good results on single phase non-branched models. Three phase models did not give good results within the time frame of the project. Further more it is concluded that the method is highly dependent on the load model, which is very sensitive to its parameters.

In [Bo *et al.*, 1997] the traveling time of the fault generated transient is used to estimate the fault location. The fault initiates a surge moving in both directions on the cable section away from the fault location. When the surge reaches the termination of the cable it is reflected back and this continues until the attenuation of the cable has decreased the amplitude of the surge to zero. If the cable is non-branched and the length of the cable is known, the fault location can be determined by observing these surge reflection patterns. This method will obviously have problems in a branched network as multiple reflections will be very difficult, or maybe impossible, to track.

Extremely thorough and interesting work is presented in [Matti, 1992]. The primary focus is on the charge and discharge transients that arises when the ground fault occurs. The charge transient is caused by the voltage rise over the network capacitance on the two non-faulted phases, and the discharge transient is equivalently caused by the voltage collapse on the faulted phase. Three methods are developed and verified on experimental data. The localization accuracy obtained is about one kilometer and the fault resistance must not be larger than $50\,\Omega$ for the fault to be located reliably. The localization problem in a branched network is not discussed.

5.2 Impulse Response Model of Feeder

The three methods described in the previous section all use signal information with a limited frequency range. In [Zhu *et al.*, 1997] only the 50 Hz

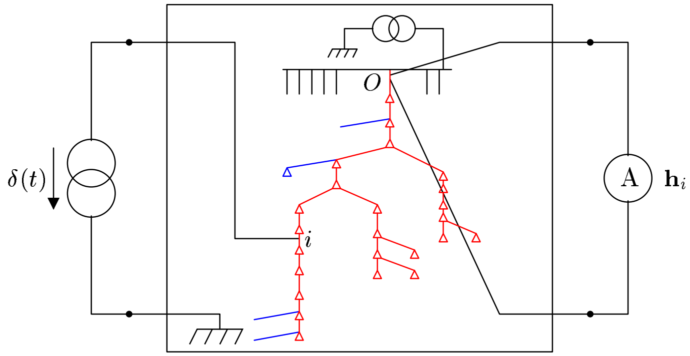


Figure 5.1: Impulse response generation.

component is used, in [Bo *et al.*, 1997] only frequencies in the MHz range are used as this is where information on reflection patterns is located, and in [Matti, 1992] the low frequency charge transient is used. The approach used in this project aims at using as much signal information as possible to solve the problem, and in the light of the complexity of the problem it is clear that all possible knowledge of the system has to be exploited.

Incorporation of network information usually means either making approximations to be able to develop analytical models, or using computationally potentially complex simulations tools. In this project these two approaches are combined so that a simulation tool is used to generate a set of impulse responses that describes the network and these impulse responses are then used in the signal processing algorithms to estimate the fault location. This principle was published in [Jensen *et al.*, 1998]. The advantage is that the impulse responses only need to be recomputed when the network topology is changed. This process can be automated and carried out off-line. The algorithms that operate on the impulse responses may be optimized for fast on-line estimations.

The basic idea is to represent the branched feeder by a set of impulse responses, covering the network in an equidistant spaced grid. Figure 5.1

illustrates how the impulse response is generated. A single phase impulse current source is connected between location i and ground, and the impulse response $\mathbf{h}_i$ is measured on the same phase at the observation point O by the substation busbar. This means that $\mathbf{h}_i$ is the impulse response of the network from the i'th grid point to the observation point. The impulse response $\mathbf{h}_i$ is a sampled version of the time signal $h_i(t)$. If T is the sampling time, the M length impulse response vector is defined by

$$\mathbf{h}_i = [h_i(0)\, h_i(T) \ldots h_i(mT) \ldots h_i((M{-}1)T)]^T \tag{5.1}$$

If this impulse response is generated for all N grid points, the network may be represented by the matrix

$$\mathbf{H} = [\mathbf{h}_0\, \mathbf{h}_1 \ldots \mathbf{h}_i \ldots \mathbf{h}_{N-1}] \tag{5.2}$$

where $\mathbf{h}_i$ is the i$^{\text{th}}$ column in an M by N matrix. The length of an impulse response for the A12 feeder described in Section 4.4 is a few milli-seconds, so if 3 ms is allocated at a sampling frequency of 100 kHz this involves 300 samples. The total length of A12 is approximately 8 km, and the network is represented by an impulse response matrix with 40 m between the grid points, so in that case $\mathbf{H}$ in Equation 5.2 is an 300 by 200 matrix. In double precision this is less than 0.5 MB. This amount of data is easily contained in the memory of a standard PC, so this may provide the basis for very fast optimized algorithms to solve on-line estimations.

5.3 The Deconvolution Approach

The measurements that we operate on, are performed at the observation point O by the primary substation busbar. It is assumed here that the signal represents the current of the faulted phase. In principle this signal is of infinite duration, so to get a causal signal for processing this current signal is high-pass filtered. This high-pass filter will remove the fundamental power frequency, and the loss in the network will ensure that transient will die out. It is assumed that it contains the transient caused by the ground fault and it will be called $\mathbf{y}_F$.

If $\mathbf{x}_F$ is the transient caused by the ground fault in grid point F, and assuming that we have the true impulse response matrix, the measured transient at the observation point is given by the convolution

$$\mathbf{y}_F = \mathbf{x}_F * \mathbf{h}_F \tag{5.3}$$

where F is the unknown index that we want to find, and $\mathbf{y}_F$ is the transient measured at the observation point. Both $\mathbf{x}_F$ and $\mathbf{y}_F$ are M length column vectors defined analogous to $\mathbf{h}_i$ in Equation 5.1. An estimate of the transient at the fault point can therefore be found by deconvolution as

$$\hat{\mathbf{x}}_{F,i} = \mathbf{y}_F * \mathbf{h}_i^{-1}, \qquad i = 0, \ldots, N{-}1 \tag{5.4}$$

where $\mathbf{h}_i^{-1}$ is the inverse impulse response defined by

$$\mathbf{h}_i * \mathbf{h}_i^{-1} = \boldsymbol{\delta} \tag{5.5}$$

The impulse vector $\boldsymbol{\delta}$ has only one nonzero value which is unity.

If Equation 5.3 is substituted into Equation 5.4, the estimate of the transient at the fault location can be written as

$$\hat{\mathbf{x}}_{F,i} = \mathbf{x}_F * \mathbf{h}_F * \mathbf{h}_i^{-1}, \qquad i = 0, \ldots, N{-}1 \tag{5.6}$$

From this expression it is seen that the estimate of the ground fault transient equals the true ground fault transient convolved by an mismatch function,

$$\mathbf{e}_{F,i} = \mathbf{h}_F * \mathbf{h}_i^{-1}, \qquad i = 0, \ldots, N{-}1 \tag{5.7}$$

By the definition of the inverse impulse response in Equation 5.5, the mismatch function for $i = F$ equals the delta pulse.

$$\mathbf{e}_{F,F} = \boldsymbol{\delta} \tag{5.8}$$

The network that separates two impulse responses consists primarily of electrical elements that are distributed in nature, i.e. a cable or overhead line. The lumped elements connected to the network will be shunt

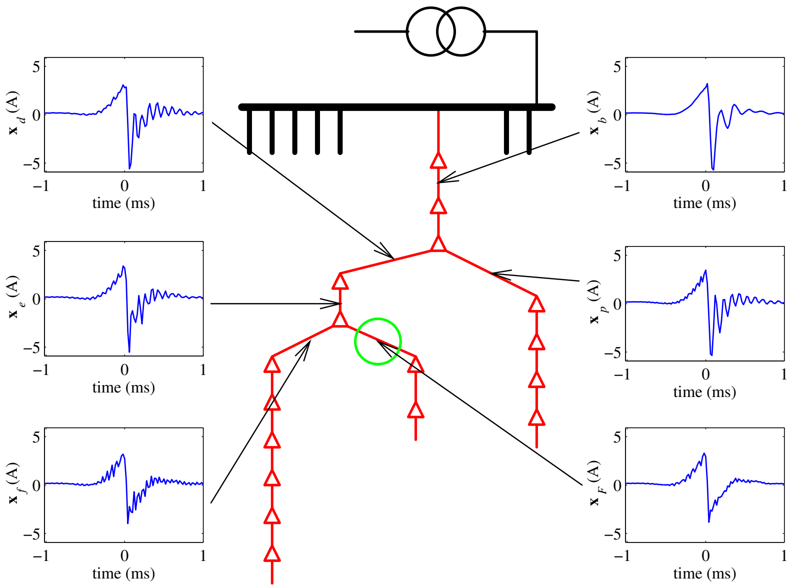


Figure 5.2: Deconvolution of ground fault.

elements such as a loaded distribution transformer. This means that the difference between two impulse responses will become zero as the physical distance between them goes to zero. If $d_{i,j}$ is the distance between impulse response $\mathbf{h}_i$ and $\mathbf{h}_j$, this can be expressed as

$$\lim_{d_{i,j}\to 0}(\mathbf{h}_i - \mathbf{h}_j) = \mathbf{0} \tag{5.9}$$

The N impulse responses $\mathbf{h}_i$ are therefore quantized instances of a function that is continuous along the physical network. This implies that when the locations corresponding to F and i are far apart, $\mathbf{e}_{F,i}$ will contain a large contribution of the network impulse response. As i is approaching F, $\mathbf{e}_{F,i}$ will gradually reduce to the delta pulse in Equation 5.8.

To describe this, Figure 5.2 shows six deconvolutions of the same ground fault transient $\mathbf{y}_F$. The ground fault is simulated by a switch and a resistor as shown in Section 4.6 on page 39. The deconvolutions are computed as $\hat{\mathbf{x}}_{F,i}$ in Equation 5.4 in the time domain by the iterative algorithm which is described in Appendix A.1 on page 99. The true ground fault location is marked in Figure 5.2 with a green circle.

These six deconvolutions all have a general shape of a high-pass filtered step function with some high frequency content. The high-pass filter is applied to the measurement $\mathbf{y}_F$ and the step function is due to the switch that is used to model the ground fault. The high frequency contents come from the mismatch function in Equation 5.7. If the amount of high frequency content is observed with regard to the distance to the true fault location, the general picture is that the larger this distance is to the true fault location, the more high frequency contents is contained in the deconvolved transient.

This means that if we can find some way of estimating this high frequency content, we may have an error measure that will point directly to the fault location.

5.4 Deconvolution in the Frequency Domain

The inverse impulse response $\mathbf{h}_i^{-1}$ in Equation 5.5 may not be physically realizable and the algorithm in Appendix A.1 on page 99 does not compute it explicitly. In fact a very large number of iterations is needed to compute the deconvolutions in Figure 5.2. It is therefore beneficial to transform the problem to the frequency domain.

If $h_i(t)$ is a continuous causal signal, the continuous Fourier transform is defined by

$$H_i(j\omega) = \int_{t=0}^{\infty} h_i(t)\mathrm{e}^{-j\omega t}dt \tag{5.10}$$

If $h_i(mT)$ is a sampled version of $h_i(t)$ with sampling frequency $f_s = \frac{1}{T}$ the discrete Fourier transform $H_i(k)$ is a sampled version of its continuous counterpart and is defined by [Ahmed and Natarajan, 1983]

$$H_i(k) = \sum_{m=0}^{M-1} h_i(mT)\mathrm{e}^{-j\frac{2\pi m k}{M}}, \quad k = 0, \ldots, M{-}1 \tag{5.11}$$

where $h_i(t)$ is assumed to be zero for $t < 0$ and $t > MT$. If M is even, $H_i(k)$ is the complex conjugate of $H_i(M{-}k)$ so all the information in $H_i(k)$ is contained in the signal for $0 \le k < \frac{M}{2}$.

The discrete Fourier transform will in the following be referred to as the spectrum, so $H_i(k)$ is the spectrum of $h_i(mT)$ or analogously to $\mathbf{h}_i$.

If $X_F(k)$ and $Y_F(k)$ are defined analogous to $H_i(k)$, the convolution in Equation 5.3 becomes a product and can be written as

$$Y_F(k) = X_F(k)\, H_F(k) \tag{5.12}$$

The spectrum of the fault transient in Equation 5.4 is written as

$$\hat{X}_{F,i}(k) = \frac{Y_F(k)}{H_i(k)} \tag{5.13}$$

and the spectrum of the mismatch function $\mathbf{e}_{F,i}$ in Equation 5.7 as

$$E_{F,i}(k) = \frac{H_F(k)}{H_i(k)} \tag{5.14}$$

By rearranging Equation 5.12 and insertion into Equation 5.14, the spectrum of the mismatch function can be rewritten as

$$E_{F,i}(k) = \frac{Y_F(k)}{H_i(k)\, X_F(k)} \tag{5.15}$$

The right hand side of this equation is the spectrum of the measured transient divided by the network impulse response and the transient at the fault location. Of these three signals, the latter is the only unknown factor, so if we have some knowledge about the transient at the fault location, Equation 5.15 could be used to compute the mismatch function.

Note that for index $i = F$ we get a perfect match, and Equation 5.15 becomes the spectrum of the unit delta pulse in Equation 5.8 which is

$$E_{F,F}(k) = 1, \quad \text{for all } k \tag{5.16}$$

5.5 Ground Fault Model

Assuming that it is valid to model the fault transient as a step function (i.e. as a switch), a model of the transient at the fault location $\mathbf{x}_F$ may be derived directly from the observed transient $\mathbf{y}_F$. This assumption will only have to be valid during the length of network impulse response. For a normal loaded network this is only a few milli-seconds for frequencies above 1 kHz.

The model can be derived by taking the minimum and the maximum value of the high-pass filtered transient at the observation point $\mathbf{y}_F$ and using the difference as the size of a step function. If this difference is called Δ_F, it can be expressed as

$$\Delta_F = \max_m(y_F(mT)) - \min_m(y_F(mT)), \quad m = 0, \ldots, M{-}1 \tag{5.17}$$

The time domain model of the ground fault transient at the fault location can then be written as

$$g_F(mT) = \left(\mathrm{u}((m - m_F)T) - \frac{1}{2}\right)\Delta_F, \quad m = 0, \ldots, M{-}1 \tag{5.18}$$

where $\mathrm{u}(t)$ is the unit step function, and m_F is a delay that aligns the step with the ground fault transient $\mathbf{y}_F$. Index F of g_F denotes that the model is derived from the measured transient $\mathbf{y}_F$.

Figure 5.3 shows a simulation of the ground fault transient $\mathbf{y}_F$ that was used for the deconvolutions in Figure 5.2. This transient has a range of approximately 7 A and it has a leading edge going upwards. The model in Equation 5.18 will therefore initially be 3.5 A and then step down to −3.5 A. This signal is then high-pass filtered to make it consistent with $\mathbf{y}_F$ and shown in Figure 5.4. Appendix D.4 gives a Matlab function which performs this task.

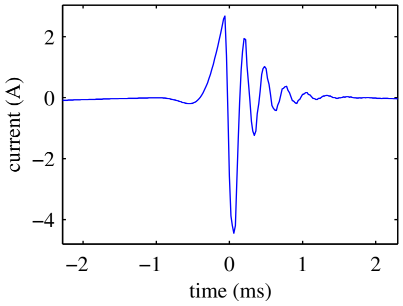


Figure 5.3: The high-pass filtered ground fault transient $\mathbf{y}_F$ at the observation point.

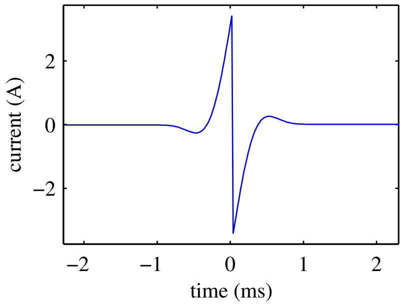


Figure 5.4: Model derived from the ground fault transient at the observation point.

5.6 Ground Fault Localization Estimation

Returning to the problem of estimating the high frequency contents in Figure 5.2, we now have the tools for deriving an error measure, which can be used to estimate the fault location.

If the spectrum of the ground fault model in Equation 5.18 is $G_F(k)$, and this is used as an estimate for $X_F(k)$, an estimate of mismatch spectrum in Equation 5.15 is

$$\hat{E}_{F,i}(k) = \frac{Y_F(k)}{H_i(k)\,G_F(k)} \qquad k = 0, \ldots, \frac{M}{2} - 1 \tag{5.19}$$

The interpretation of this in the time domain is that the estimates $\hat{\mathbf{x}}_{F,i}$ in Equation 5.4 is deconvolved with the ground fault transient $\mathbf{x}_F$, which then gives the network mismatch function $\mathbf{e}_{F,i}$.

Figure 5.5 shows the spectrum $G_F(k)$ of the ground fault transient model in Figure 5.4. Figure 5.6 shows the spectrum of four of the deconvolved transients from Figure 5.2, $\hat{X}_e$, $\hat{X}_f$, $\hat{X}_m$, and $\hat{X}_p$ (the hat is omitted in the legend of the figure). For comparison, G_F is shown together with the deconvolutions.

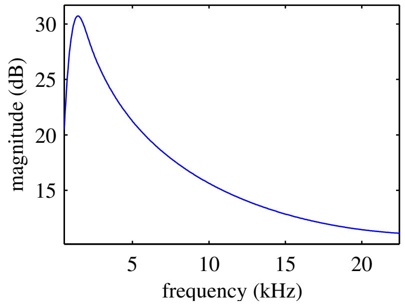


Figure 5.5: FFT of the ground fault simulation in Figure 5.3.

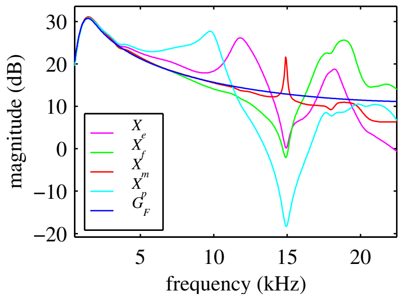


Figure 5.6: A comparison of the ground fault model and the deconvolutions in the frequency domain.

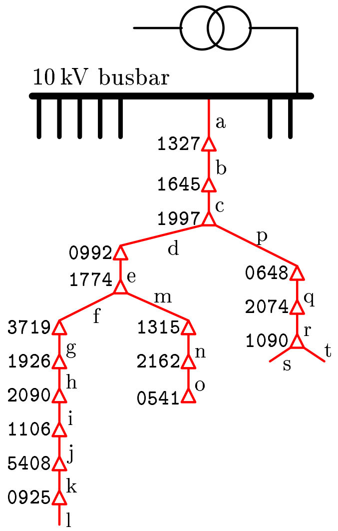
It is seen from the figure that the deconvolutions all differs from G_F except for $\hat{X}_m$, which exactly corresponds to the true fault location.

It is evident from Figure 5.6 that if we take the absolute value of the difference between the model and each of the deconvolutions and take some norm of this, we have a feature that will have a minimum near the true fault location. As the functions are plotted in decibels, this is exactly what is expressed in Equation 5.15. An error measure can therefore be written as

$$\mathcal{E}_{F,i} = \sum_{k=0}^{\frac{M}{2}-1} \left| 20 \log_{10}\left(\left| \hat{E}_{F,i}(k) \right| \right) \right| \tag{5.20}$$

Assuming that the transient model G_F is valid, the error estimate for index $i = F$ is 1, as in Equation 5.16, which gives an error measure $\mathcal{E}_{F,F}$ of zero. This means that the minimum value of the error measure in Equation 5.20 is an estimate of the fault location F.

$$\hat{F} = \left\{ i | \min_i \left(\mathcal{E}_{F,i} \right) \right\} \tag{5.21}$$



Transf.	Load 1		Load 2	
	P	*pf*	*P*	*pf*
1327	85.0	0.92	88.6	1.0
1645	254.0	0.9	311.7	0.87
1997	164.0	0.95	200.2	0.94
0992	461.0	0.89	542.4	0.91
1774	205.0	0.91	220.7	0.89
3719	362.0	0.9	484.3	0.88
1926	151.0	0.98	203.8	0.96
2090	146.0	0.95	199.1	0.92
1106	281.0	0.97	348.2	0.95
5408	538.0	0.87	687.3	0.93
0925	163.0	0.94	172.9	0.92
1315	528.0	0.92	647.0	0.96
2162	241.0	0.9	290.6	0.98
0541	164.0	0.93	237.2	1.0
0648	422.0	0.89	479.6	0.88
2074	270.0	0.96	304.4	0.89
1090	172.0	0.93	246.4	1.0

Figure 5.7: Simulation model for evaluation of ground fault localization algorithm. The transformer loads are given in the table with power P in kW and power factor $pf = \cos(\varphi)$. Two different load conditions are used, Load 1 and Load 2.

The location estimate in Equation 5.21 is a very simple way of utilizing the error measure $\mathcal{E}_{F,i}$ which might not be sufficiently precise in the presence of noise and model mismatch. It will, however, be used for the localization experiments in the rest of this chapter.

5.7 Simulation Results

To evaluate the localization method derived in this chapter, ground fault and impulse response simulations have been computed. The model used for these simulations is described in Section 4.4 on page 36 and an overview of the model is repeated in Figure 5.7 for convenience. Note that the figure

does not reflect the length of the cable sections between the distribution transformers. It only shows the interconnection between the transformers and the cables.

5.7.1 Two Different Load Conditions

The exact load condition on a network is not known in practice, so the localization method should not be too sensitive to changes in the load. Therefore two different load conditions called Load 1 and Load 2 are used for the ground fault simulations but only Load 1 is used for the impulse response model. This has the purpose of investigating the influence of a change in the load (a load mismatch) from the impulse response model to the ground fault simulations.

A table with the power in kW and power factor as $\cos(\varphi)$ are given for each distribution transformer in the model. The transformer numbers in the Transf. column correspond to the numbers in the figure. The power values in column Load 1 are taken from [Munk, 1995] as a normal daytime load condition. The power factor is taken from a normal distribution with mean 0.95 and a standard deviation of 5% (values larger than 1 are truncated). Load 2 is derived from Load 1 by multiplying the Load 1 power by a normal distribution of mean value 1 and standard deviation 10 %. The power factor is found the same way as for Load 1.

5.7.2 Impulse Response Model

The impulse response model is computed with the Load 1 condition. The distance between grid points in the impulse response grid is 40 m, which results in a number of grid points $N = 213$.

5.7.3 Ground Faults on the Load 1 Condition

Ground faults have been simulated on the Load 1 condition for all N grid points on the simulation model as described in Section 4.6 on page 39.

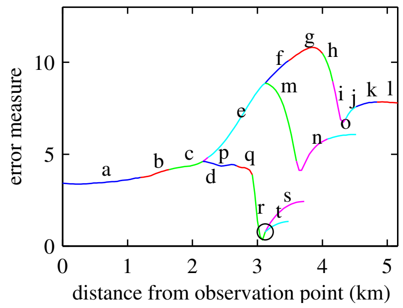


Figure 5.8: Error measure for section r with the Load 1 condition for ground fault simulations.

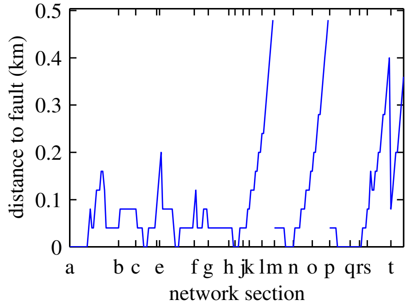


Figure 5.9: Distance from estimated to true fault location with the Load 1 condition for ground fault simulations.

The error measure $\mathcal{E}_{F,i}$ given in Equation 5.20 has been computed for all N grid points of the network. Figure 5.8 shows the error measure for a ground fault at the end of cable section r (see Figure 5.7) for the Load 1 condition. The error measure is plotted as a function of the distance to the observation point O by the busbar and *not* index i in $\mathcal{E}_{F,i}$. For each new cable section the color is changed to better distinguish the sections and identify them in Figure 5.7. The true fault location at the end of section r is marked with a circle.

The structure of the network is clearly recognized in the error measure as it seems continuous around branch points, e.g. at the point where section e, f, and m are joined together. This supports the assumption that the impulse responses, and thereby the error measure, is continuous along the network as expressed in Equation 5.9.

The estimated fault location as given by Equation 5.21 is simply the minimum of the error measure $\mathcal{E}_{F,i}$ with respect to index i. This is seen to give a very good estimate of the fault location. Notice that the error measure is very close to zero at the fault location.

Figure 5.9 shows the distance between the true and the estimated

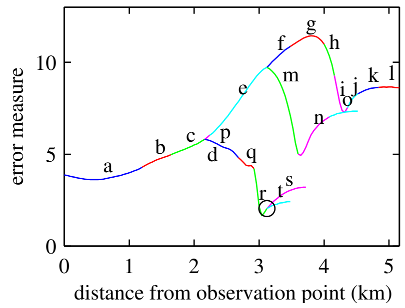


Figure 5.10: Error measure for section r with the Load 2 condition for ground fault simulations.

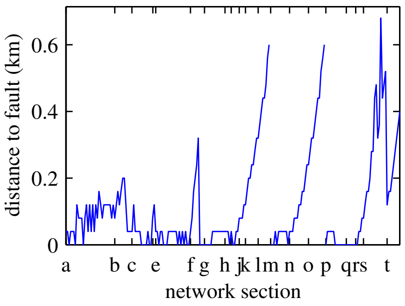


Figure 5.11: Distance from estimated to true fault location with the Load 2 condition for ground fault simulations.

fault location for all N ground fault simulations. Each point on the curve if found by minimizing a function similar to Figure 5.8 for the relevant fault location. The distance is computed along the network and *not* as the difference in the distance to the observation point. The horizontal axis corresponds to the different ground fault locations, so that the cable section starts at the section name and continues to the next section name. All sections are plotted after each other in alphabetic order, so no network structure can be seen from the figure. The mean value of the distance to the true fault location in Figure 5.9 over all grid points is 92 m and the standard deviation is 106 m.

5.7.4 Ground Faults on the Load 2 Condition

In this section the ground faults have been simulated on the Load 2 condition.

Figure 5.10 and 5.11 shows the same functions as Figure 5.8 and 5.9, only the load condition is changed to Load 2 for the ground fault simulations. The load condition for the impulse response model is still Load 1.

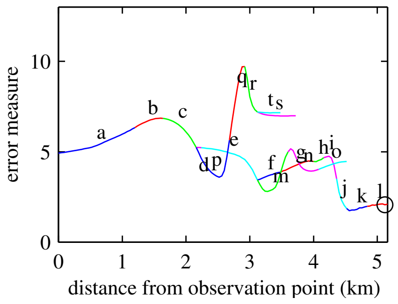


Figure 5.12: Error measure for section l.

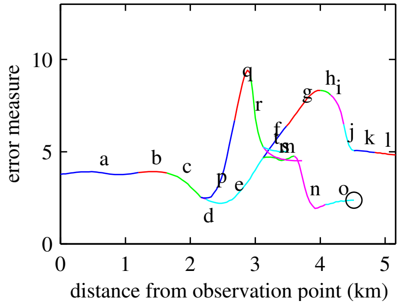


Figure 5.13: Error measure for section o.

This would be the realistic situation where to load would be known as some long term mean values.

The error measure in Figure 5.10 still points to the right section, but the minimum is not close to zero as in Figure 5.8, rather it is approximately 2. This means that the mismatch function in Equation 5.16 is not the ideal constant value of one, but it is still sufficiently small for this algorithm to detect the fault location.

Figure 5.11 shows the distance between the true and the estimated fault location for all grid points and it has a mean value of 122 m and a standard deviation of 148 m. This means that the estimated fault location will typically not be more than approximately 300 m away from the true fault.

In terms of evaluating the theoretical performance of the algorithm, this estimation accuracy should be compared to the total length of the network which is approximately 8 km. For a practical application of the algorithm, the distance between the transformers are an important factor when evaluating the estimation accuracy.

Both Figure 5.9 and Figure 5.11 have large errors at the end of each branch at section l, o, s, and t. To find out what causes these errors, consider Figure 5.12 and Figure 5.13. These two figures show the error

measure for section l and section o respectively. These two figures are the basis for the two large values at section l and o in Figure 5.11.

From Figure 5.12 and Figure 5.13 it is seen that the cause of the error is a flat minimum in the error measure towards the end of the branch. The stair case curve in Figure 5.11 is therefore caused by the situation that minimum in the error measure stays at the same location while the true fault location moves towards the end of the branch. This is also the case with section s and section t.

In addition to this, Figure 5.13 is close to giving a misleading result, as another local minimum at section e is close to be the global minimum. This would produce a significant error in the localization estimate when the estimate is based on the simple minimization of the error measure. This gives rise to the thought that a more information in the error measure could be utilized than expressed in the localization estimate in Equation 5.21.

The preceding discussion shows that the location estimate alone may provide a misleading picture of the performance of the localization algorithm. A visualization of all N error measures might therefore be useful. An example of this is shown in Figure 5.14. This figure is a vertical stacking of the N error measures. Here the vertical axis corresponds to the horizontal axis in Figure 5.13, but unlike Figure 5.13 the unit is not distance to the observation point, but simply the cable section in alphabetic order. The section name marks the end closest to the observation point. The level of the error measure is used as index into the color map shown in the figure.

Each point on the horizontal axis corresponds to a ground fault simulation and the axis index is the cable sections similar to the vertical axis.

The network branch points on the horizontal axis is indicated by arrows at the top of the figure. That is, the column just to the right of solid line at section m is close in physical location to the dotted line at the top of section e where the arrow points. This branch point is the point were section d, e, and m meets in Figure 5.7 on page 58.

The continuity of the error measure as given by Equation 5.9 can be

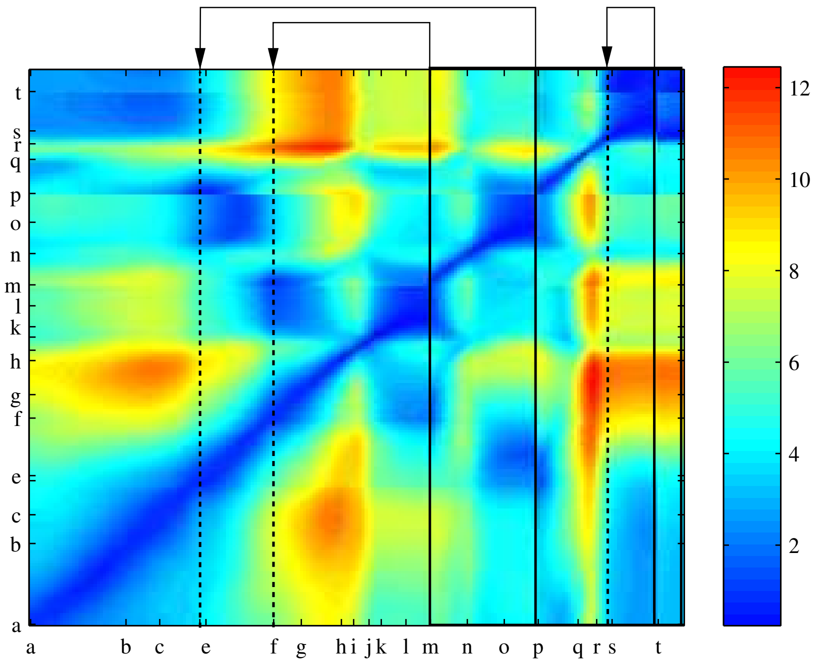


Figure 5.14: Visualization of the error measure $\mathcal{E}_{F,i}$ for ground faults simulated at all grid points. Impulse response model simulated with the Load 1 condition and ground faults simulated at the Load 2 condition.

seen by these branch points both at the vertical level (indicated by the arrows) and at the horizontal level. At the horizontal level this means that the error measure is continuous when the error is moved from one location to another, and at the vertical level it means that the error measure is a continuous function along the network for a given fault location.

The ideal situation is to have clear minimum at one single location for each ground fault, and for this single location to be the true fault location. This means that the ideal image of the error measure in this visualization

Impulse response		Ground fault		Mean
R_F (Ω)	Load	R_F (Ω)	Load	error (m)
1	1	1	2	57.8
1	1	100	2	1013.7
1	1	10000	2	1253.0
100	1	1	2	1319.6
100	1	100	2	99.5
100	1	10000	2	121.5
10000	1	1	2	1462.3
10000	1	100	2	99.0
10000	1	10000	2	94.8

Table 5.1: Localization results on simulation models with different fault resistance and load conditions.

is a dark blue diagonal line from the lower left corner to the upper right and red every where else.

5.8 Fault Resistance

During the work on simulating the experimental data which will be described in Section 6.6.4 it was found that the impulse response model in Section 4.5 on page 37 lacks a ground fault resistance. This resistance R_F should be inserted in parallel with the impulse current source. A number of impulse response models and ground faults were therefore computed for different fault resistances.

When a real ground fault is considered this resistance is unknown. It is therefore necessary to investigate the influence of this resistance. The two figures of the load change in Figure 5.12 and 5.13 were actually computed with two different fault resistances — 100 Ω for the impulse response model and 10 kΩ for the ground fault simulation.

The results on a more complete investigation of the influence of the ground fault resistance is given in Table 5.8. Load 1 is used for the impulse responses and Load 2 for the ground fault simulation. Three different values are used for the ground fault R_F, 1 Ω, 100 Ω, and 10 kΩ. For each

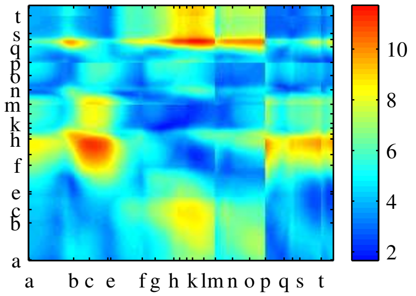


Figure 5.15: Error measure for impulse response: 100 Ω, ground fault: 1 Ω

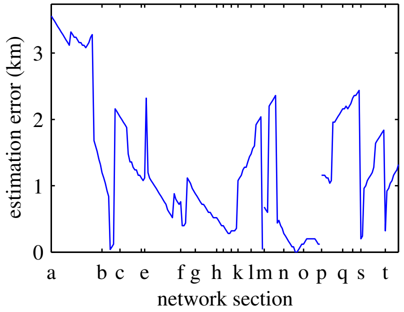


Figure 5.16: Estimation error for impulse response: 100 Ω, ground fault: 1 Ω

combination of these three resistances the localization algorithm is run on ground faults simulated at all nodes in the model and the estimation error is computed. The last column in Table 5.8 is the mean value of this estimation error for all the ground faults in the network. It is seen from the table that when the fault resistance for the impulse response is low the mean error is large for both a medium and a high ground fault resistance. When R_F is 100 Ω for the impulse response model, the mean error is approximately 100 m for the medium and the high fault resistance.

Figure 5.15 and 5.16 shows the error measure and the estimation error respectively computed for a impulse response fault resistance of 100 Ω and a ground fault of 1 Ω (the fourth row of the table). It is seen that the algorithm gives bad estimates at at the ends of the branches in the model and somewhat better estimates at the central parts.

Figure 5.17 to 5.20 shows the results for the fifth and the sixth row of the table. It is seen that the results are far better as the error measure in Figure 5.17 and 5.19 has the characteristic low value diagonal from the lower left to the upper right corner. From the estimation error in Figure 5.18 and 5.20 it is seen that the mean value is caused by a few high values as discussed in connection with Figure 5.12 and 5.13.

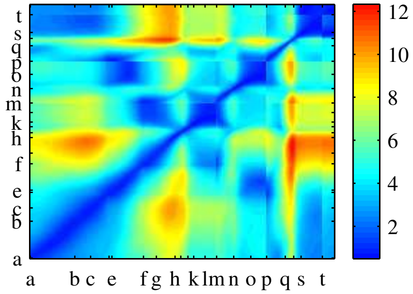


Figure 5.17: Error measure for impulse response: 100 Ω, ground fault: 100 Ω

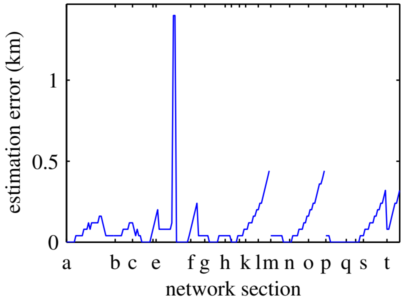


Figure 5.18: Estimation error for impulse response: 100 Ω, ground fault: 100 Ω

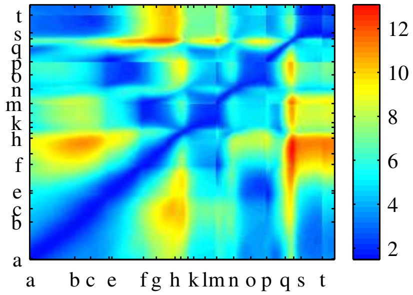


Figure 5.19: Error measure for impulse response: 100 Ω, ground fault: 10 kΩ

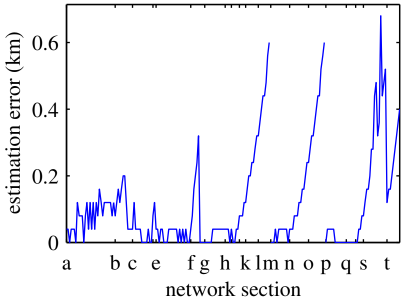


Figure 5.20: Estimation error for impulse response: 100 Ω, ground fault: 10 $k\Omega$

Chapter 6

Full Scale Ground Fault Experiment

This chapter describes the ground fault experiments that were carried out during the autumn of 1998. The experiments were part of a cooperation between NESA and ABB which again is a part of the DISMO project. The main purpose of the experiments is to establish a fundamental knowledge of the nature of a ground fault in a radial compensated medium voltage (MV) distribution network.

Note that the term *ground fault* is used in a broad sense, as it covers both the real case typically caused by worn-out cable insulation, as well as the connection made between one phase and ground during an experiment.

The facility used for the experiments is a 10 kV laboratory run by the Department of Development and Research in Power Distribution, DEFU. Originally the laboratory network was part of NESA's distribution network. During a restructuring, this part of the distribution network was taken out of service, and instead of discarding the network it was turned into this unique large scale laboratory. This makes the laboratory a very realistic environment for experiments as it consists of a wide variety of both old and new equipment just like a normal distribution network.

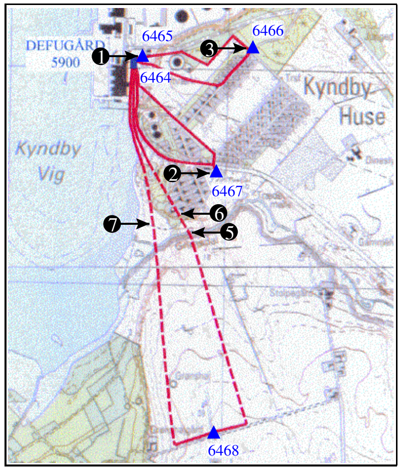


Figure 6.1: Geographical locations of the ground fault experiments at the DEFU 10 kV Laboratory.

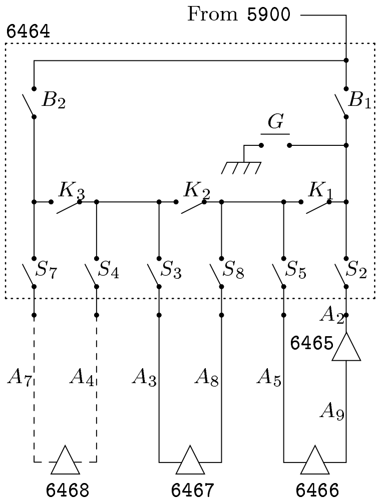


Figure 6.2: Topology of the laboratory network and overview of the breaker arrangement in 6464.

The MV network consists of approximately 4.5 km cable and 2.5 km overhead line. In addition to this, four 10 kV/0.4 kV distribution transformers and some LV network are included in the laboratory. The laboratory is connected to NESA's 50 kV network through a 50 kV/10 kV transformer.

6.1 Laboratory Network Overview

An overview of the network at the laboratory is shown in Figure 6.1. The network consists of three loops connected to the coupling station 6464. The coupling station is supplied from station 5900 which contains the main breakers of the laboratory. The network is grounded through a Petersen coil which is connected to the MV network via a Zy coupled grounding transformer, see Section 4.3 on page 35. This is the normal way of grounding the distribution network in Denmark. Both the Petersen coil and the grounding transformer are located in station 5900.

Four distribution transformers called 6465, 6466, 6467, and 6468 are located along the network as shown by the blue triangle symbols in Figure 6.1. The coupling station 6464 contains the breakers shown in Figure 6.2 and they allow the network to be reconfigured in a wide variety of topologies. The dashed lines are overhead lines and the solid lines are underground cables.

Each of the distribution transformer stations contains three breakers that connects the 10 kV/0.4 kV transformer and the two outgoing lines to the MV station busbar.

The only load that was available during the experiments was eleven 9 kW three phase fan heaters. Four of these was placed at the low voltage (LV) side of station 6467 and the other seven at station 6465.

6.2 Experimental Outline

It was planned to perform ground fault experiments at seven different locations, but one location, number 4, had to be omitted. The other six locations are marked in Figure 6.1 with numbers 1–3 and 5–7. Location 1–3 is the MV side of three of the distribution transformers and 5–7 is different locations at the overhead lines.

Two different network configurations were used for the ground fault experiments — a branched and a non-branched. These configurations are shown in Figure 6.3 and 6.4 and they will be referred to as configuration 1 and configuration 2. The two configurations can be switched between by operating only breakers K_2 and S_8 in Figure 6.2. At each physical location two sets of experiments was therefore performed — one set at each configuration.

In order to make the resulting data material as complete as possible, both the fault resistance and the closing angle was varied. The *fault*

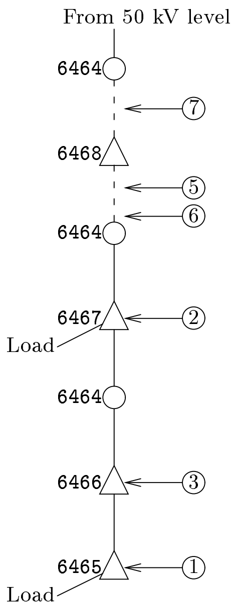


Figure 6.3: Network configuration 1.

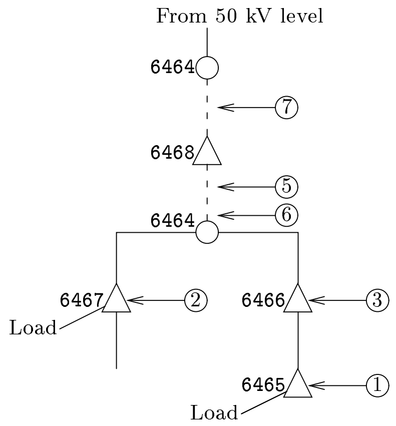


Figure 6.4: Network configuration 2.

resistance is the resistance connected between the MV phase and ground. The resistance was varied in steps from 0 Ω to 20 kΩ. The actual values of the resistances was determined by the values of the available resistors. The *closing angle* is the time instance of the ground fault connection relative to a zero crossing of the MV power supply, and it was varied in steps of 30° from 0° to 180°. One period of the 50 Hz sinusoid corresponds to 360°.

To summarize the outline of the complete experiment, two sets of experiments was performed at each of the six physical locations shown in Figure 6.1. One set on configuration 1 in Figure 6.3 and one set on configuration 2 in Figure 6.4. Each set of experiments covers a number of ground fault resistances and for each resistance a number of ground faults

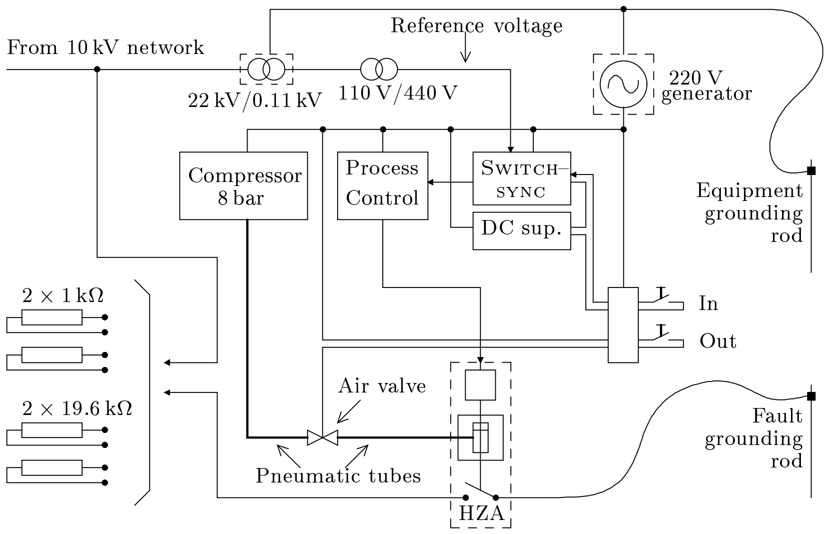


Figure 6.5: Mobile ground fault equipment.

were performed at different closing angles.

6.3 Ground Fault Equipment

As described in Section 1.4 on page 4, the type of ground faults that this project focuses on is caused by old insulation that eventually is unable to withstand the electrical field from the power supply. This will produce an arc through the insulation of the cable. In these experiments this situation is emulated by a breaker and a resistor. It may not be the best model of a real ground fault, but no information was available for improvments.

All the equipment is mounted on a trailer to make it mobile, and a gasoline generator acts as power supply to make the equipment independent of the stationary power supply.

location	1	2	3	5	6
resistance	$55\,\Omega$	$70\,\Omega$	$60\,\Omega$	$200\,\Omega$	$100\,\Omega$

Table 6.1: Resistance of the transition between the grounding rod and the surounding ground.

An overview of the ground fault equipment is shown in Figure 6.5. The central elements are a high precision breaker, a synchronizing unit, and a process control. The high precision breaker is called HZA, it has a well defined closing delay, and is designed for 15 kV and 4.5 kA. It is normally used in a stationary laboratory setup at ABB Corporate Research. The synchronizing unit is called SWITCHSYNC and is able to syncronize a trigger signal to a 50 Hz reference voltage. The process control trips the HZA breaker with an adjustable delay after being triggered by the SWITCHSYNC.

The setup containes both a MV and a LV circuit. The MV circuit starts in the top left corner from the MV network, goes to the resistors and via the HZA breaker to the fault grounding rod. The LV circuit starts at the lower side of the 220 V generator and supplies the compressor, the air valve, the process control, the DC supply, the SWITCHSYNC, and the In/Out box with power. The LV circuit are all two phase connections. The only electrical connection between the MV and the LV circuit is the 22 kV/0.11 kV transformer which produces a reference voltage to the SWITCHSYNC.

The grounding rod shown in Figure 6.5 connects the ground fault equipment and ground. The transition between the rod and the ground has in general an impedance different from zero. Properties of the soil and humidity has influence on this impedance. During the experiments the resistance in this transition was measured for five of the six ground fault locations. The measurement was performed with equipment specially designed for this purpose and the result of these measurements are listed in Table 6.1.

When the In-button in Figure 6.5 is pressed, the voltage from the DC

supply (DC sup. in the figure) gives the SWITCHSYNC the go-signal. The SWITCHSYNC then waits for a zeros crossing on the reference voltage, adds a preset time delay, and gives a trigger signal to the process control unit. The process control waits another preset time delay before it gives the HZA the In-signal and closes the switch.

When the Out-button is pressed the air valve lets the air pressure from the compressor through to the HZA which then opens the switch.

The reason for using the process control and not letting the SWITCHSYNC give the HZA the trigger signal directly is that the output voltage level of the SWITCHSYNC trigger signal is not the correct voltage level for the HZA. In addition, the process control unit has an easy access to adjustment of the time delay compared to the SWITCHSYNC. This is an advantage when the closing angle is changed during the experiments.

The resistors are shown as $2 \times 19.6\,\mathrm{k}\Omega$ and $2 \times 1\,\mathrm{k}\Omega$. The $19.6\,\mathrm{k}\Omega$ resistors are in fact series connections of four $4.9\,\mathrm{k}\Omega$ elements.

The shielding of the gasoline generator and the reference voltage transformer are connected to an equipment grounding rod. The frame of the trailer is also connected to this grounding rod together with the shielding of all other electrical equipment. This serves as a protection for both personnel and equipment. To simplify the figure, these connections are not shown in Figure 6.5.

Figure 6.6 shows the trailer at location 6 by the overhead line, and Figure 6.7 shows the ground fault equipment in more details. The gasoline generator is seen at the corner in the front of the picture, and behind that the compressor. To the left of the generator the 22 kV/0.11 kV transformer for the reference voltage is seen. Behind this transformer is the element of eight yellow high voltage $4.9\,\mathrm{k}\Omega$ resistors and in the right side of the picture the two $1\,\mathrm{k}\Omega$ resistors are seen.

Figure 6.8 shows a rear view of the trailer with the ground fault equipment. In the front of the picture in the left side, the syncronization and timing equipment is seen, in the middle the compressor and in the right side the generator is seen. Behind the syncronization unit is the HZA breaker.

Figure 6.6: Trailer with ground fault equipment at the field by the overhead line.

Figure 6.7: Ground fault equipment.

Figure 6.8: Trailer with ground fault equipment seen from the rear.

6.4 Acquisition Equipment

The data acquisition is performed at two different points in the laboratory network — at the feeding point of the network and by the Petersen coil.

At the feeding point three different sets of sensor arrangements are used. Each set produces measurements of voltage and current for all

three phases.

The first set is a high bandwidth measurement with a duration of 0.1 s acquired using a Tra800 transient recorder from W+W Instruments AG. Voltage sensors are 1000:1 voltage probes type P6015A from Tektronix, and current sensors are flexible Rogowski coils type CWT15 from PEM. The CWT15 are wound twice around the cable to increase the sensitivity.

The second set is a medium bandwidth measurement with a duration of 20 s acquired using the DISMO-PC (see Section 2.2 on page 9). Voltage and current sensors are the ABB combi-sensors which uses a resistive voltage divider as voltage sensor and a Rogowski coil as current sensor.

The third set is a high bandwidth measurement with a duration of 0.1 s also acquired using a Tra800 transient recorder. Voltage and current sensors are the voltage and current transformers mounted in station 5900 at the laboratory.

The transient recorders use an internal representation of 12 bit which is effectively not more than 10–11 bit. This gives a dynamic range of 60–66 dB. The DISMO-PC uses a 16 bit representation.

Figure 6.9 shows the first two sets of sensors for three phase voltage and current measurements in station 6464. The three brown cylinders at the rear are the ABB combi-sensors. At the bottom of the picture, the three voltage probes are standing in their respective fuse boxes. Above the probes are the three yellow flexible Rogowski coils wound around the red MV cables (only two of them are visible). A piece of grey foam plast centers the coil around the core.

Figure 6.10 shows the Tra800 transient recorder in station 6464, and in the rear of the picture is the DISMO-PC.

The voltage over the Petersen coil is acquired with both a Tektronix probe and a voltage transformer mounted in the coil. The current through the Petersen coil is acquired with both a Rogowski coil from Pearson Electronics and a current transformer mounted in the coil. These four signals are acquired with a transient recorder from Bakker.

The DISMO-PC does not need to be triggered automatically since it is easy to start the acquisition manually and still capture the ground fault

Figure 6.9: Sensor arrangement at feeding point in station 6464.

Figure 6.10: The TRA800 and the DISMO-PC in station 6464.

transient during the 20 seconds of acquisition.

The transient recorders require a trigger signal. The voltage over the Petersen coil is used to trig the recording as it raises very quickly from a near zero value when the ground fault occurs. An output signal from the transient recorder channel that records the Petersen coil voltage is used as input for a trigger box. This box gives a trigger output when the input reaches a certain level. The trigger output is transmitted through three optical fibers to each of the three transient recorders — the BAKKER and the TRA800 in 5900 and the TRA800 in 6464. The trigger level has to be adjusted for the different fault resistances in order to prevent false trigger signals.

Figure 6.11: The trailer by an overhead line pole.

6.5 The Experiment Procedure

For each of the six locations in Figure 6.1 the following procedure is carried out.

- The trailer is transported to the appropriate location for the ground fault. Figure 6.11 shows the trailer by a pole at the overhead line. Here the two grounding rods, one for the equipment and one for the ground fault injection, are driven approximately one meter into the ground. The equipment grounding is placed next to the trailer and the ground fault rod is placed 10–15 m away for security reasons.

Figure 6.12: Connection to the overhead line.

Figure 6.13: Connection to a distribution transformer.

- The relevant resistance is connected on the trailer and the HZA breaker is ensured to be in the open position. The connection is made to the MV network after it has been properly grounded. This connection is either to the overhead line as shown by Figure 6.12 (the red cable coming up by the pole) or to the MV side of a distribution transformer as shown by Figure 6.13.

- The transient recorders are set ready to receive a trig signal and the DISMO-PC is started. As the acquisition equipment always is located at the feeding point of the network, a radio link is needed to synchronize actions between the personnel operating the acquisition equipment and the personnel operating the ground fault equipment. This radio link is used to give the personnel at the ground fault equipment the go-signal. The In-button on the trailer is then pressed and ground fault will be connected. This triggers the transient recorders and when the DISMO-PC has finished the acquisition after 20 seconds, the power is taken off the network using one of the breakers in station `6464`. This is reported back to the personnel at the ground fault equipment which then opens the HZA breaker, adjusts the closing angle, and reports back with a ready-signal for the next run.

6.6 Acquired Data

The above procedure was followed for five different fault resistances: $0\,\mathrm{k}\Omega$, $0.5\,\mathrm{k}\Omega$, $1\,\mathrm{k}\Omega$, $2\,\mathrm{k}\Omega$, and $20\,\mathrm{k}\Omega$, and the closing angle was varied in six different steps: 0^o, 30^o, 60^o, 90^o, 120^o, and 150^o. At location 1 the above range of resistances was expanded with a fault resistance of $10\,\mathrm{k}\Omega$. With six different locations and two different configurations, this gives a total of almost 400 ground faults. Each ground fault was recorded in 22 different signals as described in Section 6.4 — 16 signals on transient recorders and 6 signals on the DISMO-PC. The total number of acquired signals is therefore more than 8000.

The sampling frequency is 1 MHz for the transient recorder data and 20 kHz for the DISMO-PC data. After a low-pass filtering and decimation the sampling frequency for the transient recorder data has been reduced to 100 kHz and this is the sampling frequency for all data presented in this section.

A general property of all data is that the closing angle have very little effect on the spectrum of the data except for a small variation in the magnitude. A closing angle of 0^o is an exception because, as one might expect, the fault is clearly seen in the 50 Hz time signal but nothing is seen in the spectrum above 2 kHz. In other words — when the fault occurs at a zero crossing of the voltage, no transient containing high frequency components is generated even though the fault is clearly visible in the

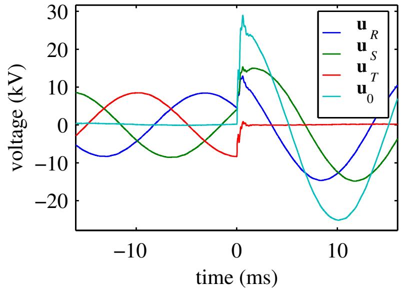


Figure 6.14: Voltage of faulted phase for location 1, configuration 1

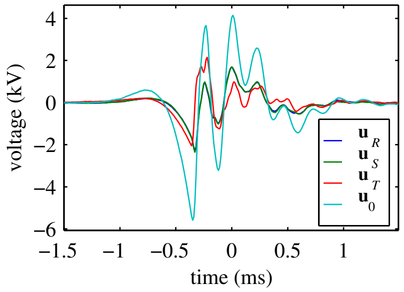


Figure 6.15: Transient of voltage of faulted phase for location 1, configuration 1

time domain by the collapsing voltage of the faulted phase.

All data presented in this section are therefore acquired at a closing angle of 90°.

6.6.1 Phase Measurements

Figure 6.14 shows the voltage of the three phases for a fault resistance of $0\,\Omega$ and a closing angle of 90° at location 1. As the figure shows, the fault resistance is connected to the T phase, which very rapidly falls to zero. At the same time the voltage increases on the other two phases. This is an effect of the Petersen coil which allows the center of the three phase voltage system to move away from zero. The zero system has been computed as the sum of the three phase voltages and is shown in the figure as $\mathbf{u}_0$.

Figure 6.15 shows the transients of the same voltage signals as above. It has been computed by high-pass filtering the voltage signals from Figure 6.14. The filter has a cutoff frequency of 2 kHz. The zero system transient has been computed as described above, after the filter has been applied. The signal from phase R and phase S are almost exactly the

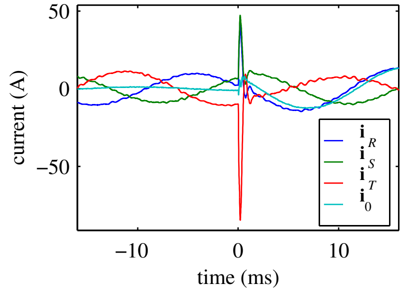


Figure 6.16: Current of faulted phase for location 1.

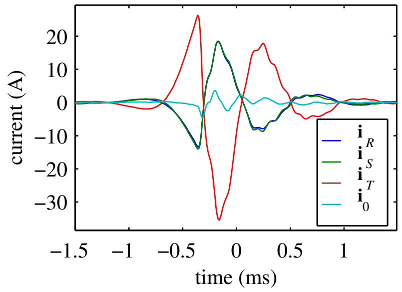


Figure 6.17: Transient of current of faulted phase for location 1.

same, so the phase S signal is covering the phase R signal in the figure. It is seen that the three transients has the same general shape and that they all add up in the zero signal.

Figure 6.16 shows the three phase current signals and the zero system from the same experiment as above. The faulted phase is again phase T, and as in Figure 6.15 the phase S signal is covering the phase R signal. Apart from the transient and a phase shift, the fault does not seem to have a large effect. The zero system current grow from a near zero value to be almost coinciding with phase R. Figure 6.17 shows the transients also computed using a high-pass filter with a cutoff frequency of 2 kHz. They show very clearly that the transients of the two non faulted phases are identical. Their sum is almost identical to the faulted phase T, but shifted 180°. This is seen from the zero system current which is much smaller than the phase transients.

The duration of the transients in both Figure 6.15 and 6.17 is approximately 2 ms for a signal bandwidth of 2–50 kHz.

Note that the transient $\mathbf{i}_T$ in 6.17 is the signal referred to as the measured transient $\mathbf{y}_F$ in Chapter 5. The spectrum of the voltage and current signals are computed by taking 10 ms of the transient signal from

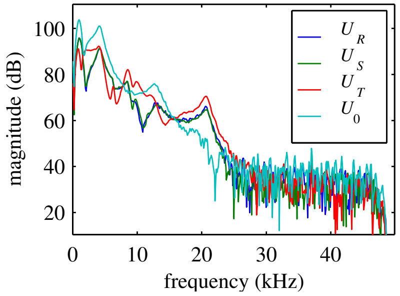


Figure 6.18: Magnitude spectrum of voltage transient for all three phases on location 1.

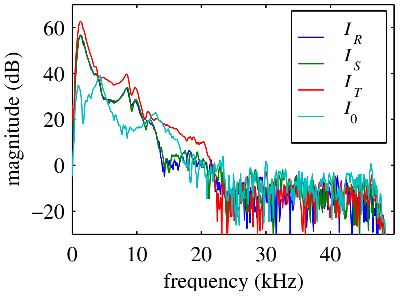


Figure 6.19: Magnitude spectrum of current transient for all three phases on location 1.

Figure 6.15 and 6.17, applying a Hanning window[1] and using an FFT. The first half of the resulting data is the spectrum of the transient from 0 to 50 kHz (half of the sampling frequency). The zero system is computed as the sum of the phase signals in the *time domain* and not as the sum of the magnitude spectras.

The result is shown in Figure 6.18 and 6.19. The absolute level of the vertical axis in decibel is relative to 1 V and 1 A respectively and is different in the two figures, but the range on the axis is kept the same.

The dynamic range seems to be larger for the voltage signals in Figure 6.18 than for the current signals in Figure 6.19 as they reach the noise level already at 15–20 kHz whereas the voltage signals are well defined up to 25 kHz. It may also be that the current signals does not have a frequency contents within the dynamic range for frequencies above 15 kHz. In any case the dynamic range complies very well with the predicted 60–66 dB (10–11 bit).

While Figure 6.15 shows that the transients does not cancel each other out, Figure 6.18 shows that this is only partly true. At 20 kHz all three

[1]A Hanning window or a raised cosine is defined by

$$h_n = \frac{1}{2}(1 - \cos(2\pi \frac{n+1}{N+1})), \quad n = 0, \ldots, N-1$$

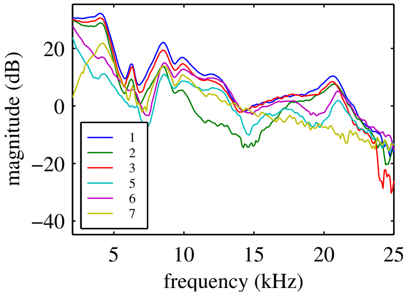


Figure 6.20: Magnitude spectrum of voltage transient for all locations on configuration 1

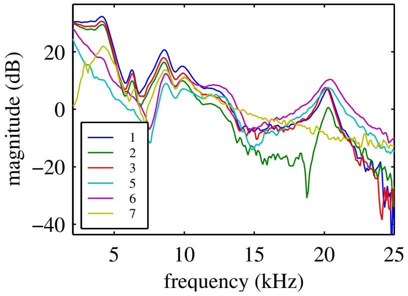


Figure 6.21: Magnitude spectrum of voltage transient for all locations on configuration 2

phases has a peak in the spectrum which is not found in the zero system.

6.6.2 Faulted Phase for All Locations

Figure 6.20 and 6.21 shows a comparison of the spectras for all six ground fault locations in the network. The spectras are computed from the voltage transient of the faulted phase. Figure 6.20 are measurements on configuration 1 and Figure 6.21 are measurements on configuration 2. Each location in the network is shown in Figure 6.3 and 6.4 on page 72.

In terms of the localization algorithm in Chapter 5, the ideal situation would be that all six spectras in each figure could be clearly distinguished from each other. The spectras does not show this ideal behavior, but with two exception, it is possible to tell the difference between the locations.

The spectrum for location 7 is very different from the other spectras.

In the frequency range from 10–20 kHz, location 1 and 2 are clearly distinguishable. For configuration 2, this means that the two branches can be distinguished. These two locations are the two loaded transformers. Location 5 and 6 are only 50 m apart and their spectras are also very much alike, so these two locations would be difficult to distinguish in this

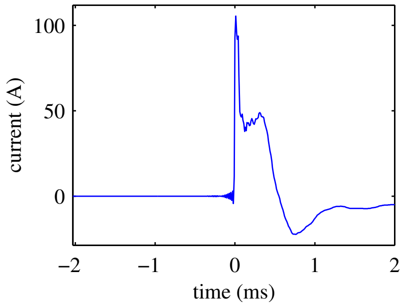


Figure 6.22: Ground fault current.

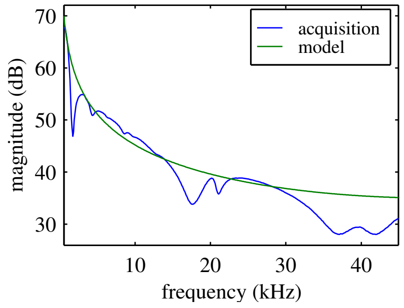


Figure 6.23: Spectrum of the ground fault transient.

frequency range. This is the first exception. These two locations can, however, be distinguished from the other locations in the network.

The spectrum for location 3 is very close to location 1. The two locations are on the same branch so this means that the localization algorithm may have difficulties in this part of the network. This is the second exception. This difficulty in distinguishing these two locations is in agreement with the analysis of the simulations in Section 5.7 (see the discussion in connection Figure 5.12 on page 62).

6.6.3 Ground Fault Current

Location 1 is actually located by the coupling station `6464` and station `5900` where the data acquisition equipment is placed, even though it is the last transformer station in the network. This comes from the loop structure of the network. This means that at this particular location it is possible to make an acquisition of the current in the fault itself. Figure 6.22 shows an acquisitions of this current. Six different acquisitions of this current was made at a closing angle of 90°, and they all look exactly the same. Figure 6.23 shows the spectrum of the ground fault current together with the spectrum of a model derived as described in

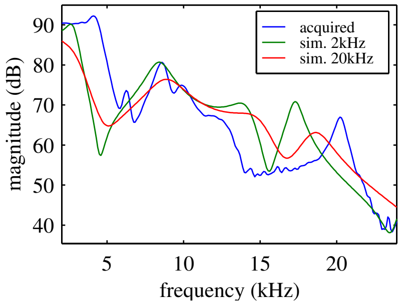


Figure 6.24: Voltage, location 1, configuration 2.

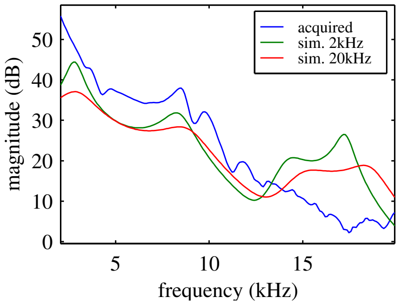


Figure 6.25: Current, location 1, configuration 2.

Section 5.5. The figure shows that the two spectras have the same general shape ($-20\,\mathrm{dB/decade}$), but the acquired ground fault signal has a few zeros which is not accounted for in the initial approximation described in Section 5.5.

6.6.4 Modeling the Experiment Data

This section describes the first steps in the process of validating the ground fault localization algorithm. It is assumed that a valid model of the network and a model of the current in the fault exist. If this requirement is met, it is possible to simulate the ground fault experiments as all parameters are known. This is the subject of this section.

Simulations has been computed for the network at the laboratory. The primary input file for ATP is discussed in Appendix C. The cable parameters for the model has been computed for 2 kHz and for 20 kHz. This gives two different simulations for each experiment signal.

This section gives a few representative examples of the comparison between the simulations and the acquired data. Appendix F.4 on page 185 includes figures of both configurations and all locations plotted together with their simulated version.

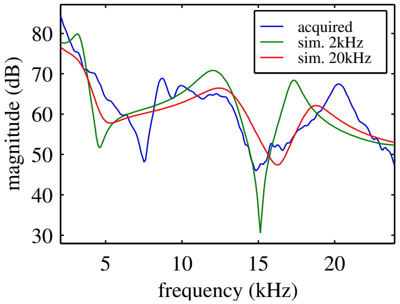


Figure 6.26: Voltage, location 5, configuration 2.

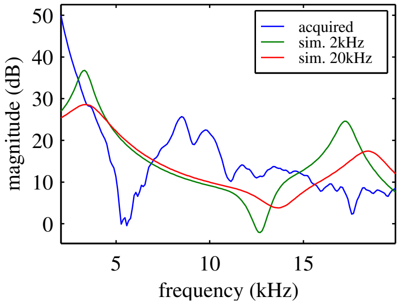


Figure 6.27: Current, location 5, configuration 2.

Figure 6.24 and 6.25 shows the voltage and the current spectras of the faulted phase at location 1, configuration 2, together with their two simulated versions. From Figure 6.24 it is seen that the simulations have the same general shape as the acquired data. Some of the peaks are shifted towards lower frequencies and the 20 kHz simulation seem to give the best general fit. A few small variations at 6 and 9 kHz in the acquired data are missing in both the simulations. The spectrum of the current in Figure 6.25 show the same general properties as the voltage below 13 kHz. Above 13 kHz a peak in the simulation is not found in the acquired signal. From the noisy look of the acquired signal this peak may be missing because it falls under the noise level. Location 3 shows the same characteristics as this location (1).

Figure 6.26 and 6.27 show the same signals and simulation, only it is from location 5 at the overhead line. The voltage in Figure 6.26 show the same general properties as above. The current in Figure 6.27 seems to be totally different, but it might be a shift in the spectrum as the 2 kHz simulation seems to have the same peaks at 13 kHz as the acquired signal has at 5 kHz. These shifts does not come from the processing of the data. They seem to be dependent on the grounding of the model. That is, the resistance in the ground branch of the Π-sections and the

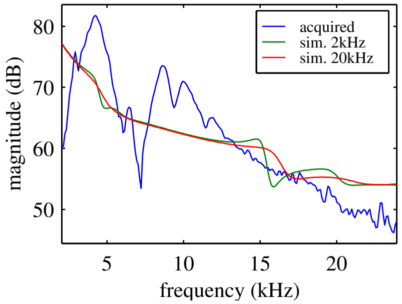


Figure 6.28: Voltage, location 7, configuration 2.

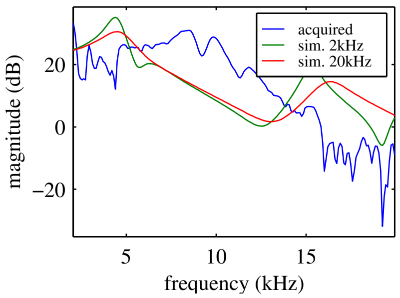


Figure 6.29: Current, location 7, configuration 2.

resistance between the ground of the Petersen coil and the ground of the power generators in the model.

Figure 6.28 and 6.29 show the same signals and simulations for location 7 — the location closest to the acquisition point. Here both the voltage and the current signals differ from the simulations. Whether this is caused by a bad model of the overhead line, or it is caused by a general problem with the model is not possible to determine within the time frame of the project. It may also be the same shifting problem as above. The frequency range of the data is too small to show if this is the problem.

In general it was found that the grounding of the model had a significant influence on on the fitting to the experimental data. In the model used to simulate the data in this section, the grounding of the Petersen coil and the star point of the power generators were keep separate. The Petersen coil was only connected to the ground branch of the Π-sections of the network. The only connection between the generator star point and the network was a 1 MΩ resistor connected to each of the loads. This resistor was only included for numerical stability of the simulation.

Chapter 7

Conclusion and Future Work

7.1 Conclusions

The main issue of this project is the design of a general monitoring system for a medium voltage power distribution network. The target network is a typical Danish distribution network, i.e. a radial network, compensated by a Petersen coil at the primary substation. On the basis of three phase voltage and current measurements, the monitoring system should be able to detect events such as faults, start and stop of decentralized power production, large changes in loads, etc.

A new representation of a distribution network is presented, where the network is modeled by a set of impulse responses referring to a number of equidistant locations along the network. This allows for using standard signal processing tools for estimation instead of simulation tools, some of which are computationally very demanding. This principle was published in a paper at the International Conference on Acoustics, Speech, and Signal Processing 1998 (ICASSP'98) in Seattle, USA [Jensen *et al.*, 1998].

Using this representation of a distribution network, a ground fault localization algorithm, a method of estimating the location of a ground fault in a branched, compensated, radial, distribution network, is proposed. The method uses only measurements of voltage and current at the primary substation, and network data which is generally available at the utility data base. It is assumed that a valid network model and a model of the current in the fault exists. The method is verified successfully on simulated data.

A full scale experiment on ground faults in medium voltage distribution networks was performed and a large amount data of was acquired. The ground faults were emulated by connecting a 10 kV phase to ground using a high precision controllable switch, and voltage and current signals were acquired at the feeding point of the network using measurement transformers, probes and Rogowski coils.

The experimental data sustains the potential of the ground fault localization algorithm. An actual localization test on the experimental data was not possible to perform within the time frame of the project. The results, however, indicate that this will be possible with improvement of the network models.

A specific feeder is chosen for investigations of the activity on a power distribution network during normal operation. A number of signals covering a 24 hour cycle are acquired and analyzed. Each of the signals have a duration of 45 minutes and a bandwidth of 10 kHz. Data was analyzed for transients and some initial classification of the detected transients were performed. Approximately 75 % of the transients are classified as a motor start.

All numerical simulations in this project are computed using ATP and the cable models are composed by a large number of Π-sections. A method for inclusion of a distributed ground resistance (impedance) in these Π-sections is proposed.

7.2 Suggestions for Future Work

This chapter aims at describing the open questions that remain, and to give a list of further ideas that the author has generated but not investigated during the project. It is intended by the author to give the best opinion of what might be done in relation to future work.

In terms of a general monitoring system much work still has to be done. At present, no general algorithm can detect and classify all events. Each type of event has to be dealt with separately. Detection and classification of the motor start event is treated in [Munk, 1995]. In this thesis the detection and localization of a ground fault event has been treated. These algorithms, however, are still in their development phase.

7.2.1 Localization Algorithm

The error measure, derived in Section 5.6, is used directly to give the estimate of the ground fault location as the location with the smallest error. This means that only one element of the error measure is used and therefore valuable information may be wasted. Below are a few ideas of how more information could be utilized.

- The actual level of the error measure minimum could be used as a certainty indicator. E.g. the lower this level is the more certain the algorithm is of the estimated ground fault location. This is a direct consequence of the construction of the localization algorithm in Section 5.6.

- Instead of just giving the global minimum as an estimate of the fault localization, the function could be traced for local minima with an error measure close to the global minimum. In this way two (or more) estimates instead of one could be the result, and the estimates could be prioritized with a certainty indicator as described above.

- The error measure could be used as input for a neural network. As the available data on real ground faults are very limited, the training

> data for the neural network will have to be based on simulated ground faults. The drawback to this approach is that assumptions have to be made of the ground fault properties such as the fault impedance. The training data and the input for the neural network may therefore have to be normalized in some way to improve the generalization properties of the neural network. For information on neural networks and the generalization property, see [Haykin, 1994]. The experiment data and the models described in Section 6.6.4 on page 87 may be used to investigate this possibility.

The results of modeling the experimental data in Section 6.6.4 indicate that the voltage signals of the simulation model gives a better fit to the experimental data. This suggests that the impulse response model for the localization algorithm should be based on voltage impulse responses instead of current impulse responses as described in Chapter 5. An even better solution might be to base the algorithm on *both* the voltages *and* the current impulse responses.

The estimation error, described in Section 5.6, is computed in the frequency domain. Other domains might provide a better basis for an error measure. The cepstrum (see [Oppenheim, 1989]) might give a higher degree of separability of the different ground fault transients on the network.

7.2.2 Network Element Models

The network models used in this project are all generated by ATP and are computed in the time domain. The advantage of this is that both the impulse responses and the ground fault simulations can be computed on the same model. The localization algorithm computes the error measure in the frequency domain so it may be considered to generate the impulse response model directly in the frequency domain, e.g. with a the ***FREQUENCY SCAN*** subroutine of ATP.

Cable Models

Analysis of experimental data shows that the network models need some improvement regarding the ground fault simulation. A number of suggestions of possible improvements are:

1. All cable models are computed using the ***CABLE CONSTANTS*** subroutine of ATP which assumes circular core cross section. The ***CABLE PARAMETERS*** subroutine allows for arbitrary cross sections and will therefore probably give a better model of the sectionalized cables.

2. All cables are modeled by Π-sections which assume constant distributed parameters. Two different approaches may improve the model in this respect:

   - The ***NODA SETUP*** subroutine based on ARMA models allows for a *frequency dependent* distributed parameter model to be used for *time domain* simulations. The steady state simulation (ground fault) may suffer from large initial transients caused by the ARMA model. This means that if the simulation is started at zero condition a large part of the simulation time is wasted before the simulation reaches steady state, and the ground fault can be connected. As a very large number of simulations are needed, this waste of simulation time may be so substantial that the model is unusable in practice. Referring to the impulse response simulation this is not a problem, as the initial condition in this case is zero.
   - EMTP has a frequency scan option called ***EXACT-PI*** which uses a separate set of ***CABLE CONSTANTS*** output computed at each of the frequencies in the scan. In this way the frequency dependence of the cable parameters are taken into account.

During this project it has not been possible to verify the inclusion of the distributed ground resistance in the Π-section as described in Chapter 4 on page 21. The problem with the ground fault experiment data in that respect is that the network at the test facility is composed of loops coming back to the ground point of the network. The physical distance of the two electrical remote ends of the network is therefore small and does not reflect the conditions on a normal radial feeder. The Π-section may be validated by experiments on the network coupled as six individual feeders.

Transformer Models

It was attempted to include a transformer model in the simulation using the *BCTRAN* subroutine. No real improvement was detected in terms of the mismatch problem between the experimental data and the simulations (see Section 6.6.4 on page 87). This may be caused by the fact that the *BCTRAN* transformer model is a low frequency model that does not take the stray capacities into account. Another possibility may be that the signals in question do not pass through the transformer, as both the fault and the acquisition are on the MV network. In all circumstances it cannot be excluded that a transformer model might be significant for future models wherefore it is treated in Section 4.2.

Ground Fault Experiments

A very large and interesting data material has been acquired during these experiments. Unfortunately it has not been possible to conduct an extensive analysis of this data during this project, but an effort has been made in preprocessing the data so it is readily available for analysis.

For research purposes in terms of the ground fault localization algorithm it would be desirable if the data acquisition equipment has a dynamic range larger than the 60–66 dB (10–11 bit) that was used for these experiments. As the data in Section 6.6.4 on page 87 shows, the bandwidth of the data above the noise level is not more than 25 kHz. A possible solution to this problem may be to apply an analog high-pass filter prior to the data acquisition. This way the large fundamental component can be suppressed and the full dynamic range of the data acquisition equipment be utilized. An obvious solution would be to use equipment with a higher precision.

The experimental data show in some cases an extensive noise component around 2 kHz. This may be caused by serial resonance circuit between the overhead line inductance and the cable network capacitance. In order to investigate this possibility further experiments are necessary.

7.2.3 Ground Fault Current

Knowledge about the current through a *real* ground fault is needed as well as the resistance in the fault as a function of time. This information should be used to verify both the experimental emulation and the simulation model by a switch and a resistor. It should also be investigated if a general model might be constructed, either in the time domain or in the frequency domain. Such a model might be used to improve the localization algorithm in Section 5.6 on page 56.

Appendix A

Signal Processing Algorithms

A.1 Iterative Deconvolution Algorithm

Assume that we have a vector, $\mathbf{y}$ that is defined by

$$\mathbf{y} = \mathbf{x} * \mathbf{h} \tag{A.1}$$

where vector $\mathbf{y}$ and $\mathbf{h}$ are known, and that we want to find $\mathbf{x}$ by deconvolution. This can not be done directly due to accumulated numerical errors and noise. An iterative approach, however, that minimizes an error vector can be used to get an estimate of $\mathbf{x}$. If we have an initial guess, $\hat{\mathbf{x}}_0$ an update of the estimate, $\hat{\mathbf{x}}_i$ can be found by applying the method of steepest decent [Haykin, 1996]

$$\hat{\mathbf{x}}_i = \hat{\mathbf{x}}_{i-1} - \lambda \frac{\partial ||\mathbf{y} - \mathbf{x}_{i-1} * \mathbf{h}||^2}{\partial \mathbf{x}_{i-1}} \tag{A.2}$$

where λ is called a step size parameter and determines the length of each step taken along the error surface gradient. Convergence is in general highly dependent on λ.

If we define an error vector as

$$\mathbf{e} = \mathbf{y} - \mathbf{x} * \mathbf{h} \tag{A.3}$$

we have that the squared error can be written as

$$\mathbf{e}^T\mathbf{e} = ||\mathbf{y} - \mathbf{x} * \mathbf{h}||^2 \tag{A.4}$$

The gradient to the squared error with respect to $\mathbf{x}$ can then be written as

$$\begin{aligned} \frac{\partial \mathbf{e}^T\mathbf{e}}{\partial \mathbf{x}} &= -2(\mathbf{h} * \mathbf{e}^{\mathrm{rev}})^{\mathrm{rev}} \\ &= -2(\mathbf{h} * (\mathbf{y} - \mathbf{x} * \mathbf{h})^{\mathrm{rev}})^{\mathrm{rev}} \end{aligned} \tag{A.5}$$

where the superscript rev represents a reversing of the elements of the vector. That is, if $\mathbf{x} = [x_1\, x_2\, x_3\, x_4]$ then $\mathbf{x}^{\mathrm{rev}} = [x_4\, x_3\, x_2\, x_1]$.

If Equation A.5 is inserted into Equation A.2 the iteration update is found as

$$\hat{\mathbf{x}}_i = \hat{\mathbf{x}}_{i-1} + 2\lambda(\mathbf{h} * (\mathbf{y} - \mathbf{x}_{i-1} * \mathbf{h})^{\mathrm{rev}})^{\mathrm{rev}} \tag{A.6}$$

All that is needed by Equation A.6 is an initial guess for $\mathbf{x}$, $\mathbf{x}_0$ and a step size λ. As initial guess, $\mathbf{y}$ may be used if no prior knowledge is available.

A.2 Design of Digital Integrator

The transfer function for the ideal integrator is $1/s$. With $s = j\omega$ this function has a magnitude of -20 dB per decade and a constant argument of -90^{o}. The magnitude of this function is infinite at DC which is undesirable in this application. The design must therefore include some kind of DC (mean value) extraction. Three different design approaches of the digital integrator is considered:

1. Integration of a periodic signal.
2. Low-pass filter.
3. IIR filter designed using bilinear transformation.

A.2.1 Integration of a periodic signal.

An approximation to an integration of a periodic signal $x(n)$ can be calculated as the mean value of a number of previous samples of $x(n)$ as shown in Equation A.7.

$$\begin{aligned} y(n) &= \frac{1}{L}\sum_{k=0}^{L-1} x(n-k) \\ &= y(n-1) + \frac{1}{L}[x(n) - x(n-L)] \end{aligned} \tag{A.7}$$

The Z-transform of Equation A.7 gives the transfer function in Equation A.8.

$$\begin{aligned} Y(z) &= \frac{1}{L}X(z)(z^{-(L-1)} + \ldots + z^{-2} + z^{-1} + 1) \\ &= Y(z)z^{-1} + X(z)\frac{1}{L}[1 - z^{-L}] \\ H_1(z) &= \frac{1}{L}\frac{1 - z^{-L}}{1 - z^{-1}} \end{aligned} \tag{A.8}$$

The frequency response of Equation A.8 can be found by substituting z with $\mathrm{e}^{j2\pi fT}$. This gives the expression in Equation A.9 where T is the sampling time.

$$\begin{aligned} H_1(\mathrm{e}^{j2\pi fT}) &= \frac{1}{L}\frac{1 - \mathrm{e}^{-j2\pi fLT}}{1 - \mathrm{e}^{-j2\pi fT}} \\ &= \frac{1}{L}\frac{\mathrm{e}^{j\frac{2\pi fLT}{2}} - \mathrm{e}^{-j\frac{2\pi fLT}{2}}}{\mathrm{e}^{j\frac{2\pi fT}{2}} - \mathrm{e}^{-j\frac{2\pi fT}{2}}}\frac{\mathrm{e}^{-j\frac{2\pi fLT}{2}}}{\mathrm{e}^{-j\frac{2\pi fT}{2}}} \\ &= \frac{1}{L}\frac{\sin(\pi fLT)}{\sin(\pi fT)}\mathrm{e}^{-j\frac{2\pi f(L-1)T}{2}} \end{aligned} \tag{A.9}$$

Figure A.1 shows the magnitude and argument of $H_1(\mathrm{e}^{j2\pi fT})$ in Equation A.9 as function of the frequency f with $L = 400$ and $T = 0.05$ ms. The figure shows that the magnitude has zeros at 50 Hz and at all the

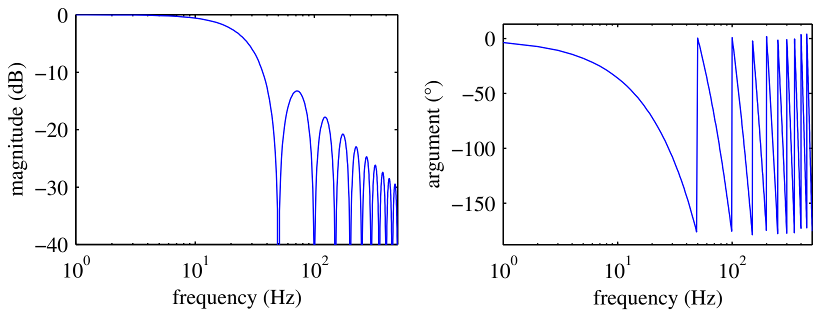


Figure A.1: Magnitude and argument of $H_1(e^{j2\pi fT})$ in Equation A.9 as a function of the frequency f.

higher harmonics. The argument seems to be linear with jumps from -180^o to 0^o every 50 Hz, and the jumps only occurs when the magnitude is zero. To get a better view of the function in Equation A.9 it is plotted in the complex plane with the frequency f as parameter in Figure A.2. The unit circle is shown with dotted line.

A.2.2 Low-pass filter.

An ideal integrator is a first order low-pass filter with a pole at zero. An approximation to this can be implemented as a cumulative summation.

$$\tilde{y}(n) = \tilde{y}(n-1) + x(n) \tag{A.10}$$

The DC amplification of this function is infinite so some kind of mean value extraction must be added. The main component of the integrand is known to be a 50 Hz sinusoid, so the mean value should at least cover one period at 50 Hz. If this mean value differs from zero it is regarded as undesired and is subtracted. This is described in Equation A.11 where L

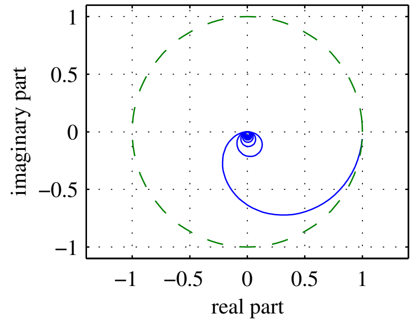


Figure A.2: Real and imaginary part of $H_1(e^{j2\pi fT})$ in Equation A.9 with the frequency f as parameter. The unit circle is shown with dotted line.

is the number of samples of one period at 50 Hz.

$$y(n) = \tilde{y}(n) - \frac{1}{L}\sum_{k=0}^{L-1}\tilde{y}(n-k) \tag{A.11}$$

The Z-transform of Equation A.10 is given as:

$$\begin{aligned}\tilde{Y}(z) &= \tilde{Y}(z)z^{-1} + X(z)\\ \tilde{Y}(z) &= \frac{1}{1-z^{-1}}X(z)\end{aligned} \tag{A.12}$$

and the Z-transform of Equation A.11 is given as:

$$\begin{aligned}Y(z) &= \tilde{Y}(z) - \frac{1}{L}\tilde{Y}(z)[1 + z^{-1} + z^{-2} + \ldots + z^{-(L-1)}]\\ &= \tilde{Y}(z)\left\{1 - \frac{1}{L}\left[1 + z^{-1} + z^{-2} + \ldots + z^{-(L-1)}\right]\right\}\end{aligned} \tag{A.13}$$

and the overall Z-transform can be found by combining Equation A.12

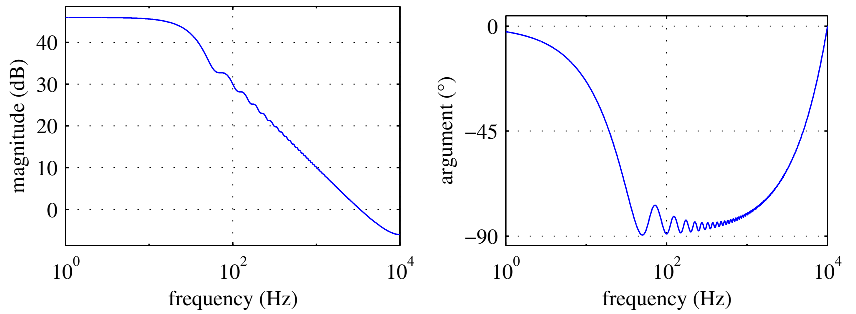


Figure A.3: Magnitude and argument of the transfer function in Equation A.14

and A.13:

$$\begin{aligned} H_2(z) &= \frac{Y(z)}{X(z)} \\ &= \frac{1}{1-z^{-1}}\left\{1-\frac{1}{L}\left[1+z^{-1}+z^{-2}+\ldots+z^{-(L-1)}\right]\right\} \end{aligned} \qquad \text{(A.14)}$$

The frequency response of $H_2(e^{j2\pi fT})$ is shown in Figure A.3. The magnitude response of H_2 seems to have a cutoff frequency of about 20 Hz. To find out where this cutoff comes from, the function $H_2(z)$ can be considered as consisting of two contributions as implied by Equation A.14: one from a low-pass term with a pole at $z = 1$ (the fraction), and one from the function in the braces. The magnitude response of the fraction is a function with 20 dB decay per decade, so the cutoff frequency comes from the function in the braces. This function is equal to 1 minus the Z-transform in Equation A.8 and as such it will look like the function in Figure A.2, only mirrored in the imaginary axis and shifted one unit to the right. The cutoff frequency (or 3 dB-frequency) is determined by this function through the parameter L. The decaying ripple of both the magnitude and argument in Figure A.3 is also explained hereby.

A.2.3 IIR filter designed using bilinear transformation.

This approach uses bilinear transformation to design a second order band-pass IIR filter [Ahmed and Natarajan, 1983]. The basis of the design is the Laplace transform of the second order band-pass filter in Equation A.15.

$$H(s) = \frac{\frac{s}{\omega_0}}{(1+\frac{s}{\omega_0})^2} \quad \text{, where } \omega_0 = 2\pi f_0 \qquad \text{(A.15)}$$

The Z-transform of Equation A.15 is found using bilinear transformation by substituting s with $\frac{z-1}{z+1}$ in Equation A.15. This gives the Z-transform in Equation A.16.

$$H_3(z) = \frac{\omega_0}{(\omega_0+1)^2}\cdot\frac{1-z^{-2}}{1+\frac{\omega_0-1}{\omega_0+1}z^{-1}+(\frac{\omega_0-1}{\omega_0+1})^2 z^{-2}} \qquad \text{(A.16)}$$

By inspection the Z-transform in Equation A.16 can be transformed to the difference equation in Equation A.17.

$$\begin{aligned} y(n) =& \frac{\omega_0}{(\omega_0+1)^2}(x(n)-x(n-2)) - \\ & \frac{\omega_0-1}{\omega_0+1}y(n-1) - \left(\frac{\omega_0-1}{\omega_0+1}\right)^2 y(n-2) \end{aligned} \qquad \text{(A.17)}$$

Again by substituting z with $e^{j2\pi fT}$ the frequency response of the Z-transform in Equation A.16 is calculated and shown in Figure A.4. The center frequency f_0 for the band-pass filter is set to 0.1 Hz.

The time delay through the integrator is important because only the signal from the current sensor is integrated, and it is crucial to have synchronism between voltage and current signals. The group delay of Equation A.16 is defined as $-\frac{d\varphi}{d\omega}$ where φ is the argument of $H_3(e^{j2\pi fT})$ in Figure A.4. An approximation to this is calculated numerically as given

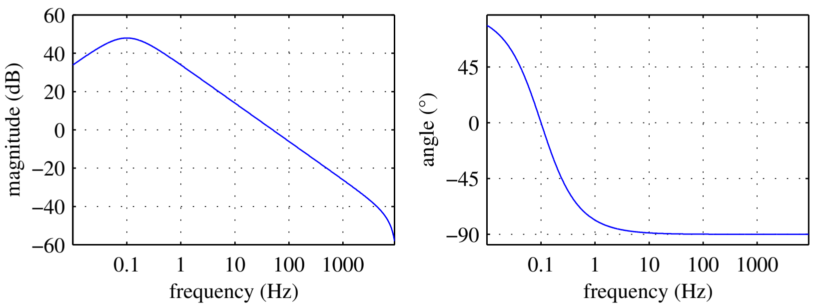


Figure A.4: Magnitude and argument of transfer function in Equation A.16.

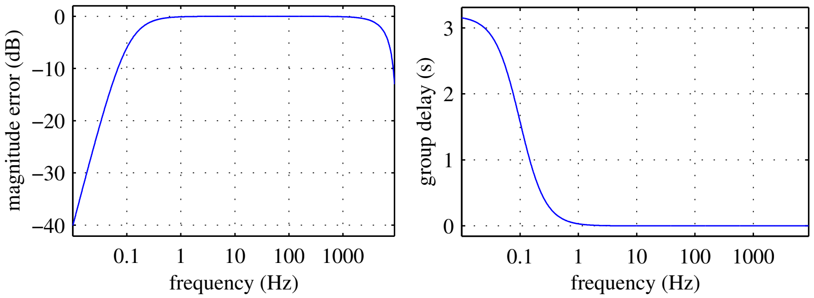


Figure A.5: Group delay of transfer function in Equation A.16.

by Equation A.18 and the result is shown in Figure A.5.

$$
\begin{aligned}
d(n) &= -\frac{\varphi(n) - \varphi(n-1)}{2\pi(f(n) - f(n-1))} \\
\text{where} \quad & \varphi(n) = \arg(H_3(\mathrm{e}^{j2\pi f(n)T}))
\end{aligned} \tag{A.18}
$$

The impulse response of Equation A.17 is shown in Figure A.6 in two views; the left with a time scale from 0 s to 10 s, and the right where only

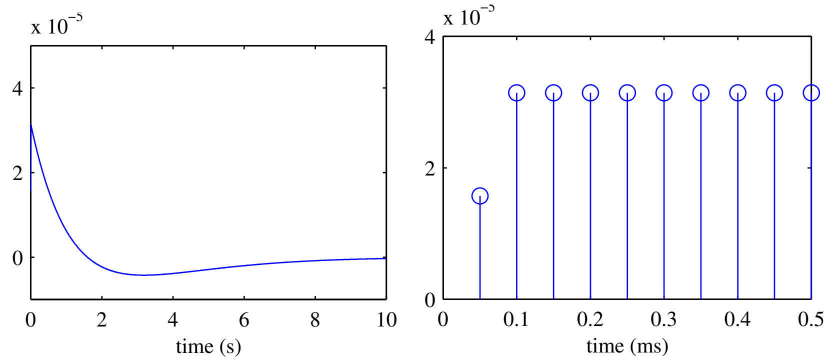


Figure A.6: Impulse response of Equation A.17.

the first 10 samples is shown. The impulse response of an ideal integrator is a step function. Apart from the first sample, this seems to be true for the first few samples.

A.2.4 Discussion.

The frequency interval of interest is from a few Hz to some kHz. The designs described in this section will be evaluated with respect to this frequency interval.

The integration of a periodic signal as described in Section A.2.1 has a very bad magnitude response (Figure A.1) with zeros for every 50 Hz (when $L = 400$ and $T = 50\mu$s). This does not at all look like the ideal magnitude response of −20 dB of the ideal integrator.

The low-pass filter with mean value extraction described in Section A.2.2 has a better magnitude response (Figure A.3) although it still has some ripple. The argument is only a rough approximation to −90° in a narrow frequency interval.

The IIR filter in Section A.2.3 has a very good magnitude response (Figure A.4). Between 1 Hz and 2–3 kHz this magnitude response is exactly $-20\,\mathrm{dB}$ per decade as the ideal integrator apart from a constant factor. The argument of the frequency response in Figure A.4 is very close to -90° for frequencies above 20–30 Hz. The important thing in this context is the group delay of the filter. The group delay should be well below the sampling time $T = 0.05\,\mathrm{ms}$.

According to the right plot of Figure A.5 this is the fact for frequencies above 50 Hz. For frequencies below 50 Hz the group delay is larger than the sampling time and at 5 Hz the group delay is 1.25 ms.

When the analysis with the DISMO-toolbox (see Section D.8) is run it might be sufficient with the strict synchronism between the fundamental component and the high frequency part of the signal. The analysis of the frequencies below 50 Hz might be carried out separately from the other parts of the signal, so this design of the integrator is accepted.

If the future proves it necessary to have synchronism between the low frequency parts of the voltage and current signals an appropriate all-pass filter might be able to correct this problem.

Appendix B

Transfer Function for Π-section

If a cable model consists of a large number of Π-sections the two shunt admittances from neighboring sections can be replaced by one element with twice the admittance. If one of these sections are far from the ends of the model the input impedance Z_{in} for the section must be equal to the load impedance as shown by Figure B.1.

The impedance Z_{in} can be found by solving following equation for Z_{in}.

$$
\begin{aligned}
Z_{\mathrm{in}} &= Z + \frac{1}{Y} \parallel Z_{\mathrm{in}} = Z + \frac{\frac{1}{Y} Z_{\mathrm{in}}}{\frac{1}{Y} + Z_{\mathrm{in}}} = Z + \frac{Z_{\mathrm{in}}}{1 + Y Z_{\mathrm{in}}} \\
&= \frac{Z(1 + Y Z_{\mathrm{in}}) + Z_{\mathrm{in}}}{1 + Y Z_{\mathrm{in}}} = \frac{Z + (ZY + 1) Z_{\mathrm{in}}}{1 + Y Z_{\mathrm{in}}} \\
\Leftrightarrow \quad Z_{\mathrm{in}}(1 + Y Z_{\mathrm{in}}) &= Z + (ZY + 1) Z_{\mathrm{in}} \\
\Leftrightarrow \quad Z_{\mathrm{in}}^2 - Z Z_{\mathrm{in}} - \frac{Z}{Y} &= 0 \\
\Rightarrow \quad Z_{\mathrm{in}} &= \frac{Z}{2} + \sqrt{\frac{Z^2}{4} + \frac{Z}{Y}}
\end{aligned}
\tag{B.1}
$$

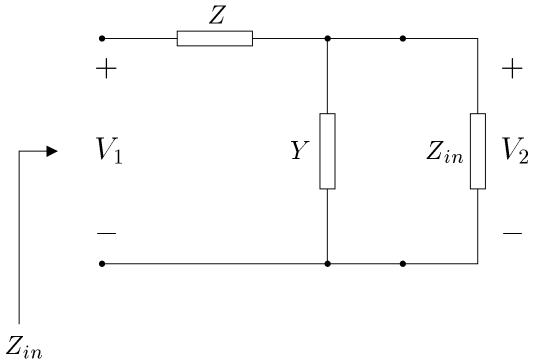


Figure B.1: Model of Π-equivalent.

where

$$
\begin{aligned}
Z &= L(r + j\omega l) \\
Y &= L(g + j\omega c)
\end{aligned} \tag{B.2}
$$

When the length L of the cable that the Π-equivalent represents approaches zero, it is seen from Equation B.1 that in the limit Z_{in} becomes the characteristic impedance Z_0 for the cable. That is

$$
\lim_{L \to 0} Z_{\text{in}} = \sqrt{\frac{Z}{Y}} \tag{B.3}
$$

With reference to Figure B.1 the transfer function for a Π-section is given by

$$
\begin{aligned}
H(j\omega) = \frac{V_2}{V_1} &= \frac{\frac{1}{Y} \parallel Z_{\text{in}}}{Z + \frac{1}{Y} \parallel Z_{\text{in}}} = \frac{\frac{\frac{1}{Y} Z_{\text{in}}}{\frac{1}{Y} + Z_{\text{in}}}}{Z + \frac{\frac{1}{Y} Z_{\text{in}}}{\frac{1}{Y} + Z_{\text{in}}}} = \frac{\frac{Z_{\text{in}}}{1 + Y Z_{\text{in}}}}{Z + \frac{Z_{\text{in}}}{1 + Y Z_{\text{in}}}} \\
&= \frac{Z_{\text{in}}}{Z(1 + Y Z_{\text{in}}) + Z_{\text{in}}} = \frac{Z_{\text{in}}}{Z + Z_{\text{in}}(1 + Y Z)}
\end{aligned} \tag{B.4}
$$

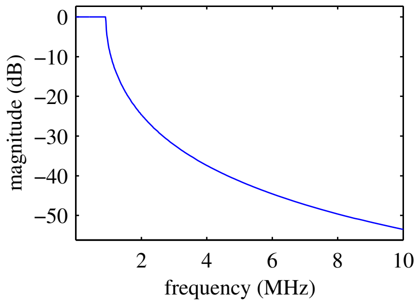


Figure B.2: Magnitude of transfer function in Equation B.4.

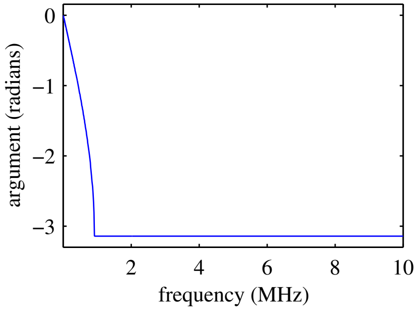


Figure B.3: Argument of transfer function in Equation B.4.

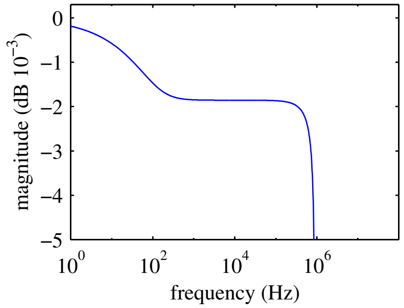


Figure B.4: A zoom on the Magnitude of transfer function in Equation B.4.

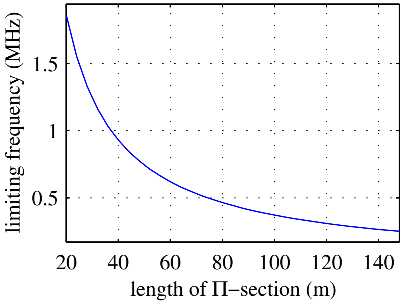


Figure B.5: Limiting frequency as a function of the length L of the Π-section.

Figure B.2 and B.3 shows the magnitude and argument of the transfer function in Equation B.4. Figure B.4 shows a zoom on the first vertical part of magnitude in Figure B.2. The frequency where the response drops may be called the cutoff frequency, or the limiting frequency for the Π-section. This frequency is computed for different lengths L of the Π-section is plotted in Figure B.5.

Appendix C

ATP Network Models

This appendix describes the ATP models used for impulse response generation and for simulation of ground faults utilized in this thesis. The fundamental structure and syntax of ATP input files are not treated here, but a full documentation can be found in the Rulebook [Leu, 1987].

Batch scripts and program commands in this appendix will be shown in Unix style.

C.1 The Primary Input File

The ATP simulations are used in connection with the impulse response representation of the network as described in Section 5.2 on page 48, and as such an extremely large number of simulations have to be performed on almost identical models. The only difference between these simulations is the location of either the impulse source or the ground fault switch. The ATP input file for each simulation is therefore only a command to include a main input file with a node name as parameter, which represents the location where either the impulse source or the ground fault switch is connected.

There are very little difference between the impulse response and the ground fault simulations, so the model will be described in terms of the impulse response simulation, and in Section C.3 the difference between these two simulation types will be described.

For node `AA000` the input file is called `iaa000.atp` and looks like:

iaa000.atp

```
$INCLUDE, ia12.atp, AA000
```

Prefix `i` in the file name denotes an impulse response simulation and `atp` is the extension used for all ATP circuit files in this thesis. The `$INCLUDE` command includes the main file `ia12.atp` in the data case with the parameter `AA000`. Again prefix `i` denotes an impulse response simulation, and `a12` is the name of the feeder which is modeled (see 1.1 on page 3).

C.2 The Main Input File

The main input file `ia12.atp` for the A12 feeder is listed below:

ia12.atp

```
1  KARD  6  7
2  KARG  1  1
3  KBEG  3  3
4  KEND  7  7
5  KTEX  1  1
6  BEGIN NEW DATA CASE
7  POWER FREQUENCY                    50.00
8  C DELT >< TMAX >< XOPT >< COPT ><EPSILN><TOLMAT><TSTART>
9   2.0E-06   0.004
10 C  IOUT>< IPLOT><IDOUBL><KSSOUT><MAXOUT>< IPUN ><MEMSAV>< ICAT ><NENERG><IPRSUP>
11        1       1       0       3       0       0       0       0       0
12 C Low-pass filtered impulse source at phase r
13 C   Node     : defined by $INCLUDE parameter.
14 C   Direction: from phase to ground.
15 /SOURCE
16 C < n 1><>                                                    <  start ><  stop  >
17  1_NOD_R-1                                                           0.     0.004
18  1_NOD_G-1                                                           0.     0.004
19 $INCLUDE, /phd/kjn/matlab/downsmpl/h10.dat
20 /SWITCH
21 C < n 1>< n 2>< Tclose ><Top/Tde ><   Ie   ><Vf/CLOP ><  type  >
22   GLN1_RAA000R                                          MEASURING                1
```



```
23   GLN1_SAA000S                                          MEASURING                0
24   GLN1_TAA000T                                          MEASURING                0
25   GLN1_GAA000G                                          MEASURING                0
26 $INCLUDE, ../a12.dat
27 /BRANCH
28 C Definition of source impedance.
29 C <BUS1><BUS2><BUS3><BUS4><res ><ind ><cap >
30  0      GLN1_R           .00907.34621
31  0      GLN1_S           .00907.34621
32  0      GLN1_T           .00907.34621
33  0      GLN1_G            100.0
34 C The arc-supression coil is attached to the network.
35 C The definitions are courtesy of Mr. Harald Wehrend, Hannover University.
36 $INCLUDE, ../a12_asc.atp
37 BLANK CARD TERMINATING BRANCHES
38 BLANK CARD TERMINATING SWITCHES
39 BLANK CARD TERMINATING SOURCES
40 BLANK CARD TERMINATING OUTPUT
41 BLANK CARD TERMINATING PLOT
42 BEGIN NEW DATA CASE
43 BLANK CARD ENDING TOTAL EMTP INPUT
```

The first five lines control the insertion of the parameter in the file. This is done in line 17 and 18 (line 6 and 7 not counting comment lines and the first five lines) from column 3 to 7. The supporting routine *DATA BASE MODULE* can be useful to generate the parameters in the first five lines. If `ia12.atp` is included from `iaa000.atp` the node names in line 17 and 18 from column 3 to 8 will expand to `AA000R` and `AA000G` respectively.

Line 17, 18, and 19 defines the impulse source as a type-1 source (see Rulebook). This source is user specified for all time steps and the actual values are here included in line 19. The file `h10.dat` is generated by the Matlab function `WRITESRC` (see Appendix D.6) and it contains the impulse response of a low-pass filter with a cut-off frequency of one 10^{th} of the half sampling rate ($\frac{f_s}{20}$). In ATP all sources are connected between a node and `TERRA` (global reference node) so to connect the source between node `AA000R` and `AA000G` two sources with opposite signs must be connected between the respective nodes and `TERRA`.

The top node of the cable network is node `AA000`, so this node is identical to the observation point. Lines 22–25 makes the connection between the cable network and the primary substation at node `GLN1_`.

In line 26 the cable network is include from file a12.dat. This file is generated by the makenet program (see Appendix E.1 on page 164) and defines the full cable network. Input for makenet is discussed in Appendix C.6.

At lines 30–33 the power source impedance is defined. This impedance is connected between node GLN1_ and TERRA (default when node name is omitted) because the power source must be short circuited during the impulse response generation. This source impedance represents the impedance seen into from the 10 kV network towards the power source. The actual values are taken from [Munk, 1995] except for the 100 Ω resistance in line 33, which represents the resistance between the ground at the primary substation GLN1_G and the ground at the power generator TERRA. The actual value of this resistance is a simple guess.

Line 36 includes the file a12_asc.atp which is the definition of the Petersen coil (arc suppression coil) and the Zy grounding transformer as described in Section 4.3 on page 35. This definition is copied directly from [Munk, 1995] except for the inductance of the Petersen coil, which is calculated as given in Section 4.3.

C.3 Ground Fault Simulation Model

This section describes the three properties of the main input file that are different between the impulse response and the ground fault simulation. The prefix for this input file is a g for ground fault simulation.

ga12.atp

```
8  C DELT >< TMAX >< XOPT >< COPT ><EPSILN><TOLMAT><TSTART>
9   1.0E-06 4.0E-03
10 C  IOUT>< IPLOT><IDOUBL><KSSOUT><MAXOUT>< IPUN ><MEMSAV>< ICAT ><NENERG><IPRSUP>
11        1       1       0       3       0       0       0       0       0
12 C Ground fault switch and resistor
13 /SWITCH
14 C < n 1>< n 2>< Tclose ><Top/Tde ><   Ie   ><Vf/CLOP ><  type  >
15   __N__RGFNODE   2.0E-03    1.0E99
16 /BRANCH
17 C < n 1>< n 2><ref1><ref2>< R  >< L  >< C  >
18  OGFNODE__N__G            1.0E+0
19 /SWITCH
20 C < n 1>< n 2>< Tclose ><Top/Tde ><   Ie   ><Vf/CLOP ><  type  >
```



```
21   GLN1_RAA000R                                                    MEASURING        1
22   GLN1_SAA000S                                                    MEASURING        0
```

Line 12–19 in ia12.atp are replaced by line 12–18 in ga12.atp where a switch (in line 13) and a resistor (in line 18) are defined.

ga12.atp

```
27 C Definition of source impedance.
28 C <BUS1><BUS2><BUS3><BUS4><res ><ind ><cap >
29  OGLNS_RGLNT_R            .00907.34621
30  OGLNS_SGLNT_S            .00907.34621
31  OGLNS_TGLNT_T            .00907.34621
```

Next the source impedance is connected between node GLNS_ and GLN1_ in line 29–31 of ga12.atp instead of the grounded source in line 30–32 in ia12.atp

ga12.atp

```
38 C BEGIN Definition of sources:
39 C Sinusodal generators, 10kV
40 C <BUS1>VC<AMPLITUD><FREQUENC><TIME-0  ><A1      ><TIME-1  ><TSTART  ><TSTOP   >
41 14GLNS_R 0    8165.0      50.0     140.0                            -1.0
42 14GLNS_S 0    8165.0      50.0    -100.0                            -1.0
43 14GLNS_T 0    8165.0      50.0      20.0                            -1.0
44 C 1:1 Yd transformer
45 11GLNT_R     1.0E-20
46 18               1.0GLN1_RGLN1_S
47 11GLNT_S     1.0E-20
48 18               1.0GLN1_SGLN1_T
49 11GLNT_T     1.0E-20
50 18               1.0GLN1_TGLN1_R
51 BLANK CARD TERMINATING BRANCHES
52 BLANK CARD TERMINATING SWITCHES
```

and last the 10 kV power source in line 41–43 of ga12.atp is connected to the network through a 1:1 ideal transformer in line 45–50 of ga12.atp. This transformer is needed to allow the voltage on one phase to be shorted to ground while keeping the voltage between the phases.

C.4 Node Naming Conventions

ATP allows only node names with six characters in upper case. The first character denotes to which cable section the node belongs. A cable section

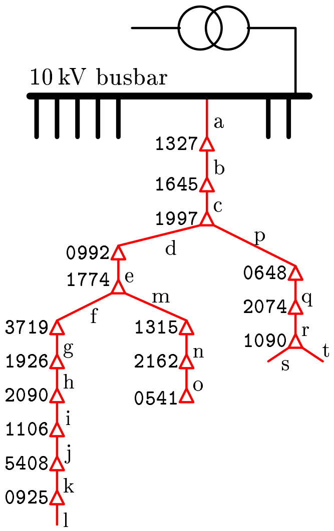


Figure C.1: Cable section names for the A12 feeder.

is the part of the network that connects two transformers. The section names for A12 are the letters at the right side of the network in Figure C.1.

Each of these sections consist of several different types of cable, so the second character of the node name denotes to which cable subsection the node belongs.

Character three to five represents the distance in meters to the end of the subsection nearest the observation point at the primary substation.

Character number six in the node name represents the phase, which is called R, S, T, or G. Phase G represents the distributed ground as described in Section 4.1.5 on page 28.

This means that the node names conform to the following format: '`ssdddp`', where '`ss`' is the section and subsection characters, '`ddd`' is the three digit distance to the end of the subsection, and '`p`' denotes which phase of the system the node belongs to.

As an example, node `PB160S` is 160 m along phase S of the second subsection (`B`) of section `P`. From Figure C.1 it is seen that section `P` is the section between distribution transformer `1997` and `0648`.

C.5 File Naming Conventions

The names of the data files (primary input files) closely follow the node names, as each of the primary input files correspond to a given node in the network. The only differences between the node names and the file names are that node names are in uppercase whereas file names are in lowercase, file names have a prefix and an extension, and the file names omit the sixth character in the node name which denotes the phase. All impulse response and ground fault simulations are simulated at phase R by convention. This choice is arbitrary as the model is perfectly symmetrical.

If '`ssdddp`' is the format of the node name as described in Section C.4, '`tssddd.eee`' is the format for the file name of the primary input file. The simulation type is denoted with a '`t`' in the format string and is either an '`i`' for impulse response or a '`g`' for ground fault. The '`ssddd`' is the same as for the node name only in lower case, and the '`eee`' is the extension '`atp`' used here for the ATP input files.

Using the same example as in Section C.4, a ground fault simulation at node `PB160R` would have a primary input file called `gpb160.atp` and the input file for the impulse response simulation would be called `ipb160.atp`.

C.6 Cable Network Model

As described above, the network models contains a very large number of Π-sections, so editing the network input file by hand is not a good solution. Therefore a program `makenet` was written to generate the input file. Appendix E.1 on page 164 describes describes how to run the program. The program reads an input file with the extension `net` which defines the properties of the network, and writes two files. One file with the ATP model of the network (with extension `dat`) and a log file (with extension `log`) with messages of all actions and with names of all the nodes in the network. The `LOG` file has the additional purpose of serving as input file for the atpinput program which writes all the primary input files. This process is discussed in Section C.8.

C.6.1 Input File Syntax for makenet

Following list of keywords are allowed in the input file for makenet. Note that the first four keywords makes initializations and must be given in this order as shown in the following example.

`SetGroundResistance`: A resistance in Ω/m must be given. This resistance is inserted in the ground phase of the Π-section as described in Section 4.1.5 on page 28.

`SetPiLength`: The length of each Π-section in meters must be given.

`SetCableDir`: Optional keyword that can be followed by a directory with the cable definitions.

`DefineCable`: This keyword takes three parameters. The name of the ATP output file with the impedance and admittance matrices (generation of these files is discussed in Appendix D.7 on page 141). This file name is relative to the directory given by keyword `setcabledir`. Next an alias for the cable for reference in the makenet input file, and last a four character upper case node name for the ATP input file (makenet output file). This node name makes it possible to read the final ATP output file for debugging purposes.

`NewCableSection`: This keyword also takes three parameters. A two character node name in upper case that represents the cable subsection in the network, the length of the subsection in meters, and the alias of the cable type as given by the keyword `definecable`. The subsection name is described in Section C.4. The length of the cable subsection will be rounded off to equal an integer multiple of the Π-section length. If a cable subsection is rounded off to zero, makenet will issue a warning.

`AddSplit`: No parameters. This keyword makes a branch in the network and must be followed by a corresponding `usesplit` keyword. This is only for error checking of the interconnection of the network.

`UseSplit`: A two character upper case subsection name must be given. This name gives the subsection to which end the following cables will be connected (with the `newcablesection` keyword). This node name must correspond with the location of a `addsplit` keyword.

`Load`: This keyword takes two parameters. The load in kW and a power factor as $\cos(\varphi)$. The load is implemented as three delta connected parallel connections of a resistor and an inductor.

`ExternNode`: This keyword takes a node name as parameter. This is for external connection outside of the makenet generated file. The name must be 5 characters long and will be concatenated with the letters R, S, T, and G to give four terminals for the connection.

Following is a few examples of these keywords in the input file `a12.net` for the A12 network. The first 13 lines is initialization of the Π-section properties and cable definitions. The rest of the file is the definition of the network. Comments must be preceded by a % character and empty lines are ignored.

a12.net

```
1  % Definition of the A12 feeder at Glentegaarden
2  SetGroundResistance 0.000625
3  SetPiLength 40
4  % Use cable parameters computed at 10kHz
5  SetCableDir /phd/kjn/matlab/cabledef/f010000
6  DefineCable a24a.lis Al240PEX A24P
7  DefineCable a24a.lis Al240APB A24A
8  DefineCable a15a.lis Al150PEX A15P
9  DefineCable a15a.lis Al150APB A15A
10 DefineCable c15a.lis Cu150APB C15A
11 DefineCable a95a.lis Al95APB  A95A
12 DefineCable c95a.lis Cu95APB  C95A
```

```
13 DefineCable c50a.lis Cu50APB  C50A
14 % GLN -1327
15 NewCableSection AA  79 Al240PEX
16 NewCableSection AB 654 Cu95APB
17 NewCableSection AC 114 Cu150APB
```

In line 2 the resistance in the ground phase is set to $0.625\,\mathrm{m}\Omega/\mathrm{m}$ and in line 3 the length of the Π-section is set to 40 m. In line 5 the directory for the cable parameters are set to `/phd/kjn/matlab/cabledef/f010000`. In this directory the cable parameters are computed at 10 kHz. By changing this directory, other cable definitions can easily be switched to. Generation of these cable definition files is discussed in Appendix D.7 on page 141. Line 6 to 13 defines all the cable types for the following network definition. From line 15 the network definitions starts and the first cable section will start with node `AA000` and is a 79 m PEX cable with a $240\,\mathrm{mm}^2$ aluminum core.

a12.net

```
25 NewCableSection CA 181 Al150PEX
26 NewCableSection CB 306 Al150APB
27 Load 164 0.95
28
29 AddSplit
30
31 % 1979 - 0992
32 NewCableSection DA  31 Al150APB
```

In line 27 a load of 164 kW and a power factor of 0.95 is defined. This load succeeds cable subsection `CB` and will be connected to the last Π-section of this cable subsection. Line 29 makes a branch at the junction of cable subsection `CB` and `DA`

a12.net

```
84 NewCableSection OA 458 Cu95APB
85 Load 164 0.93
86
87 UseSplit CB
88
89 % 1979 - 0648
90 NewCableSection PA  88 Al95APB
```

and in line 87 the branch is completed with the `UseSplit` keyword. If the `AddSplit` keyword is not followed by a corresponding `UseSplit` keyword

makenet will issue an warning.

C.6.2 Network Data at A12

In Table C.1 all cable sections of A12 are listed together with the length and the cable type. The first column lists the two transformer stations that is connected by the cable. The second column is the first two characters of the node name in the ATP input file as given by the makenet input file `a12.net`. The third column is the length of the cable and the fourth column is the cable type.

C.6.3 Network Data at the 10 kV Laboratory

In Table C.2 all cable sections of the 10 kV laboratory at Kyndby are listed with the length and the type of each section. Note that the network has a special topology which means that transformer station `6464` is at one end of all sections except for the last one. See Figure 6.2 on page 70 for an overview of the network.

Two configurations of this network were used during the experiments described in Chapter 6: branched and a non-branched as shown in Figure 6.3 and 6.4 on page 72. In the makenet input file the non-branched configuration (configuration 1) is generated by inserting the cable sections in Table C.2 in the listed order using keyword `NewCableSection`. The branched configuration (configuration 2) is generated by adding the `AddSplit` keyword after subsection `BE` and a `UseSplit BE` before subsection `EA`.

C.7 Conversion of the ATP output file

ATP writes output to a `LIS` file which contains a descriptive interpretation of all input data. With the `KSSOUT` variable set to 3 as in line 11 of `ia12.atp` on page 114 the numeric output goes to the `LIS` file. To convert the `LIS` file to a `MAT` file which can be read by Matlab, the atp2mat program was written (see Appendix E.2.1 on page 165). The advantage of the `MAT`

Section	Name	Length (m)	Cable Type
GLN–1327	AA	79	Al 240 PEX
	AB	654	Cu 95 APB
	AC	114	Cu 150 APB
	AD	363	Cu 95 APB
1327–1645	BA	246	Cu 95 APB
	BB	181	Al 150 PEX
1645–1997	CA	181	Al 150 PEX
	CB	306	Al 150 APB
1979–0992	DA	31	Al 150 APB
	DB	17	Cu 95 APB
	DC	52	Al 150 APB
0992–1774	EA	38	Al 240 APB
	EB	10	Cu 150 APB
	EC	821	Cu 95 APB
1774–3719	FA	260	Cu 50 APB
	FB	73	Al 95 APB
3719–1926	GA	71	Al 95 APB
	GB	457	Cu 50 APB
1926–2090	HA	149	Cu 50 APB
2090–1106	IA	215	Cu 50 APB
1106–5408	JA	115	Cu 50 APB
	JB	57	Al 95 APB
5408–0925	KA	58	Al 95 APB
	KB	267	Cu 50 APB
0925–0633	LA	110	Cu 50 APB
	LB	7	Al 95 APB
	LC	53	Cu 50 APB
	LD	6	Al 95 APB
	LE	124	Cu 50 APB
	LF	4	Al 95 APB
	LG	51	Cu 50 APB
1774–1315	MA	476	Cu 95 APB
1315–2162	NA	481	Cu 95 APB
2162–0541	OA	458	Cu 95 APB
1979–0648	PA	88	Al 95 APB
	PB	452	Cu 50 APB
0648–2074	QA	181	Cu 50 APB
	QB	23	Al 95 APB
2074–1090	RA	22	Al 95 APB
	RB	178	Cu 50 APB
1090–1315	SA	599	Cu 50 APB
1090–1344	TA	360	Cu 50 APB

Table C.1: Cable data for A12.

Section	Name	Length (m)	Cable Type
6464–6468	AA	343	Al 95 PEX
	AB	4	Al 150 APB
	AC	381	Cu 50 APB
	AD	965	Cu 25 OH
	AE	340	Cu 35 OH
6468–6464	BA	308	Cu 35 OH
	BB	270	Cu 50 OH
	BC	900	Cu 50 OH
	BD	278	Cu 50 APB
	BE	193	Al 95 PEX
6464–6467	CA	181	Al 95 PEX
	CB	31	Al 150 PEX
	CC	612	Al 240 APB
6467–6464	DA	512	Cu 50 APB
	DB	239	Al 95 PEX
6464–6466	EA	759	Al 240 PEX
	EB	149	Al 95 PEX
6466–6465	FA	26	Cu 50 APB
	FB	165	Al 95 PEX
	FC	5	Al 150 APB
	FD	635	Cu 95 APB

Table C.2: Cable data for Kyndby network file format is that it is well documented and easy to read and write and it saves disk space compared to the ASCII file format (the `LIS` file).

A simulation will often have to run at a larger sampling frequency than needed for the final data. This means that the ATP output has to be low-passiltered and decimated. In order run the simulations from a batch script the downsmpl program was written to perform this task, see Appendix E.3 on page 167. To produce a single impulse response following script, runsim, can be run.

runsim

```
1 #!/bin/sh
2 /usr/local/atp/bin/tpbig disk $1. $1. -r
3 atp2mat $1.lis
4 rm $1.lis
5 downsmpl $1.mat /phd/kjn/matlab/downsmpl/h10.mat
```

This script takes the name of the primary ATP input file without extension as parameter. Line 2 of runsim runs the ATP program (which is called tpbig), and line 3 converts the `LIS` file to a `MAT` file. Line 4 removes the `LIS` file (otherwise they may fill up the disk) and line five performs the low-pass filtering and decimation.

C.8 The Complete Impulse Response Model

To generate a complete model of impulse responses for the A12 feeder, following tasks must be performed:

- Edit the ia12.atp and the a12.net
- Run makenet on a12.net
- Run atpinput in a12.log
- Run the batch script produced by atpinput

If the `a12.net` is in current directory and `ia12.atp` is in directory `imp`, this procedure can be summarized in the following sequence of commands:


```
makenet a12
cd imp
atpinput ../a12.log -m ia12.atp -p i
runall
```

The `../a12.log` is the input file for atpinput written by makenet. The parameter '`-m ia12.atp`' tells atpinput that `ia12.atp` is the main input file, and the parameter '`-p i`' sets the file prefix for the primary input files to `i`. The last command runs the batch file written by atpinput. This batch file may be run with low priority ('`nice`') and in the back ground with the command '`nice runall &`'.

Appendix D

Matlab Functions

The primary signal processing tool used in this project is Matlab from MathWorks. Matlab is a matrix based computing tool with extensive graphical visualization possibilities and a high degree of flexibility.

This appendix provides a description of some of the most important Matlab functions developed during this project. This is not a full documentation of the Matlab code but should be regarded as a highlight of the central parts of the code. It is the intention to provide a basis for future projects to utilize the material presented in this thesis.

The functions described here represent several thousand lines of code and is therefore impossible to list verbatim. It is therefore assumed that the full version of the code has been provided to the reader by other means and that the reader is familiar with the Matlab programming syntax.

The code is written for Matlab 5.x but many parts of the code will be compatible to Matlab 4.2 as the transition to 5.x was made at a late stage in the project.

Those parts of the code that is listed has line numbers that refer to the original file and they are written in a box with the file name at the top.

Note that this is research code under development and not an official release, therefor the code has not been cleaned for out-commended lines, and inconsistencies may occur.

For an overview of the Matlab functions in this appendix, following list provides a brief description of the functionality and a reference to the section and page number in this appendix where the documentation can be found. In addition the Index on page 197 contain references for all Matlab functions.

D.1 Variable Naming Conventions

Naming of variables have mainly been done according to the principle known as Hungarian Naming. This means that a variable name has a lower case prefix indicating the type, so e.g. an integer variable counting lines might be called `iLine` and a string holding a file name could be called `strFileName`.

D.2 Ground Fault Localization Algorithm

The ground fault localization algorithm in Equation 5.20 on page 57 is implemented in the function `GFERR`.

Input

Command line parameters for the function are a predefined string that determines the action, a directory name pointing to a set of impulse response simulations, and a directory name pointing to one or more ground fault signals. The help for the function (invoked by "`help gferr`") gives information on the parameters for the function.

All input data (impulse response and ground fault signals) must be in Matlab data format (`MAT`).

Output

Output of the function is saved to disk as a `MAT` file, `gferr.mat`, containing the error measure given by Equation 5.20 together with some information on node names and network interconnections. This `MAT` file can then be used by e.g. `SHOWERR` to visualize the error measure.

Implementation

`GFERR` provides several possibilities of plotting intermediate data, specifying input directories, etc. This is described in the function help.

Line 129 and 130 retrieves the file names for each of the impulse response simulations and the ground fault signals using the `GETAMF` (see Section D.5).

gferr.m

```
129 strImpFile = getamf(strImpRoot);
130 strGfFile  = getamf(strGfRoot);
131 eval(['load ' strConnectFile])
```

Next the impulse responses for the network is loaded into a matrix, `dImp`. The energy center is computed for debugging purposes, in order to check whether the impulse has been captured within the signal, i.e. if the time duration of the simulation is long enough for this particular network.

gferr.m

```
191 for nImp = 1:NImpNode
192   strImpName = [strImpRoot strImpFile(nImp,:) '.mat'];
193   eval(['load ' strImpName])
194   % Find center of energy
195   dEnergy = x.*x;
196   dEnergySum = cumsum(dEnergy);
197   iMaxTmp = min(find(dEnergy > 0.9*max(dEnergy)));
198   if iMaxTmp > iMax, iMax = iMaxTmp; end
199 %  dImp(:,nImp) = x*dDownSamplingScale;
200   dImp(:,nImp) = x;
201   dImpFft(:,nImp) = addwin(dImp(:,nImp),Nfft);
202   dImpFft(:,nImp) = fft(dImpFft(:,nImp));
203 end
```

Line 256 to 296 is a for-loop over all ground fault signals. Most of these lines are related to plotting intermediate data, however, in line 267 the file name is retrieved, in line 268–270 a sub-function `ComputeAllError` computing two matrices (`Fftgfm` and `Fftcim`) is invoked, and in line 271 the error measure for the ground fault is computed from these matrices.

gferr.m

```
267   strGfName = [strGfRoot strGfFile(nGf,:) '.mat'];
268   [Fftgfm,Fftcim] = ComputeAllError(strGfName,nGf,strGfFile,strImpFile,dImp,...
269       dImpFft,freq,fIndx,hpFir,NHpFir,Nfft,...
270       bPlotImp,nFigImp,bPlotFft,nFigFft,bPlotModel,nFigModel);
271   dErr(:,nGf) = [sum(abs(Fftgfm-Fftcim))/size(Fftgfm,1)]';
```


`Fftgfm` and `Fftcim` correspond to the numerator and the denominator of Equation 5.19 on page 56. Each column in the matrices is the log-magnitude spectrum for a location in the network given by the impulse response. This means that the columns correspond to index k in Equation 5.19 and the rows correspond to index i. The summation in line 271 is performed along the columns and implements the summation in Equation 5.20 on page 57. The result is a row vector with the error measure for each node i in the network.

If data is to be saved to `gferr.mat` this is done in line 299. Four variables are saved:

`dErr` matrix with error measure for all ground faults and for all nodes

`strImpNode` string matrix with node names of all impulse responses

`strGfNode` string matrix with node names of all ground faults

`cConnect` char matrix describing the interconnection of all sections in the network

gferr.m

```
298 if bSaveData == 1
299   eval(['save ' strGferrFile ' dErr strImpNode strGfNode cConnect'])
300 end
```

In line 324 `ComputeAllError` is defined as

gferr.m

```
323 %==================================================================
324 function [Fftgfm,Fftcim] = ComputeAllError(strGfName,nGf,strGfFile,...
325       strImpFile,dImp,...
326       dImpFft,freq,fIndx,hpFir,NHpFir,Nfft,...
327       bPlotImp,nFigImp,bPlotFft,nFigFft,bPlotModel,nFigModel)
```

Most of the parameters to this function are used to control the graphical output and even if it makes the code less readable it proved very useful to be able to follow the computations in various details. E.g. this means that it is possible to watch the numerator and denominator of Equation 5.19

as the nodes i is stepped through. The most important input parameters are:

`strGfName` file name of the ground fault signal

`dImp` matrix with the network impulse responses

`dImpFft` matrix with the FFT of the network impulse responses

`iIndx` index corresponding to the frequencies for which the error measure will be computed

`hpFir` high-pass filter

In line 336 the ground fault signal is loaded and in line 338–340 the transient is extracted and high-pass filtered.

gferr.m

```
335  % Load ground fault
336  eval(['load ' strGfName])
337  % Get transient
338  x = x(80:length(x)); % Remove transient from downsampling preprocessing
339  hpx = filter(hpFir,1,(x-mean(x)));%.*hanning(length(x)));
340  hpx = hpx(NHpFir+1:end);
```

The model of the ground fault current is computed by GFMODEL2 (see Section D.4 on page 137) and both the model and the ground fault signal is windowed and stored in `mdlw` and `gfw` respectively.

Line 381 to 411 is a for-loop over the all nodes in the network and in line 388 and 389 the FFT of `mdlw` and `gfw` are computed. For each node the model is convolved with the network impulse response in line 390. Next the log-magnitude is computed and in line 393 a scale factor is computed another sub-function `getscale` (explained below). This scale factor compensates for the possible mismatch of the ground fault model. The model is computed in the time domain by a very simple algorithm and mismatch of a few dB is very likely to occur. In line 394 the scale factor is added to the (log-magnitude of the) convolved model and impulse response and the result is stored in the output matrices in the appropriate columns.

gferr.m

```
388      fftmdl = fft(mdlw);
389      fftgf  = fft(gfw);
390      fftci  = fftmdl.*dImpFft(:,nImp);
391      fftgfm = 20*log10(abs(fftgf(fIndx)));
392      fftcim  = 20*log10(abs(fftci(fIndx)));
393      dScale = getscale(fftgfm,fftcim);
394      fftcim = fftcim + dScale;
395      Fftgfm(:,nImp) = fftgfm(:);
396      Fftcim(:,nImp) = fftcim(:);
```

The scale factor is found simply by trying a range of factors and choosing the one which minimizes the difference. This is done in three steps in line 419–421 using the sub-function `getscalestep`.

gferr.m

```
413  %==================================================================
414  function dScale = getscale(fftgfm,fftcim)
415
416  m1 = fftgfm(1:10:end);
417  m2 = fftcim(1:10:end);
418  dScale = 0;
419  dScale = getscalestep(m1,m2,dScale,100);
420  dScale = getscalestep(m1,m2,dScale,10);
421  dScale = getscalestep(m1,m2,dScale,1);
422  %dScale = getscalestep(m1,m2,dScale,0.1);
423  %dScale = getscalestep(m1,m2,dScale,0.01);
424  %disp(['dScale = ' num2str(dScale)])
425
426  %==================================================================
427  function dStep = getscalestep(m1,m2,dScale,dSize)
428
429  l1 = [-dSize:dSize/10:dSize] + dScale;
430  Le = [repmat(m1-m2,1,length(l1)) - repmat(l1,size(m1,1),1)];
431  le = abs(sum(Le));
432  dStep = l1(find(min(le) == le));
```

D.3 Visualization of the Localization Algorithm Output

SHOWERR uses the data computed by GFERR and produces plots and images like those shown in Figure 5.10, 5.11, and 5.14 on page 61 and 64 respectively.

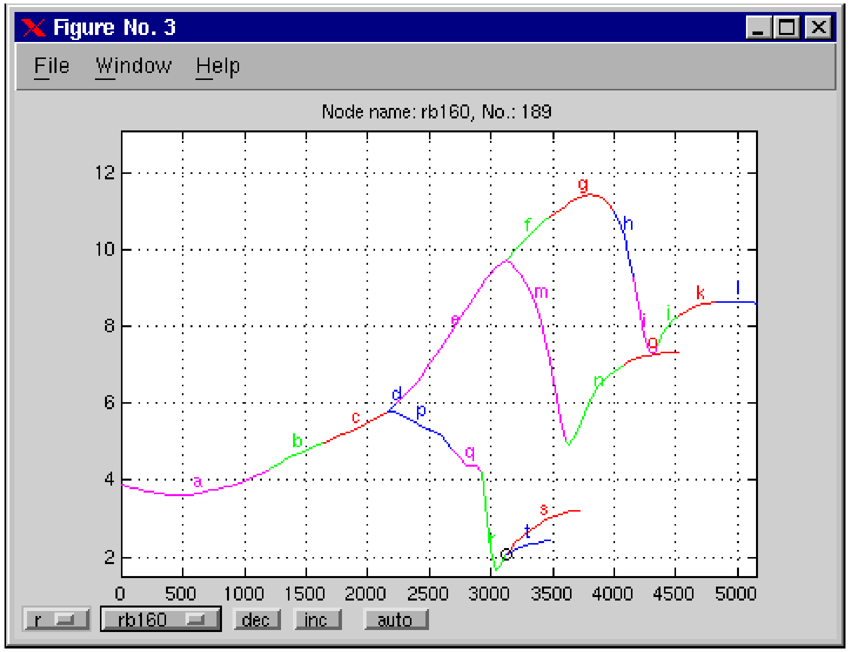


Figure D.1: Interactive plot by SHOWERR.

The SHOWERR figure window has five controls which gives access to the ground fault location. This means that the error measure can be observed as the ground fault is moved. The figure window with controls is shown in Figure D.1. The network section can be chosen by leftmost control in the lower left corner of the window. The nodes within that section is chosen by the second control from the left. The next two controls decreases and increases the fault location on the network by one step and the rightmost control gives an automatic step through all nodes in the network. This gives a very good impression of how the error measure changes as the fault moves through the network.

D.4 Ground Fault Model

GFMODEL2 computes a model of the current *in* the ground fault *at* the fault location and is an implementation of Equation 5.18 on page 55. The function is used by is used GFERR.

Input

GFMODEL2 takes a high-pass filtered ground transient as the first argument and the high-pass filter it self as the second argument.

Output

The return value is an estimate of the current transient at the fault location given by the above mentioned equation. The transient is high-pass filtered and delayed to make it consistent with the observed (input) transient.

Implementation

In line 15–17 the minimum and the maximum of the transient is found and in line 21 and 22 the index of these two value is found. In line 32–38 the step function in Equation 5.18 is composed with a delay so that the high-pass filtering and truncation in line 40 and 41 will place the transient approximately at the same time instance as in the input transient.

gfmodel2.m

```
1  function dMhp = gfmodel2(hpgf,hp)
2  %GFMODEL Computes a ground fault model.
3  %
4  %  dMhp = gfmodel2(hpgf,hp)
5
6  % Kaare Jean Jensen, 1998-06-22
7
8  iDim = isvector(hpgf);
9  if iDim == 0, error('hpgf must be a vector'), end
10
11 NFilter = length(hp);
12 NHpDelay = fix(NFilter/2);
13 NMdl     = length(hpgf) + NFilter - 1;
```

```

hpgf = hpgf - mean(hpgf);
dGfMin = min(hpgf);
dGfMax = max(hpgf);
if (dGfMin > 0) | (dGfMax < 0)
  disp('(gfmodel) hmm, dGfMin should be < 0 and dGfMax > 0')
end
iGfMin = find(hpgf == dGfMin);
iGfMax = find(hpgf == dGfMax);
if iGfMin < iGfMax
  i1 = iGfMin;
  i2  = iGfMax;
else
  i1 = iGfMax;
  i2  = iGfMin;
end
d1 = (hpgf(i1) - hpgf(i2))/2;
d2 = - d1;
NGfDelay = floor((i1+i2)/2);
m1 = NGfDelay + NHpDelay;
m2 = m1 + 1;
dM = zeros(NMdl,1);
dM(1:m1) = d1*ones(size(dM(1:m1)));
dM(m2:end) = d2*ones(size(dM(m2:end)));
if iDim == 1, dM = dM.'; end

dMhp = filter(hp,1,dM);
dMhp = dMhp(NFilter:end);
```

D.5 ATP File Name Extractor

The function GETAMF retrieves file names of impulse response or ground fault data in Matlab format. These file names must conform to a certain format as described in Appendix C.5 on page 119.

Input

A string containing the directory name to be searched is given as command line parameter to the function.

Output

If the directory contains N data files, the output is a N by 6 string matrix containing the six character file names (without extensions) as rows.

Implementation

The Matlab function WHAT is used in line 18 and 19 to find the MAT files in the given directory. In line 29–35 each file name is checked against the format convention. If the file name conforms with the format, line 38–41 inserts the name without extension in a list which finally is sorted alphabetically in line 45.

getamf.m

```
1  function strAtpMatFiles = getamf(strDir)
2  %GETAMF Get ATP MAT files
3  %  strAtpMatFiles = getamf(strDir) returns a matrix with the
4  %  ATP MAT files as rows from the directory strDir.
5  %
6  %  strAtpMatFiles = getamf uses current directory.
7
8  % Kaare Jean Jensen, 1998-06-29
9
10 nCh = 6; % Length of ATP MAT file names (without extension)
11
12 nMat = 0;
13
14 if nargin < 1, strDir = '.'; end
15 if exist(strDir,'dir') ~= 7
16   error(['The directory ' strDir ' does not exist'])
17 end
18 w = what(strDir);
19 sCell = w.mat; % get dir as cells
20 sm = zeros(length(sCell),nCh);
21 for n = 1:length(sCell)
22   % Initialize loop
23   sChar = char(sCell(n));
24   OK = 1;
25   % Tests:
26   if length(sChar) ~= nCh+4, OK = 0;
27   else
28     if strcmp(sChar(7:10),'.mat') ~= 1, OK = 0; end
29     if (sChar(1) ~= 'i') & (sChar(1) ~= 'g') OK = 0; end
30     for k = 2:3
31       if (sChar(k) < 'a') & (sChar(k) > 'z') OK = 0; end
32     end
```

```
    for k = 4:6
      if (sChar(k) < '0') & (sChar(k) > '9') OK = 0; end
    end
  end
  % Insert name if OK
  if OK == 1
    nMat = nMat + 1;
    sm(nMat,:) = sChar(1:6);
  end
end

if nMat > 0
  strAtpMatFiles = sortrows(char(sm(1:nMat,:)));
else
  strAtpMatFiles = [];
end
```

D.6 ATP Source Generation

The function `WRITESRC` writes a specified time function to a file which then can be used with an ATP simulation as user defined source.

Input

The first parameter is the time function and the second parameter is the file name.

Output

Output of the function is a disk file with the given name containing the time function in ATP format as a user defined source.

Implementation

ATP only accepts sources between one node and `TERRA`. This means that if a current source is to be inserted between two nodes, two sources with opposite signs must be connected between `TERRA` and each of the two

nodes. Therefore two columns are written into the file with opposite signs.

The function `REAL2STR` is used to convert the floating point number to a string obeying the fixed column structure of ATP input files. The if-statement in line 11–17 ensures that the two columns always has the same number of digits regardless of the sign.

writesrc.m

```
1  function writesrc(src,file)
2  %WRITESRC Writes user defined source to file
3  %   writesrc(src,file)
4
5  % Kaare Jean Jensen, 1998-08-26
6
7  fid = fopen(file,'wt');
8  if fid ~= -1
9    fprintf(fid,'/PLOT\n');
10   for n = 1:length(src)
11     if src(n) > 0
12       fprintf(fid,' %s%s\n',real2str(src(n),7),...
13                             real2str(-src(n),8));
14     else
15       fprintf(fid,'%s %s\n',real2str(src(n),8),...
16                             real2str(-src(n),7));
17     end
18   end
19   fprintf(fid,'    9999\n');
20   fclose(fid);
21 else
22   error([file ' can not be opened'])
23 end
```

D.7 ATP Cable Model Generator

The `MKCABLE` function generates Π-equivalent cable models from cable handbook data such as [NKT, 1992] for a number of specified frequencies. The output is used by makenet to generate full network models (see Appendix E.1 on page 164). `MKCABLE` writes files for ATP which then computes impedance and admittance matrices using the ATP supporting routine *CABLE CONSTANTS*.

Input

MKCABLE takes no command line parameters but has a list of frequencies and a section that must be edited to reflect the cable parameters. For each cable type, the geometric properties such as core cross section area and insulation thickness must be provided and can be found in the cable handbook [NKT, 1992].

Line 23–24 defines the list of frequencies for which impedance and admittance matrices will be computed.

mkcable.m

```
23 iAtpFreq = [50,   100,   200,   500,   1000,   2000,...
24               5000, 10000, 20000, 50000, 100000]';
```

Line 104–107 defines a string matrix strCabName with names of all cable types. These names must comply to the follow format: two letters representing the core material (Al/Cu), one space, two or three letter core cross sectional area, one space, and three letters representing the the cable type (APB/PEX/OHL). This format ensures that a unique four letter file name can be derived for an ATP input file. The first, the third, the fourth, and the third last letter are used as an ATP file name, so for an Al 240 PEX the name of the ATP input file is a24p.atp. Four different cable types are listed here with various cross sections: aluminum and copper APB cables, aluminum PEX cables, and copper overhead lines[1].

mkcable.m

```
101 % -------------------- User supplied data -------------------------
102 % Character number 1, 4, 5 and length(strCabName(n,:))-2 will
103 % be used for the ATP file name so they must be non-blank.
104 strCabName = str2mat('Al 240 APB','Al 150 APB','Cu 150 APB',...
105                      'Al 95 APB', 'Cu 95 APB', 'Cu 50 APB',...
106                      'Al 240 PEX','Al 150 PEX','Al 95 PEX',...
107                      'Cu 50 OHL', 'Cu 35 OHL', 'Cu 25 OHL');
```

Next, each of these four cable types are defined in the following four matrices in terms of their geometrical dimensions. Each cross section

[1] The ATP routine *CABLE CONSTANTS* provides support for both underground cables and overhead lines. For simplicity all line types including overhead lines will be referred to as *cables*.

listed in strCabName must be found in these matrices. Aluminum APB cables are defined in line 110–115, copper APB cables in line 118–123, aluminum PEX cables in line 126–132, and overhead lines in line 135–145. All these geometric parameters are taken directly from [NKT, 1992]. Different geometrical properties are used for the different cable types as given by the cable handbook, e.g. the core-to-core distance is listed for an aluminum APB cable whereas it is the insulation thickness for a aluminum PEX cable. MKCABLE later computes geometrical properties consistent with the ATP input format. This way new cable cross sections can easily be added.

mkcable.m

```
109 % Al APB geometrical dimensions
110 AlApbGeoDim = [240,  150,  95;... % Cross section
111                5.2,  5.2, 5.2;... % Core-core distance
112                  4,    4,   4;... % Core-pipe distance
113                2.0,  1.9, 1.7;... % Pipe thickness
114               52.5,   45,  39;... % Outer pipe diameter
115                0.1,  0.1, 0.1];   % Pipe insulation thickness
116
117 % Cu APB geometrical dimensions
118 CuApbGeoDim = [150,  95,  50;... % Cross section
119                5.2, 5.2, 5.2;... % Core-core distance
120                  4,   4,   4;... % Core-pipe distance
121                1.9, 1.7, 1.6;... % Pipe thickness
122                 45,  39,  33;... % Outer pipe diameter
123                0.1, 0.1, 0.1];   % Pipe insulation thickness
124
125 % Al PEX geometrical dimensions
126 AlPexGeoDim = [240,  150,   95;... % Cross section
127               17.3, 13.5, 10.8;... % Core diameter
128                3.4,  3.4,  3.4;... % Thickness of PEX insulation
129               26.9, 23.1, 20.4;... % Total core diam. (core+PEX+2*semi-cond.)
130                 35,   25,   25;... % Pipe cross section
131               61.5, 53.3, 47.5;... % Outer pipe diameter
132                3.1,  2.8,  2.6];   % Pipe insulation thickness
133
134 % Cu OHL (overhead line) geometrical dimensions
135 CuOhlGeoDim = [50,   35,   25;... % Phase wire cross section
136                25,   25,   25;... % Ground wire cross section
137              8.05, 8.05, 8.05;... % Vertical   distance to 1st phase
138              9.27, 9.27, 9.27;... % Vertical   distance to 2nd phase
139              8.05, 8.05, 8.05;... % Vertical   distance to 3rd phase
140              6.95, 6.95, 6.95;... % Vertical   distance to ground wire
141              0.75, 0.75, 0.75;... % Horizontal distance to 1st phase
142                 0,    0,    0;... % Horizontal distance to 2nd phase
143             -0.75,-0.75,-0.75;... % Horizontal distance to 3rd phase
```

```
144                0,    0,    0;... % Horizontal distance to ground wire
145              1.2,  1.2,  1.2];   % Lowering at midspan
```

In line 147–157 the intrinsic resistance (ρ) and permeability (μ) for aluminum, copper, and iron are listed. Material parameters for the pipe and the insulation of APB and PEX cables are listed in line 160–165 and 168–175 respectively.

mkcable.m

```
147 % Material parameters for copper
148 CuMatPar = [1.786E-8,...  % core rho
149             0.999991];    % core my_r
150 
151 % Material parameters for aluminum
152 AlMatPar = [2.38E-8,...   % core rho
153             1.00002];     % core my_r
154 
155 % Material parameters for iron (only used for overhead line ground wire)
156 FeMatPar = [9.78E-8,...   % core rho
157             1.0];         % core my_r
158 
159 % Material parameters for APB cables
160 ApbMatPar = [2.2E-7,...    % pipe rho
161              0.999983, ... % pipe my_r
162              3.5,...       % pipe epsilon1 (insulator inside pipe)
163              4.0,...       % pipe epsilon2 (insulator surrounding pipe)
164              1,...         % core insulator my
165              3.5];         % core insulator epsilon
166 
167 % Material parameters for PEX cables
168 PexMatPar = [CuMatPar(1),... % pipe rho  (Cu)
169              CuMatPar(2),... % pipe my_r (Cu)
170              1,...           % pipe epsilon1 (insulator inside pipe)
171              2.3,...         % pipe epsilon2 (insulator surrounding pipe)
172              1,...           % rho  for semi conducting material
173              1,...           % my_r for semi conducting material
174              1,...           % core insulator 2 (PEX) my_r
175              2.5];           % core insulator 2 (PEX) epsilon
```

Output

MKCABLE generates impedance and admittance matrices (using ATP) for the specified list of cables and for the given range of frequencies. In addition to this, a number of tables in TEX format are written with information on the input and output data. Also some modal parameters like characteristic impedance and propagation constant are computed for each frequency and listed in the tables.

The frequency specific data like impedance/admittance matrices and modal parameters are written to a directory for that particular frequency. The general data like geometrical dimensions are written to current directory. The directory name for the frequency specific data is an `f` followed by a six digit representation of the (integer) frequency. A list of frequencies like 50 Hz, 2 kHz, and 100 kHz will result in data written to the directories `./f000050`, `./f002000`, and `./f100000`. As mentioned above, the ATP output files will have a four letter names derives from the cable type, so that the ATP output file for a Cu 95 APB will be called `c95a.lis`, an Al 150 PEX will be called `a15p.lis`, and a Cu 35 OHL will be called `c35o.lis`. Remember that ATP input are ATP files and output are LIS files. The phase and modal parameters are written to two files called `cabpar1.tex` and `cabpar2.tex` and could look the examples in Table D.1 and D.2. These parameters are computed at 1 kHz.

The TEX files written to current directory are tables with cable dimensions. The following list provides an overview of the names and contents.

- `cdvar.tex`: description of MKCABLE and ATP variable names.
- `cdapbgeo.tex`: geometrical dimensions for APB cables.
- `cdapbatp.tex`: ATP dimensions for APB cables.
- `cdpexgeo.tex`: geometrical dimensions for PEX cables.
- `cdpexatp.tex`: ATP dimensions for PEX cables.
- `cdohlgeo.tex`: geometrical dimensions for overhead lines.
- `cdohlatp.tex`: ATP dimensions for overhead lines.
- `mkcable.tex`: main TEX file which includes all other TEX files. Run this files through LaTeX to get an overview of all parameters for all cables.

cable type	r_{pos} Ω/km	l_{pos} mH/km	c_{pos} μF/km	r_{zero} Ω/km	l_{zero} mH/km	c_{zero} μF/km
Al 240 APB	0.3107	0.1572	0.3768	2.13	0.3265	0.1467
Al 150 APB	0.3787	0.1812	0.3324	2.616	0.3611	0.1373
Cu 150 APB	0.3386	0.1768	0.3324	2.576	0.3567	0.1373
Al 95 APB	0.4601	0.2062	0.2932	3.365	0.4016	0.1281
Cu 95 APB	0.4072	0.2023	0.2932	3.312	0.3977	0.1281
Cu 50 APB	0.5512	0.2396	0.2452	4.185	0.4623	0.1154
Al 240 PEX	0.2444	0.1738	0.08	1.733	0.3397	0.03707
Al 150 PEX	0.3157	0.2031	0.06951	2.405	0.3813	0.03422
Al 95 PEX	0.3978	0.2312	0.06137	2.484	0.412	0.03182
Cu 50 OHL	0.4925	1.169	0.00964	3.792	3.228	0.00486
Cu 35 OHL	0.6276	1.208	0.00935	3.927	3.266	0.00478
Cu 25 OHL	0.8169	1.242	0.00909	4.117	3.301	0.00472

Table D.1: Cable parameters computed by ATP at 1 KHz and a ground resistance of 940 Ω.

cable type	Z_0	γ ($10^{-6}\,\mathrm{m}^{-1}$)	v (10^8 m/s)
Al 240 APB	39.21 − j33.46	3.96 + j4.641	0.68
Al 150 APB	45.89 − j39.51	4.13 + j4.793	0.66
Cu 150 APB	43.69 − j37.11	3.88 + j4.562	0.69
Al 95 APB	53.60 − j46.58	4.29 + j4.938	0.64
Cu 95 APB	50.81 − j43.50	4.01 + j4.681	0.67
Cu 50 APB	64.03 − j55.88	4.30 + j4.932	0.64
Al 240 PEX	77.90 − j62.42	1.57 + j1.958	1.60
Al 150 PEX	94.00 − j76.90	1.68 + j2.053	1.53
Al 95 PEX	111.2 − j92.75	1.79 + j2.144	1.47
Cu 50 OHL	402.6 − j202.0	0.61 + j1.219	2.58
Cu 35 OHL	435.1 − j245.4	0.72 + j1.279	2.46
Cu 25 OHL	476.1 − j300.2	0.86 + j1.361	2.31

Table D.2: Cable parameters processed by Matlab at 1 KHz and a ground resistance of 940 Ω.

Implementation

First MKCABLE writes input files for ATP for each cable and each frequency, and a batch script for ATP to run all these ATP input files. This is accomplished by the call to the sub-function `writeatpcableinput` in line 36–37. Next this script is executed using the Matlab exclamation mark command, which then runs ATP on all the written input files. The output is redirected to a file with the same name as the batch script but with extension `.out` for reference in case of problems. These lines require Matlab to run on a Unix system.

mkcable.m

```
35  % Write ATP input files for all cables and all frequencies
36  [strCabName,strAtpName,RGnd,strDir,strFileNameName] = ...
37      writeatpcableinput(fidTexMain,strBatchName,iAtpFreq);
38
39  % Run ATP
40  disp('  running ATP')
41  disp(['  output is written to ' strBatchName '.out ...'])
42  eval(['!chmod 700 ' strBatchName])
43  eval(['!' strBatchName ' > ' strBatchName '.out'])
```

For each frequency, the ATP output files are then read by the sub-function `computezg` in line 54–55 the sequence and modal parameters are computed and written to the TEX files.

mkcable.m

```
51  for n = 1:NFreq
52    disp(['  computing Z0 and gamma for ',...
53          fullfile(strDir(n,:),strFileNameName)])
54    [rPos(n,:),lPos(n,:),cPos(n,:),rZero(n,:),lZero(n,:),cZero(n,:)] = ...
55        computezg(fullfile(strDir(n,:),...
56                  strFileNameName),...
57                  fidTexMain,...
58                  strCabName,...
59                  strAtpName,...
60                  iAtpFreq(n),...
61                  RGnd);
62  end
```

As the last action, the sequence and modal parameters for all frequencies are saved to the file name in `strPosVar` and LaTeX is run on the TEX files to generate a document with all computed data.

mkcable.m

```
eval(['save ' strPosVar ' rPos lPos cPos rZero lZero cZero freq strCabName'])

fprintf(fidTexMain, '\\end{document}');
fclose(fidTexMain);
[strDummy,strTexName] = fileparts(strTexMain);
eval(['!latex ' strTexName])
disp('')
disp(['  to show tables, run:'])
disp(['"!xdvi  ' strTexName ' &"'])
disp('')
disp(['  positive and zero sequence r, l, and c are saved to ',...
      strPosVar '.mat'])
disp(['  use SHOWPOSV to plot r, l, and c as a function of frequency'])
```

D.8 The DISMO-Toolbox

The DISMO-toolbox is an analysis tool for power system signals. The toolbox has three main functionalities: estimation of the fundamental frequency component, detection of transients, and wavelet analysis.

The fundamental estimation algorithm splits the signal into a fundamental frequency component and a residual component. The fundamental component is then used to estimate the instantaneous frequency and a representation of the signal in the complex s-plane. The residual component can be used to estimate the spectral content of the signal without interference from the very large fundamental component.

The transient detection uses the fundamental component to find transients in the signal such as spikes, changes in the load, etc.

The wavelet analysis uses the residual signal to make a time-frequency analysis.

The underlying algorithms have all been developed during four Masters projects. The fundamental and residual estimation algorithms were developed in [Høg, 1994], the transient detection algorithm in [Jespersen, 1994], and the wavelet analysis in [Nielsen, 1995] and [Madsen, 1996]. These algorithms are integrated into a single tool by the DISMO-toolbox.

The toolbox is in a development phase and it is the intension that new features developed in the DISMO project should be implemented

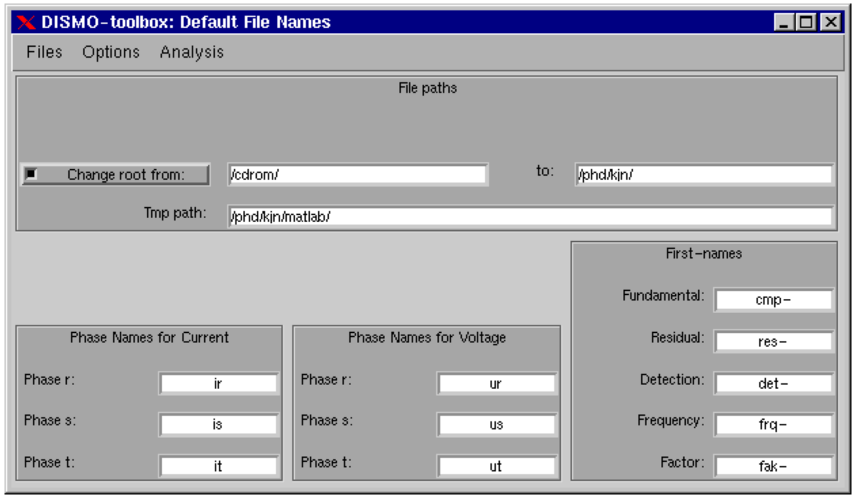


Figure D.2: The DISMO-toolbox window in the Default File Names view.

in the toolbox. This way the toolbox becomes a central demonstration environment for the DISMO project.

The work on this toolbox was done before version 5 of Matlab became common so the code is written with Matlab 4 syntax. Porting the toolbox to version 5 could, however, be a benefit as the possibility of collecting different variable types into a cell array and sub-functions makes the design simpler.

The DISMO-toolbox operates on disk files in Matlab 4 format which means that input is read from disk and output data is written to one or more disk files. One exception is the wavelet analysis which only writes the graphical output to EPS files.

D.8.1 The Main Menu

The main window has three menu items: Files, Options, and Analysis.

Under the Files menu the menu item Default file names is found. This item opens the window shown in Figure D.2 which gives access to certain file name properties. The first field provides the possibility of keeping the directory structure of the input files but changing the root directory. This is handy when the input data is read from a CD where output files can not be written. In addition default names for the three phase voltages and currents and default prefixes can be chosen for the fundamental component (`cmp-`), the residual component (`res-`), the transient detection (`det-`), the frequency estimate (`frq-`), and the conversion factors for the input files (`fak-`). The default file names are convenient when large amounts of data are processed.

The Options menu has an option to save all parameters at exit, to restore default parameters, and a function to validate the toolbox. This validation function uses a known data set to evaluate both the fundamental estimation and the transient detection algorithms to ensure at all times during the development process that everything is working properly.

The Analysis menu has three main items each of which opens a window for a specific task: fundamental estimation, transient detection, and wavelet analysis.

D.8.2 The Analysis Menu

Common for these three windows is that they have editable fields at the top for input file names and at the bottom for output file names. The input fields have a Browse button for convenience.

A description of input and output file formats is found in the respective documentation ([Høg, 1994] and [Jespersen, 1994]).

The Fundamental Estimation Item

The Fundamental estimation window is shown in Figure D.5. The input signal may be any voltage or current signal in Matlab 4 file format. The file name must be specified in top field of the window.

The first sample and the number of samples that will be processed

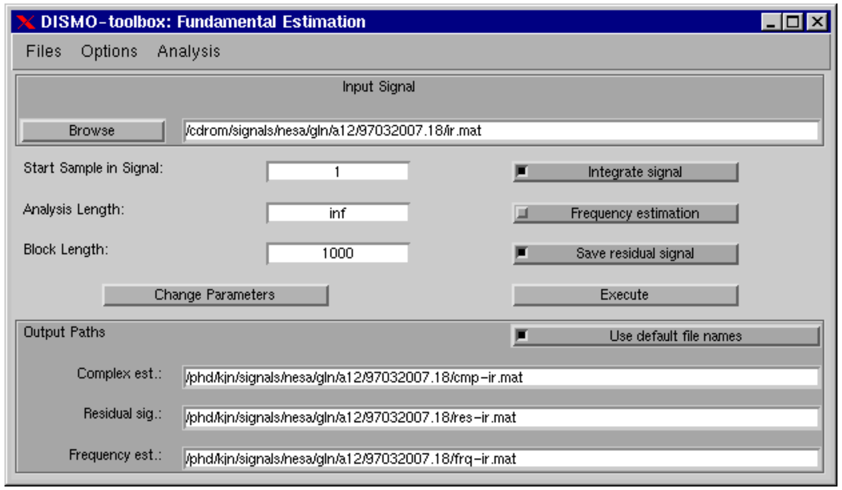


Figure D.3: The DISMO-toolbox window in the Fundamental Estimation view.

is specified in the Start sample in signal and Analysis length fields. If the characters `inf` are specified as analysis length the signal is processed to the end.

A number of parameters for the fundamental estimation algorithm can be changed through the parameter window in Figure D.4. This window is opened by pressing the Change parameters button. A description of these parameters can be found in [Høg, 1994].

If the signal to be processed is a current signal from a Rogowski coil, the signal needs to be integrated before processing. This can be specified in the Integrate signal check box. The integrator is treated in Appendix A.2 on page 100.

If the Frequency estimation check box is checked the instantaneous frequency is estimated and saved to the file name gives in the respective field at the bottom of the window. If the box is unchecked the power frequency is assumed to be 50 Hz and no frequency estimation is saved.

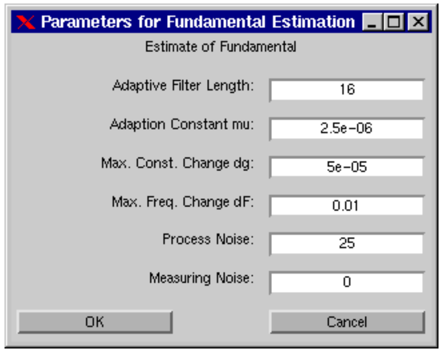


Figure D.4: Window for Fundamental Estimation parameters.

If the Save residual signal check box is checked the residual signal is saved to the file name given in the respective field at the bottom of the window.

The Execute button starts the estimation algorithm and will produce the complex estimate of the fundamental component, the instantaneous frequency if the Frequency estimation box is checked, and the residual signal if the Save residual signal box is checked. If the Use default file names box is checked the output files for these three signals are found automatically by prefixing the input file name with the three prefixes `-cmp`, `-frq`, and `-res` as given by the Default file names menu item.

The input signal is assumed to have a sampling frequency of 20 kHz and both the complex and the frequency estimate will be down-sampled by a factor 75.

The Transient Detection Item

The Transient detection window is shown in Figure D.5. The input signal for the transient detection is specified in the top field of the window and must be the complex estimate (a `cmp-` file) computed by the fundamental estimation algorithm. If the Default file name check box is checked the

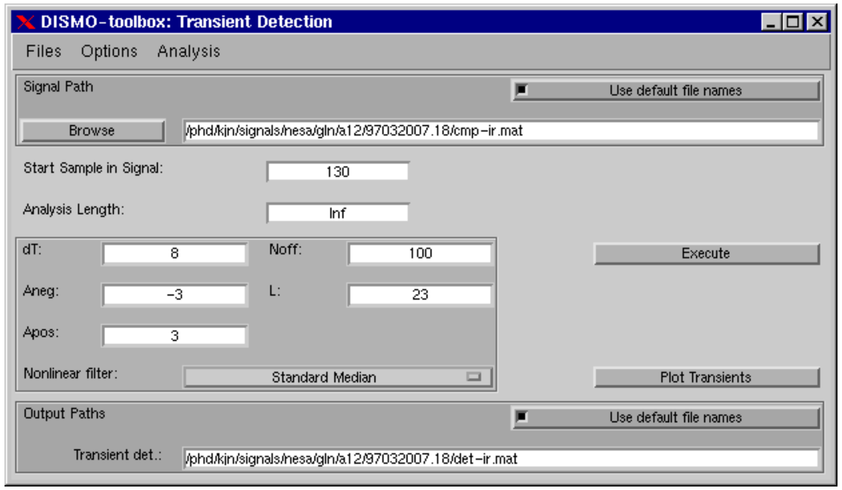


Figure D.5: The DISMO-toolbox window in the Transient Detection view.

input file is the output file from the Complex estimation output field.

This window also has two fields for the start sample and the analysis length. Note that the sampling frequency has been reduced by a factor 75 by the fundamental estimation algorithm. The parameters in the frame below these fields control the behavior of the detection algorithm and are explained in [Jespersen, 1994].

Besides the Execute button which performs the detection, a Plot transients button will subsequently plot all detected transients in `EPS` format. A `TEX` file which includes all the written `EPS` figures is written for easy viewing of the transients.

The Wavelet Analysis Item

The Wavelet analysis window is shown in Figure D.6. The input file is specified in the field at the top of the window. This file must be the

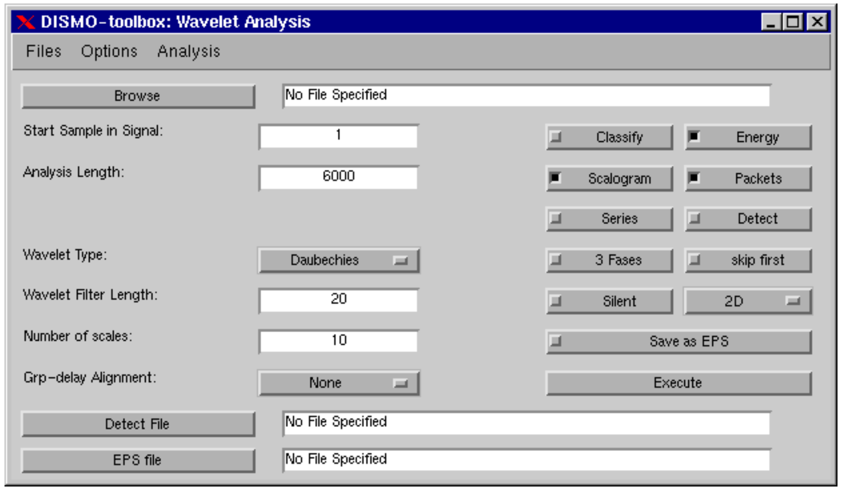


Figure D.6: The DISMO-toolbox window in the Wavelet Analysis view.

residual signal computed by the fundamental estimation algorithm (a `-res` file). If the fundamental component is not removes from the input signal nothing else that this component will be seen in the wavelet transform.

A number of the basic properties such as the mother wavelet, the filter length, and the number of scales. A detailed discussion of these properties can be found in [Vetterli and Kovačević, 1995]. Besides these basic properties a number of other options are available. This includes wavelet packets and several plotting options. At the bottom a field is available to specify an output file from the Transient detection algorithm (a `-det` file). If the Detect option is on, the detected transients are marked with vertical lines in a plot of the fundamental magnitude.

Integration of the wavelet analysis window in the DISMO-toolbox is not completed. A Default file names feature would be convenient in this window as well. Furthermore it should be possible to specify a transient number in the `-det` file to be analyzed instead of the start sample.

D.8.3 Implementation

An outline of the graphical user interface implementation of the DISMO-toolbox is discussed in this section. The reader is a assumed to have knowledge of the Matlab `UICONTROL` function which is the basic user interface element that creates push buttons, check boxes, drop-down lists, editable fields, etc.

Outline

The `DISMOTB` initializes the DISMO-toolbox. Here the menu for the main window is set up by the `MENUS` function and controls for each window is initialized by a call to the respective function (see next section) with a request to initialize.

Handles for the controls are stored in a vector as `userdata` for the respective menu item. The control handles are stored and retrieved with the functions `SETCTRLH` and `GETCTRLH`.

The menus and the main window are identified by their title and for convenience each title is associated with an alias. This is administered by the `STRALIAS` function. This means that it will not be possible to run multiple instances of the toolbox in the current implementation. Another constrain that this implementation imposes is that two menu item aliases can not have the same title.

All controls have their `'visible'` property set to `'off'`. Only the controls corresponding to the selected menu item is visible. This is controlled from the main menu via the function `HIDECTRL` and a call to the respective menu item function with the action string `'show'`. The controls for each menu item is discussed in the next section.

Setting Up Controls for a Menu Item

The Matlab Graphical User Guide recommends that functions which set up controls for windows is organized in a way that the code that initializes the controls is kept together with the code that implements the action of the controls in one single `M` file. This is done through an *action* parameter to the function which defines the action to be taken with each call of the function. The function is then build around one large `if`/`elseif`/`else` statement, where a string comparison is performed on the action string. This means that the control for the execute button in the transient detection window has a callback string that looks like: `'mtrandet(''Execute'')'`, where `MTRANDET` is the function that sets up the transient detection window. `MTRANDET` then contains an `elseif` statement that evaluates to true if the action string is equal to `Execute`.

Definitions that must be available to all actions can be placed before the `if`/`elseif`/`else` statement as the function is run every time a control is activated. This way no global variables are needed and the window is independent of the variables in the Matlab work space.

All functions that set up controls for the menu items is named after the menu item preceded with an '`m`' for menu. This makes it easier to find these functions among the large number of files contained in the toolbox. Following list gives the names of these files and the corresponding menu item.

file name	menu item name
`mdeffile.m`	Default File Names
`msetdefs.m`	Set Default Parameters
`msavepar.m`	Save All Parameters at Exit
`mvalidtb.m`	Validate Toolbox
`mfundest.m`	Fundamental Estimation
`mtrandet.m`	Transient Detection
`mwavelet.m`	Wavelet Analysis
`mcombine.m`	Combined analysis

If the menu item has user defined parameters these are saved to `MAT` file named after the `M` file but without the preceding `m`, e.g. the Fundamental Estimation menu saves parameters to `fundest.mat`.

Adding a New Menu Item

To add a new menu item the list below summarizes the changes that needs to be made in order to make the new menu item compliant with the toolbox. These steps are also listed in the file `readme.txt` contained in the toolbox.

1. Add a unique alias (e.g. `mymenu`) and corresponding menu item text to the file `stralias.m`. A line would look something like:

   ```
   elseif strcmp(alias, 'mymenu') label = 'My New Menu Item';
   ```

   It is a good idea to keep the alias seven characters long so that a function can be named after the alias preceded with an `m` (for menu) and still comply with the 8.3 file format. This makes the code more flexible in terms platform, storage on floppy, etc.

2. Create a `M`-function (e.g. called `mmymenu.m`) that defines the uicontrols and functions for the menu item. The file `mdefault.m` contains the basic elements and can be used as a template.

3. Add an uimenu to `menus.m`. This could something like:

   ```
   uimenu(fadmenu,...
          'label', stralias('mymenu'),...
          'callback',['hide, mmymenu(''show'')']);
   ```

4. Add a line to the initializing and exiting section of `dismo.m`. The two lines would look like:

   ```
   mmymenu('init')
   ```

   and

   ```
   mmymenu('exit')
   ```

5. If the menu item contains default parameters following lines must be added to the exit action of `mmymenu.m`:

```
if strcmp(msavepar('getstatus'),'on')
  message('(mmymenu) saving parameters','debug')
  TmpPath    = mdeffile('gettmppath');
  hcMyMenu = getctrlh('mymenu');
  ParamToBeSaved  = str2num(get(hcMyMenu(nParamA),'string'));
  AnotherParam    = str2num(get(hcMyMenu(nParamB),'string'));
  eval(['save ' TmpPath ' ... mymenu ParamToBeSaved AnotherParam ... '])
end
```

following lines must be added to `savepars.m` and the following lines must be added to `movepars.m`:

```
clear
message('(movepars) moving mymenu.mat','debug')
eval(['load   ' mdeffile('getoldtmppath') '/mymenu'])
eval(['delete ' mdeffile('getoldtmppath') '/mymenu.mat'])
eval(['save   ' mdeffile('gettmppath')    '/mymenu'])
```

The DISMO-toolbox prints messages to the Matlab command window via the `MESSAGE` function. This function issues messages in four different levels: information level, warning level, error level, and debug level. The line in `message.m` that prints out debug messages is normally commented out, but if problems arises this comment character can be removed and debug messages will be printed to the Matlab command window.

D.9 Experiment Data Extractor

The `EXTDATA` function was used to extract a down-sampled time-truncated version of the very large data material acquired during the ground fault experiments described in Chapter 6. The purpose of this data reduction was to make a version of the data which fitted no more than one CD, providing an overview of the full data material. A Matlab data browser for this data is described in Section D.10.

`EXTDATA` uses four C++ utilities: the three data conversion programs dat2mat, b2mat, and w2mat, and the down-sampling utility downsmpl. These programs are called from the Matlab function using the exclamation operator.

The purpose of using one single Matlab function to perform the data conversion and extraction was to make it easier to track errors and to

make it visible for others what actions were performed on the extracted data. `EXTDATA` writes terminal output telling what actions are performed and this output can be redirected to a file. The CD with the extracted data contains a directory called `/out` with this output from each converted CD.

`EXTDATA` takes two parameters: an input path and an output path. The input path is searched recursively looking for data acquisition files of `DAT` or `V01` type (DISMO-PC or TRA800 data formats). If none of these file formats are found, BAKKER format is assumed. The input directory structure is reproduced under the output path.

An attempt is made to fix few of the errors on the acquisition data described in Section F.1.3 on page 175. This include the erroneous rotation of the channels made by the DISMO-PC.

Each signal is saved to its own file in binary Matlab version 4 format and the files are given descriptive names like `ir` for the current in phase R and `u0p` for the zero system (Petersen coil) voltage acquired using a voltage probe. A full list of these names are given in Table F.2 on page 173.

D.10 Experiment Data Browser

The ground fault experiment described in Chapter 6 produced a very large amount of data. To provide easy access to this data the Matlab function `KYVDATA` has been written (the 10 kV laboratory was located at Kyndby which has the acronym KYV at NESA). The user interface is similar to the DISMO-toolbox except that `KYVDATA` only has one single window which is shown in Figure D.7. The most current version of `KYVDATA` at the time of writing is version 1.6 which is the version documented in this section.

D.10.1 Overview

The original data has been down-sampled and the transient part of the signals have been extracted and written to one single CD (the original data is stored on no less than 21 CD's). The original directory structure

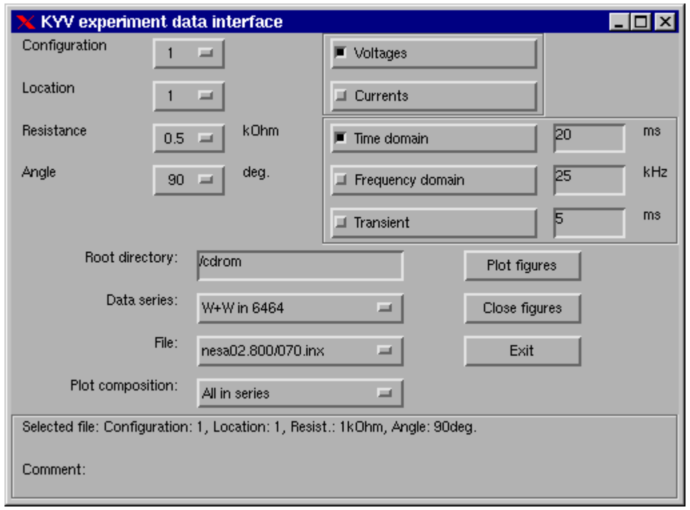


Figure D.7: The Kyndby experiment data interface window.

has been retained in order to avoid confusion. Instead of giving the directories meaningful names a large array of structures are included with the preprocessed data. This array links properties like fault location, fault resistance, acquisition point, etc. with the numbered directory structure generated by the acquisition systems. This array is returned by the Matlab function `KYVNAME` which is contained on the CD with the down-sampled data.

The data is stored as the voltage output of the different sensors. In order to convert this sensor voltage to the MV voltage and current different conversion factors must be applied to the signals. These conversion factors vary among the different signals. To avoid confusion and to be able to track errors these conversion factors are implemented in `KYVFAC`. This function returns the conversion factor for a given file name.

`KYVDATA` uses `KYVNAME` to retrieve the file names for a given experiment

setup and `KYVFAC` to get the conversion factor for the data. `KYVNAME` and `KYVFAC` could, however, be used independently of `KYVDATA`.

D.10.2 User Interface

Four main properties of the data can be chosen in the upper left corner of the main window:

Configuration: two different configuration were used, 1 and 2, which is shown in Figure 6.3 and 6.4 on page 72.

Location: six different ground fault locations were used. These locations are numbered from 1–3 and 5–7. See Figure 6.1 on page 70 for reference.

Resistance: six different ground fault resistances were used – 0, 0.5, 1, 2, 10, and 20 kΩ. The 10 kΩ resistance were only used at location 1.

Angle: six different closing angles were used – 0, 30, 60, 90, 120, 150, and 180°. The 180° angle were only used at location 1.

The root directory of the data (CD) must be specified in the Root directory field. This is could be `d:` on a Windows system or a mount directory on a Unix system.

The acquisition point can be chosen from the Data series drop-down list. Following possibilities are available:

W+W in 6464: Broad band three phase voltage and current acquired on voltage probes and Rogowski coils with a TRA800 transient recorder.

W+W in 5900: Three phase voltage and current acquired on measurement transformers with a TRA800 transient recorder.

BAKKER in 5900: Voltage across and current through the Petersen coil acquired with a BAKKER transient recorder.

DISMO in 6464: Broad band three phase voltage and current acquired on combi-sensor with the DISMO-PC.

If more than one signal were acquired for a given situation the different files can be found in the File drop-down list.

Instead of plotting all three phases from one series, one phase can be plotted from all series. This way a comparison can be made of e.g. the current from the measurement transformer and the Rogowski coil. Following possibilities are available:

All in series: all signals will be plotted from the given data series.

Zero system: the zero system will be plotted from all data series. For those data series where an acquisition of the zero system was not made, `KYVDATA` will compute the zero system as the sum of the three phases.

Phase R/S/T: the given phase R/S/T from all data series will be plotted.

The five check boxes in the upper right corner control which signals that will be plotted. The possibilities are: Voltage and/or Current and Time domain and/or Frequency domain and/or Transient. The transient of the signal is extracted by `KYVDATA` with a high-pass FIR filter applied to the time domain signal. The filter cut-off frequency is 2 kHz and the Matlab function `FIR1` is used to create the filter.

Appendix E

C++ Utility Programs

This appendix describes the C++ programs which are used to generate the network models (makenet), to process the data from ATP (atp2mat and downsmpl), to convert acquisition data (dat2mat, b2mat, and w2mat), and a program which is used in connection with the DISMO-PC as a timer for batch acquisitions (timer).

All programs except for timer are compiled on a Linux machine with an accompanying `Makefile` by running the program make. The program timer is special since it uses functions from `dos.h` to execute a child process. This program will not easily port to a Unix platform. The other programs, however, ports to a DOS/WINDOWS with very few modifications.

Following list serves as a table of contents for this appendix:

E.1 Network Model Generation

This program is called makenet and generates a full network model of Π-sections for ATP simulations. The input file syntax for makenet is described in Appendix C.6 on page 119.

If makenet is executed with option -h following output is printed to the terminal.

Terminal output

```
MakeNet version 3.5
  usage: makenet [-dhks] NetDefFile
    -d   Print debug information
    -h   Display this help
    -k   List valid keywords
    -s   Run silent
```

The -d option makes makenet print messages of all actions which can be useful for trouble-shooting. Option -s make makenet run silently.

If makenet is executed with option -k all valid keywords for the input file are listed. These keywords are described in details in Appendix C.6 on page 119.

Terminal output

```
MakeNet version 3.5
Following keywords are valid:
  setgroundresistance [resistance in ohms/m] (default: 0.025)
  setpilength [length of Pi-equivalent in meters] (default: 10)
  setcabledir [dir] (must precede definecable keyword)
  definecable [file] [cable type] [node name, 4ch]
  newcablesection [node name, 2ch] [length] [cable type]
  addsplit (no parameters)
  usesplit [node name, 2ch]
  load [kW] [cos(phi)]
  externnode [node name, 5ch]
- Note: Xch following node names means that the mandatory length of the
        node name is X characters.
```


E.2 Data Conversion

E.2.1 ATP Output Conversion

This program is called atp2mat and it converts the LIS file output from ATP to a Matlab MAT file. ATP can be instructed to write all output to a LIS. This output includes a descriptive interpretation of the ATP input file. The LIS file is searched by atp2mat for certain keywords which mark the beginning of the numerical data. This data is then written in Matlab MAT format to a file with the same name as the LIS file.

If atp2mat is executed with option -h following output is printed to the terminal.

Terminal output

```
atp2mat, version 1.1
  Converts ATP text output to binary Matlab file
  usage: atp2mat [-dhmos] atpout.lis
    -d       Print debug information
    -h       Display this help
    -m       Write multiple files (one file per column)
    -o File  Write output to file File
    -s       Run silent
```

The -d option will make atp2mat print messages for all actions. This is useful for trouble-shooting. The -o File option will instruct atp2mat to write output to File and the -s option will make atp2mat run silently.

The -m option can be used if the ATP file contains more than one signal. This will make atp2mat write separate MAT files for each signal. Otherwise all signals will be written to one single matrix.

E.2.2 Acquisition Data Conversion

This section describes the three acquisition data conversion programs dat2mat, b2mat, and w2mat. These conversion programs have very similar behavior and accepts almost the same set of options. The output format for all these programs are the binary Matlab version 4 format in 16 bit integer precision.

The acquisition program on the DISMO-PC is GlobalLab which write DAT files. These files are converted to Matlab format by the dat2mat. The -h option prints following output to the terminal.

Terminal output

```
dat2mat, version 1.4
  Converts binary output from GlobalLab (DAT) to MAT files
  Usage: dat2mat [-cdhmprs] [-o matdir] infile
  Options:
    -c          do not write data, Check only
    -d          print Debug information
    -h          display this Help
    -m          oMit directory infile in output
    -o matdir   write Output to directory matdir/infile
    -p          Print dat header
    -r          Recurse through sub directories
    -s          run Silent
  Description:
  - If infile is a  directory and option r is on,  it will be
    searched recursively
  - If a TXT file is found with the same name as the DAT file
    it will be  searched  for a time and a date string.  This
    will be used to compose a directory name.  The format for
    this file is:
    ...
    TIME=16:47
    DATE=981130
    ...
```

The -c makes dat2mat convert the input file checking for errors and overflow but no output files are written. The -d option will print detailed messages of actions which can be used to trouble-shoot and the -s will make dat2mat run silently. The option -o matdir will write output to directory matdir. To make dat2mat recurse through subdirectories looking for DAT files the -r option can be specified.

The DAT files from the DISMO-PC contain six channels in each file. These six signals are assumed to be a three phase voltage and current acquisition with the voltage signals first so the signals will be called ur, us, ut, ir, is, and it all with extension mat. The program timer (see Appendix E.4 on page 168) is able to write the time to a TXT file and if this TXT is has the same name as the DAT file, the time and date in the TXT will be used as a directory name for the output. If no TXT file is found

the DAT file name will be used as a directory name for the output files. The -m option suppresses this action.

The Bakker transient recorder also writes one single file with all channels like the DISMO-PC. The conversion program for these files is called b2mat. This program accepts the same options as dat2mat. The only two differences are that b2mat does not accept the TXT file with date and time specification for output directory naming and b2mat uses file names like ch_n.mat for channel n. The Bakker transient recorder is able to scale and offset the input signal within a certain range. Information on these settings are saved to the acquisition file together with the sampling rate. This information is read by b2mat and written to the output MAT file. Four variables are therefore written to the MAT file: scale, offset, fs, and x. The signal is called x and the other variables are self explanatory.

The Tra800 transient recorder writes each channel to a separate file. The files have numbered names like 0000n001.v01 for the signal on channel n. This name is used as output name only with the MAT extension instead of the V01. The Tra800 also uses a scale and offset parameter which is stored to the data acquisition file and like b2mat, w2mat saves this information to the MAT file in the variables scale, offset, and fs. The signal itself is here called w.

E.3 Signal Down Sampling

This program is called downsmpl and performs a down-sampling of a signal by applying a low-pass filter and a decimation to the signal. The low-pass filtering is equivalent to the Matlab function FILTER. The application of downsmpl is to reduce large oversampled signals when Matlab does not have memory available to load the full signal. Here the advantage of downsmpl is that it operates on the signal in small blocks so it can theoretically operate on infinitely large signals.

The file name of the signal and the file name of the filter must be passed to downsmpl at the command line. These files must be in Matlab 4 format and the signal file must contain signal as the first variable and the sampling frequency in the variable `fs`. The filter file must contain the filter as the first variable and a decimation factor in a variable called `dec`.

Four options can be passed to downsmpl on the command line. The `-h` option makes downsmpl print following help to the terminal.

Terminal output

```
downsmpl version 1.2
  Equivalent to Matlab's FILTER
  usage: downsmpl [-hns] [-o Outfile] Infile.mat Filter.mat
    -h       Display this help
    -o File  Write output to file Outfile
    -n       No decimation
    -s       Silent
```

The `-o Outfile` option tells downsmpl to write the output to `Outfile`. Otherwise output will be written to `Infile.mat`. The `-n` option makes downsmpl perform no decimation and the `-s` makes downsmpl run silently.

E.4 Command Execution Timer

This program is called timer and is able start a program (or programs) at a specified time on a DOS machine. The actual application of timer is to start a number of data acquisitions.

The name of a file containing the program call and an execution time must be passed to timer on the command line. The request for the acquisitions are listed in a file which is read by timer. This file has a special syntax and could look like: `13:00 c:\myprog.exe arg1 arg2`. The time must be specified as the first five characters. This will run the command `c:\myprog.exe arg1 arg2` next time the system clock reaches 1 PM. A specific date can be specified by concatenating the time by an @-character and a date like: `13:00@1999.05.01`.

If timer is executed with the option `-h` following output is printed to the terminal:

Terminal output


```
usage: timer [-options] [commandfile]
options: -b: Make a beep at execution
         -h: Display this help
         -l: Writes a log file
         -s: Skip command if time has passed
         -v: Verbose
         -w: Write the time of execution to exectime.txt

Timer reads  'commandfile'  and executes the  commands given here at the
specified time. CRTL-Q quits the program.

                      Syntax for command file.
Example of a line in a command file: '13:00 c:\myprog.exe arg1 arg2'

This line will run myprog with arguments arg1 and arg2 at 1 PM. The time
must be specified in the first 5  characters as shown.  A period '.' can
be used to separate hours and  minutes instead of a colon.  A percentage
character  '%'  marks a comment, and can be placed anywhere in the line.
If the first line starts with a comment, it is echoed to the log file as
a title if the l option is on.
```

As described by the above output, other options can be given to timer. The `-l` option makes timer write a log file with the same name as the command file but with a `.log` extension. If the date and time of a command is in the past, timer will ignore it if the `-s` option is given. Otherwise it will be executed immediately. The `-v` will make timer print more terminal output and the `-w` option will make timer write the time of execution to `execute.txt`. This file will be overwritten by each new command so the executed command must move this file to a unique file name if the command file executes more than one command.

Appendix F

Ground Fault Experiment Data

This appendix provides information on the acquisitions systems and data from the ground fault experiments described in Chapter 6.

In chapter 6 selected examples of ground fault transient spectra was discussed. For the sake of completeness, this chapter also presents transient spectras for all locations, for both configurations, and for both voltage and current. Details on the experiment setup are given in Chapter 6.

F.1 Data Storage

This section provides details on the data storage such as file names and directory structure and a description of the acquired data.

F.1.1 Acquisition System Overview

In Section 6.4 on page 76 details on the acquired signals, the sensors, and the acquisition systems are provided. This section summarizes this infor-

system	station	alias	signals	f_s
TRA800	6464	wwkec520	three phase, probe/Rogowski	1 MHz
TRA800	5900	wwkec524	three phase, transformers	1 MHz
BAKKER	5900	be256	Petersen coil	1 MHz
DISMO-PC	6464	dismo	three phase, combi-sensors	20 kHz

Table F.1: Acquisition system location.

mation together with more data specific information such as file names and signal properties.

Four different acquisition systems were used electrically located at the feeding point of the network and by the Petersen coil, and physically located at two different transformer stations named 5900 and 6464.

Table F.1 lists the four acquisition systems, the transformer stations where they were located, an alias for each system, and the signals and sampling frequency for each system.

The TRA800's and the BAKKER are transient recorders which uses a sampling frequency of 1 MHz (with a few exceptions) and a time duration of 1 second. The DISMO-PC described in Section 2.2 on page 9 uses a sampling frequency of 19.84 kHz with a time duration of 20 seconds.

Station 6464 is the special coupling station shown in Figure 6.2 on page 70 and station 5900 contains the main breakers of the laboratory together with the Petersen coil.

The third column is a unique alias for each acquisition system. Apart from identifying the individual acquisition systems, the alias is furthermore used as a directory name on the CD's that store the data.

The fourth and the fifth column of Table F.1 describes the signals and the sampling frequencies for each acquisition system. The wwkec520 acquired a three phase voltage and current signal using probes and Rogowski coils at a sampling frequency of 1 MHz. The wwkec524 acquired a three phase voltage and current signal using voltage and current transformers at a sampling frequency of 1 MHz. The be256 acquired voltage and current at the Petersen coil using both probes, Rogowski coils, and voltage

signal	description
ur, us, ut	voltage on phase R, S, and T
ir, is, it	current on phase R, S, and T
u0p	zero system voltage from probe
u0t	zero system voltage from measurement transformer
i0r	zero system current from Rogowski coil
i0t	zero system current from measurement transformer

Table F.2: Description of signal names.

wwkec520		wwkec524		be256		dismo	
channel	signal	channel	signal	channel	signal	channel	signal
1	ur	1	us	1		1	ur
2	ir	2	is	2	u0p	2	us
3	us	3	ut	3	igf	3	ut
4	is	4	it	4	i0r	4	ir
5	ut	5	ur	5	i0t	5	is
6	it	6	ir	6	u0t	6	it

Table F.3: Acquisition system channel identification.

and current transformers at a sampling frequency of 1 MHz. The dismo acquired three phase voltage and current using the ABB combi-sensors at a sampling frequency of 19.84 kHz.

F.1.2 Original Acquisition Data

All preprocessing and conversion of data is prone to errors so it was considered important to always have the original data available. Therefore the acquired data was stored in the original unmodified 16 bit integer format. This section describes how to retrieve a given signal from the original data files.

Table F.2 provides a name for each acquired signal together with a description of the signal and Table F.3 provides information on which signal was acquired on each channel.

The two Tra800 transient recorders stores each channel in an individual file with a name like `0000n001.v01` for channel `n`. A conversion program called `w2mat` described in Section E.2.2 on page 165 is used to convert these acquisition files to a binary Matlab version 4 format. The acquisition files are stored in a numbered directory with an `.inx` extension. A journal was written during the experiments to keep track of which acquisitions corresponds to which file names. This information is implemented as a large array of structures returned by the Matlab function `KYVNAME` and used by the experiment data browser `KYVDATA` which is described in Section D.10 on page 159.

For the Bakker transient recorder and the DISMO-PC all channels are stored in one single file. It is therefore up to the conversion program in question to retrieve the channels. The DISMO-PC saves output to a `DAT` file and Section E.2.2 on page 165 describes the `dat2mat` program used to convert the data files to a binary Matlab version 4 format. A similar program called `b2mat` (see Section E.2.2 on page 165) is used to convert the Bakker files.

To give an example, suppose that we need to find the file name for the current of phase R for location 1, configuration 2, 0.5 kΩ fault resistance, a closing angle of 90° acquired with the Rogowski coil at a sampling frequency of 1 MHz. From Table F.1 we find that the phase current acquired with the Rogowski coil at a sampling frequency of 1 MHz is the `wwkec520` system and from Table F.3 we find that the phase R current at `wwkec520` is channel 4. The root directory is therefore `wwkec520` and the file name is `00004001.v01`. The rest of the information in connection with `KYVNAME` is used to provide the two directory names `nesa02.800/077.inx`. This leads to the full name: `/wwkec520/nesa02.800/077.inx/00004001.v01`.

If the example above concerned the current in the Petersen coil acquired through the current transformer, the process of finding the original data is as follows: the alias and the data root directory is found in Table F.1 for the Bakker transient recorder as `be256`. As above, the location, the configuration, the resistance, and the closing angle is used in connection with `KYVNAME` to retrieve the name `nesa02.066` which in this case is the name of the file containing all channels of the acquisition. This means that the full file name is `be256/nesa02.066` for the original acquisition file containing all channels. To retrieve the channel number we find the signal name from Table F.2 as `i0t` and from Table F.3 we find that `i0t` is channel 5 on the `be256`.

For the DISMO-PC data the directory name can be retrieved as above using `KYVNAME`. In addition, the DISMO-PC allowed the directory names to be chosen freely, so descriptive names was used. The top level directory was the alias `dismo`. The second level was `conf1` and `conf2` for the two different configurations. The data was stored in `DAT` files with a filename on the form `lxry_nnn.dat` where `x` is the location number, `y` represents the resistance, and `nnn` is a consecutive numbered sequence. The different resistances 0, 0.5, 1, 2, 10, and 20 kΩ is represented by the numbers 0, 1, 2, 3, 4, and 5 respectively. The numbered sequence typically represents the different closing angles 0, 30, 60, 90, 120, 150, and 180° with the numbers `000`, `001`, `002`, `003`, `004`, `005` and, `006` respectively, but there are exceptions to this rule so it is necessary to check with `KYVNAME` for the closing angle. This means that the full path of the current of phase R in the first example is `/dismo/conf2/l1r1_003.dat`.

F.1.3 Acquisition Errors

When experiments of this scale are performed it is inevitable that errors will occur in the acquired data. This section lists the errors that are known to exist at the time of writing.

Channel 3 on the `be256` was used for the ground fault current in a few acquisitions at location 1. By accident this channel was saved and stored for all acquisitions at this location except for the 0 Ω resistance, eventhough no sensor was attached to the channel. This means that these signals will all be zero except for those six signals in the range from `be256/nesa10.303` to `be256/nesa10.308` both included which is the true ground fault current.

The Tra800 transient recorders had major stability problems due to the operating system they were running (MS WINDOWS 95). The actual acquisition data were stored in a memory block private to each channel so the data was not affected but the instability made a frequent reboot of `wwkec520` and `wwkec524` necessary. This was very time consuming and after a few days a tecnician from the producer of the transient recorder came up from Switzerland to trouble shoote. The OS was changed to MS WINDOWS 3.1 which solved the problem for `wwkec520`. For the `wwkec524`, however, the problem was never solved so it had to be abandoned. The result was that `wwkec524` only was used for location 1 which means that the acquisitions on the voltage and current transformers was only performed for location 1.

The DISMO-PC had a strange (hardware) fault which had the effect that two or more channels were interchanged in the acquisition file. To make things worse not all files have this problem and the affected channels was not always the same. The voltages and currents are easily distinguishable, however, and with reference to the other acquisition systems and the three phase properties of the signals this problem can be detected and resolved. The Matlab function `EXTDATA` described in Section D.9 on page 158 implements this solution. Note that it is important to be aware of this when the original data is used.

Besides the problem described above, the DISMO-PC had a fatal error on channel 4, the current on phase R, caused by a defect anti-aliasing filter. Unfortunately this error resulted in loss of data from this channel for following series for both configuration 1 and 2 and for all closing angles.

location 2 : $2\,\mathrm{k}\Omega$ resistance.

location 3 : all resistances.

location 6 : $0.5\,\mathrm{k}\Omega$ and $1\,\mathrm{k}\Omega$ resistance.

location 7 : all resistances.

As described in the Section F.1.1 three full sets of three phase voltage and current acquisitions were performed. Two of these were acquired on the same physical location in station `6464` and one set on the measurement transformers in station `5900`. The sensors in `6464` were visible as shown in Figure 6.9 on page 78 so it was easy to assign phase names (R, S, and T) to the channels. The measurement transformers, however, was build into the main curcuit breaker in `5900` so it was not possible to determine which phase was R, S, and T by visual inspection. The phases are therefore rotated for `wwkec524` compared to `wwkec520` as reflected by Table F.3. This explains the inconsistency in the channel assignment and is not actually considered an error.

F.1.4 Extracted Acquisition Data

The full data set is very large and takes up as much disk space as 21 CD's so it is very hard to get an overview of such an amount of data. Therefore a reduced version of the data has been produced by down-sampling the original (TRA800 and BAKKER) data to 100 kHz and by selecting 12 ms of the signal centered around the fault transient (the time instant of fault connection). This process was carried out by a Matlab function called `EXTDATA` which is described in Section D.9 on page 158. The resulting data fits one single CD and is the data basis of the data browser `KYVDATA` described in Section D.10 on page 159.

The terminal output from `EXTDATA` discribes all actions performed on the data and can (together with the `EXTDATA` source) be used as reference for the down-sampling process. This output can be found on the CD with the extracted data in directory `/out` and with file names from `cd01.out` to `cd17.out` for the 17 CD's with transient recorder data and from `dismo1.out` to `dismo4.out` for the DISMO-PC data.

Although these post-processed signals provides easy access to the data, the primary source of information is still the original unmodified 21 data CD's.

All the extracted data is stored in Matlab version 4 format and each signal (channel) is stored in its own file. The names of these files are the signal names listed in Table F.3.

alias	sensor	signals	sensitivity	factor
`wwkec520`	probe	`ur us ut`	1 V/1000 V	1000 V
`wwkec520`	PEM	`ir is it`	4 mV/A	250 A
`wwkec524`	VT, A1	`ur us ut`	10 V/0.11 kV·1/800	8800 V
`wwkec524`	CT, A1	`ir is it`	150 V/5 A·1/1000	33.33 A
`be256`	probe	`u0p`	1 V/1000 V	1000 V
`be256`	VT, PC	`u0t`	11 V/($\sqrt{3}$·0.11 kV)·3/50	288.7 V
`be256`	Pearson	`i0r`	10 mV/A	100 A
`be256`	CT, PC	`i0t`	1 V/2 A·1/5	10 V
`dismo`	CS, VD	`ur us ut`	—	0.55 V
`dismo`	CS, RC(a)	`ir is it`	—	0.049 A
`dismo`	CS, RC(b)	`ir is it`	—	0.032 A

Table F.4: Conversion factors for all acquisition systems.

F.2 Conversion Factors

This section provides the conversion factors for all sensors used during the ground fault experiments. In order to retrieve the data in physical units (volt or ampere) from either the original data files or the extracted data files the signal must be multiplied with a conversion factor.

All these conversion factors are implemented in the Matlab function `KYVFAC` found on the CD with the extracted data.

F.2.1 Conversion Factors for All Systems

Table F.4 lists the conversion factors for all systems and sensors. An overview of the equipment and a definition of the aliases, the sensors, and the signal names in the table is provided in Section F.1.1 on page 171.

The fourth column of the table lists the sensitivity stated by the sensor specifications. For the measurement transformers an additional factor is applied to make the signal level match the transient recorder. The measurement transformers are designed to trip a circuit breaker or other protection gear where relatively high signal levels are needed. In order

to convert the signals to a voltage with the right level, the current transformer was connected to a small resistor which was used to convert the current output to a voltage output. These voltage outputs was then connected to a Watt-meter which had convenient signal outputs that matched the transient recorder input level. The additional factor listed in the sensitivity column after the last · is the level conversion applied by the Watt-meter and the resistor for the current transformers.

The three last rows for the `dismo` system has a specific sensitivity for each sensor and therefore for each phase and is a bit more complicated to derive than for the other sensors. These entries are therefore left blank and will be discussed in Section F.2.2.

During the experiments the conversion factor for the current part of the `dismo` system had to be changed therefore two different factors named (a) and (b) are listed.

The fifth column lists the conversion factor which must be multiplied with the signal in order to get the true physical units. For many applications the precision of the conversion factor is not crusial so a mean value of the conversion factors for three phases for the `dismo` system is provided in the table.

Following list describes the sensors in Table F.4. Section 6.4 on page 76 provides additional information and pictures of the data acquisition equipment.

probe — voltage probe with a sensitivity of 1:1000 V.

PEM — a flexible Rogowski coil which is wound two times around the cable with a resulting sensitivity of 4 mV/A.

Pearson — a solid Rogowski coil with a sensitivity of 10 mV/A.

VT, A1 — voltage transformer mounted in the main circuit breaker A1 with a sensitivity of 10:110 V and with an additional factor 1:800 applied.

CT, A1 — current transformer mounted in the main circuit breaker A1 with a sensitivity of 150 V/5 A and with an additional factor

1:1000 applied.

VT, PC — voltage transformer mounted in the Petersen coil with a sensitivity of $11\,\mathrm{V}/(\sqrt{3}\cdot 0.11\,\mathrm{kV})$ and with an additional factor 3:50 applied.

CT, PC — current transformer mounted in the Petersen coil with a sensitivity of $2\,\mathrm{V/A}$ and with an additional factor 1:5 applied.

CS, VD — resistive voltage divider as a part of the combi-sensor.

CS, RC(a) — Rogowski coil as part of the combi-sensor set to sensitivity level (a).

CS, RC(b) — Rogowski coil as part of the combi-sensor set to sensitivity level (b).

F.2.2 Conversion Factors for DISMO-PC

The signals stored by the DISMO-PC are 16 bit signed integers, and must be scaled with a conversion factor in order to become a physical unit (ampere or volt). This section deals with the calculation of this conversion factor.

The input signal is first passed through an anti-aliasing filter with a cut-off frequency of $10\,\mathrm{kHz}$ and then through an amplifier to make the signal match the input range of the A/D converter.

The anti-aliasing filter is designed as an eliptical filter and the transfer function magnitude and argument are measured with an HP4195A spectrum analyzer and shown in Figure F.1 and F.2. The eliptical filter has an argument function which is close to being linear for lower frequencies. A linear argument function is a nice feature for the filter as it results in a constant frequency independent group delay. The linear property of the argument function does of course not show in Figure F.2 on a logarithmic axis.

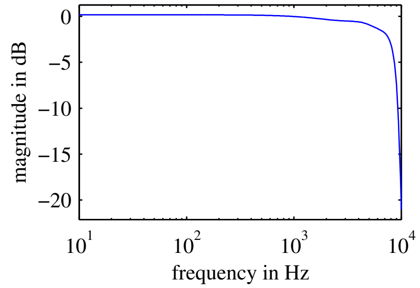


Figure F.1: Magnitude of anti-aliasing filter response.

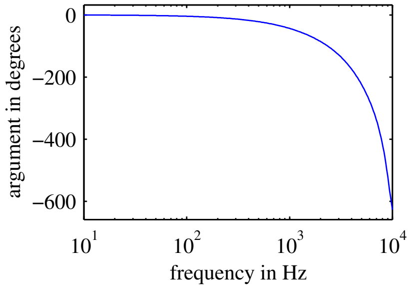


Figure F.2: Argument of anti-aliasing filter response.

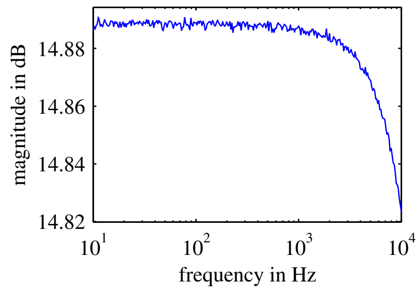


Figure F.3: Magnitude of amplifier response.

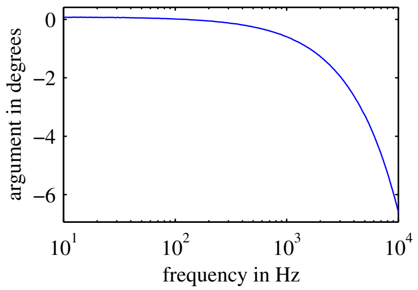


Figure F.4: Argument of amplifier response.

As seen from Figure F.1 the filter magnitude is very close to zero dB (unity) for frequencies below $1\,\mathrm{kHz}$. In terms of the conversion factor the anti-aliasing filter will therefore be ignored.

The amplifier has the purpose of making the signal match the input range of the A/D converter so the largest possible range is used without causing overflow. This minimizes quantization error and external noise. The magnitude and argument of an amplifier was measured with

	voltage	current (a)	current (b)
phase R	5.552	9.972	5.154
phase S	5.562	10.19	5.127
phase T	5.548	9.998	5.149

Table F.5: Amplifier gain for the DISMO-PC.

phase	calibration factor
R	1.0090
S	1.0072
T	1.0092

Table F.6: Calibration factors for combi-sensors.

an HP4195A spectrum analyzer and is shown in Figure F.3 and F.4. It is seen from these figures that the variation below 10 kHz is very limited so it is a reasonable approximation to regard the amplifier as constant with no argument shift.

Table F.5 summarizes the amplifier gain for the different channels of the DISMO-PC used during the ground fault experiments. Note that the gain for the current sensors were changed during the experiments and therefore Table F.5 has two columns for the current sensor names denoted with (a) and (b).

The input range of the A/D converter is ± 10 V which is converted to the 16 bit signed integer range $\pm 2^{15}$ for all signals. The voltage sensor is a resistive voltage divider and is measured between one phase and neutral. The sensitivity of the voltage divider is 1:10000 V. If the amplifier gain is called G, the conversion factor for the voltage sensor k_v is

$$k_v = \frac{10000}{G}\frac{10\text{V}}{2^{15}} = \frac{3.052}{G}V \tag{F.1}$$

The current part of the sensor has three sensitivity levels: 150 mV corresponds to either 80 A, 240 A, or 640 A. In addition a calibration factor is

phase	voltage	current (a)	current (b)
R	0.5497	0.04940	0.03186
S	0.5487	0.04824	0.03197
T	0.5501	0.04929	0.03190

Table F.7: Conversion factors for DISMO-PC.

provided by the manufacturer which listed in Table F.6. If this sensitivity level is called S, the calibration factor is called C, and the amplification gain described above is called G, the conversion factor k_c for the current signal is:

$$k_c = \frac{10 \cdot CS}{2^{15} \cdot G \cdot 0.150} A = 2.035 \cdot 10^{-3} \frac{CS}{G} A \tag{F.2}$$

Table F.7 summarizes the conversion factors for the combi-sensors.

Using the amplifier gain from Table F.5, the calibration factor from Table F.6 and a sensitivity level of 240 A for the (a) factor and 80 A for (b), the resulting conversion factors for the DISMO-PC is listed in Table F.7.

F.3 Comparison of the Different Locations

The ground fault localization algorithm derived in Chapter 5 computes an error measure in the frequency domain which depends on sufficiently large variations of the ground fault transient spectras across the network.

In this section, the six different ground fault locations are compared to each other in the frequency domain to display the signal changes when the fault location is moved across the network. See Section 6.6.2 on page 85 for more details.

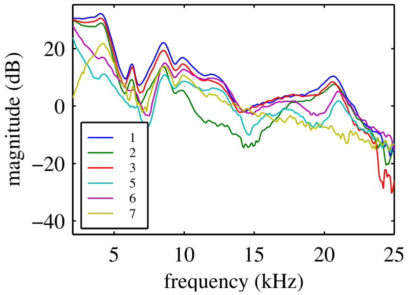


Figure F.5: Magnitude spectrum of voltage transient for all locations on configuration 1

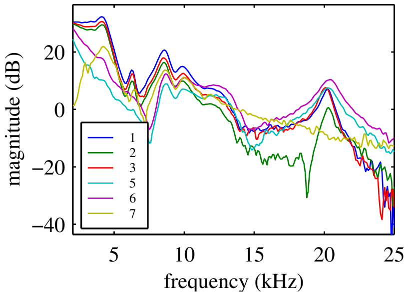


Figure F.6: Magnitude spectrum of voltage transient for all locations on configuration 2

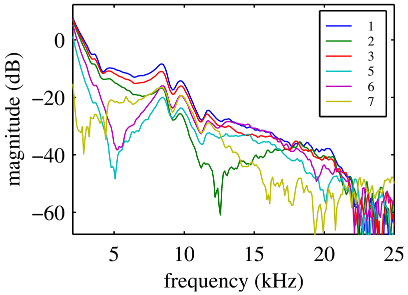


Figure F.7: Magnitude spectrum of current transient for all locations on configuration 1

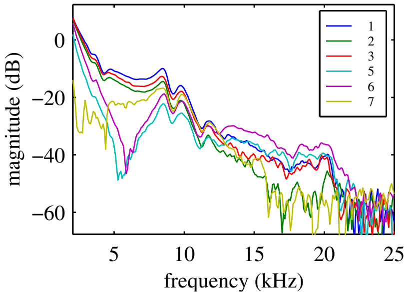


Figure F.8: Magnitude spectrum of current transient for all locations on configuration 2

F.4 Simulation of the Experiment Data

This section gives the complete set of figures in the comparison between the experimental and the simulated data which is given in Section 6.6.4 on page 87. This comparison has the purpose of validating the simulation model on the experimental data. An outline of the experiments is provided in Section 6.2 on page 71.

The comparison is made for both the voltage and current signal and for both configuration 1 and 2. For these four combinations the comparison between experimental and simulated data is made for all six fault locations.

The simulations are computed by ATP and the models are described in Appendix C. The cable type and cable lengths used for the network model are listed in Table C.2 on page 125. The impedance and admittance matrices for the cable model are computed at both 2 kHz and 20 kHz. Together with the acquired signal this gives three signals for each figure.

Voltage, Configuration 1

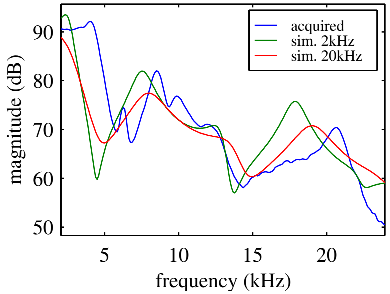


Figure F.9: Voltage, location 1, configuration 1.

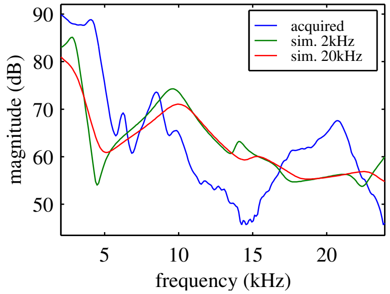


Figure F.10: Voltage, location 2, configuration 1.

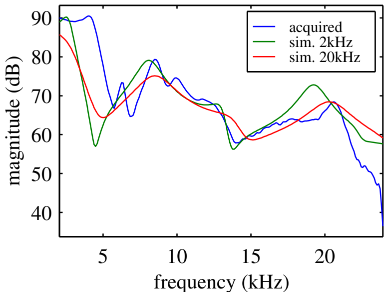


Figure F.11: Voltage, location 3, configuration 1.

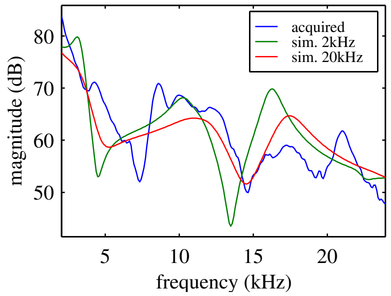


Figure F.12: Voltage, location 5, configuration 1.

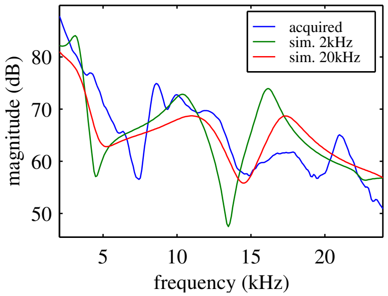


Figure F.13: Voltage, location 6, configuration 1.

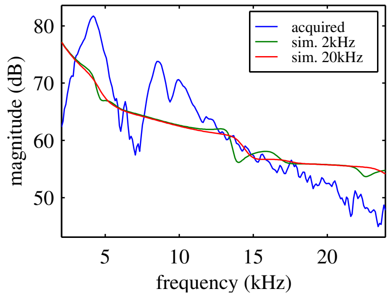


Figure F.14: Voltage, location 7, configuration 1.

Current, Configuration 1

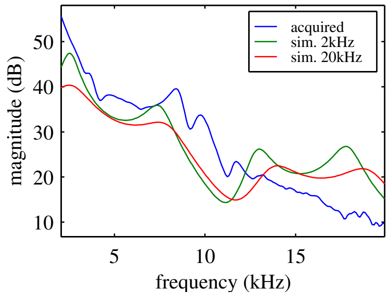


Figure F.15: Current, location 1, configuration 1.

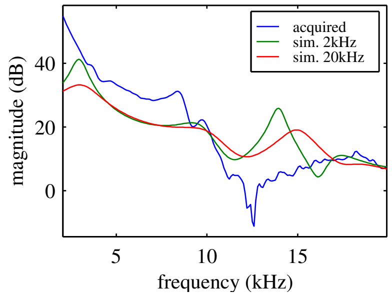


Figure F.16: Current, location 2, configuration 1.

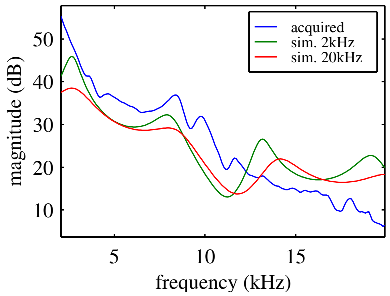


Figure F.17: Current, location 3, configuration 1.

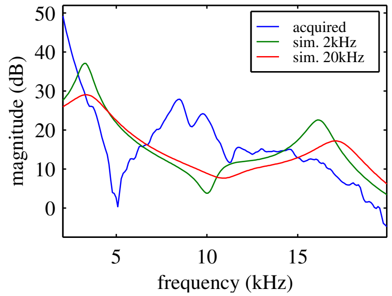


Figure F.18: Current, location 5, configuration 1.

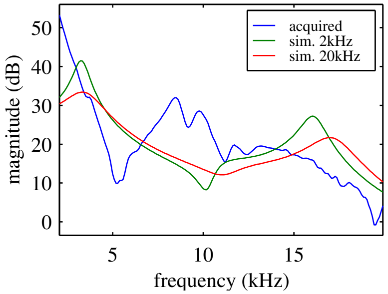


Figure F.19: Current, location 6, configuration 1.

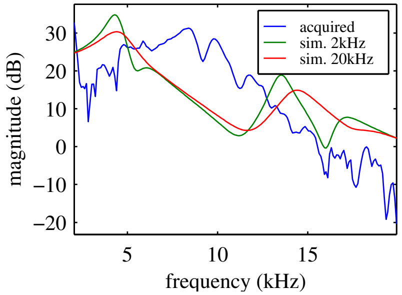


Figure F.20: Current, location 7, configuration 1.

Voltage, Configuration 2

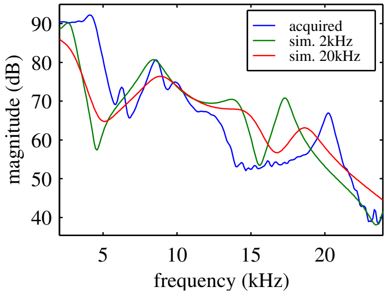


Figure F.21: Voltage, location 1, configuration 2.

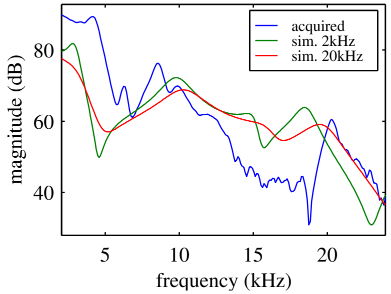


Figure F.22: Voltage, location 2, configuration 2.

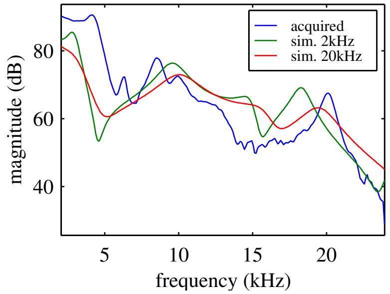


Figure F.23: Voltage, location 3, configuration 2.

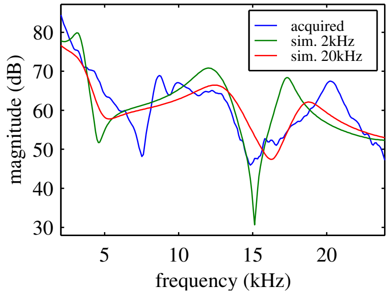


Figure F.24: Voltage, location 5, configuration 2.

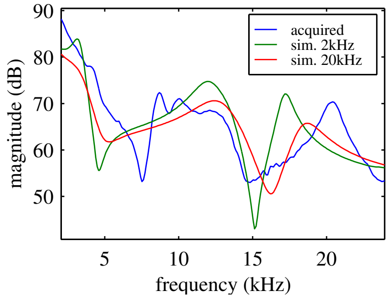


Figure F.25: Voltage, location 6, configuration 2.

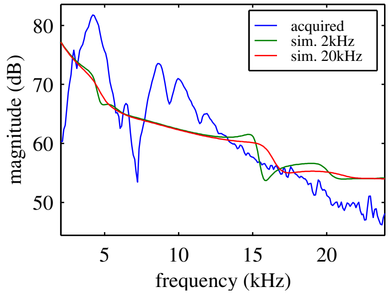


Figure F.26: Voltage, location 7, configuration 2.

Current, Configuration 2

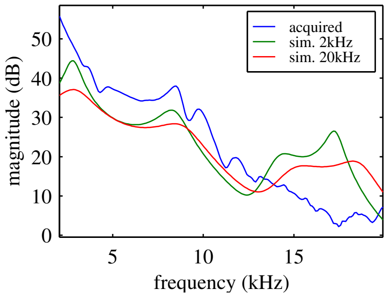


Figure F.27: Current, location 1, configuration 2.

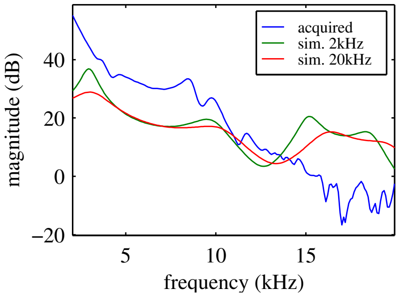


Figure F.28: Current, location 2, configuration 2.

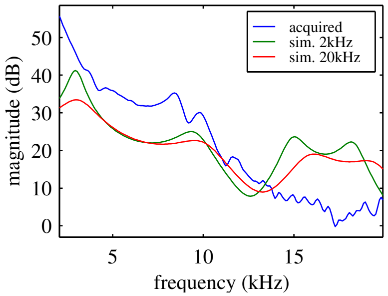


Figure F.29: Current, location 3, configuration 2.

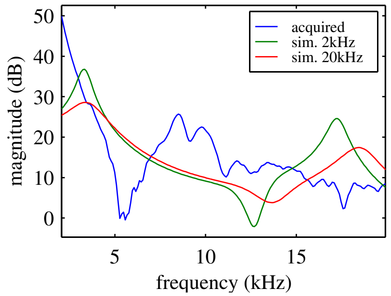


Figure F.30: Current, location 5, configuration 2.

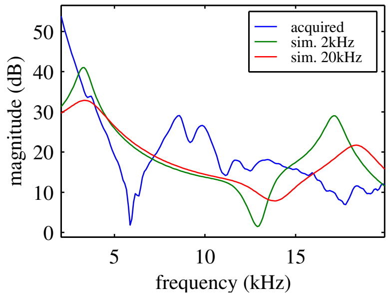


Figure F.31: Current, location 6, configuration 2.

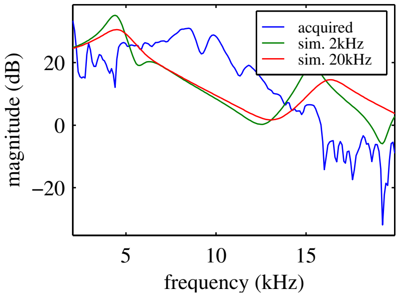


Figure F.32: Current, location 7, configuration 2.

Bibliography

[Ahmed and Natarajan, 1983] Nasir Ahmed and T. Natarajan. *Discrete–time Signals and Systems.* Prentice–Hall, Inc., 1983.

[Allan, 1991] Greenwood Allan. *Electrical Transients in Power Systems.* John Wiley & Sons, Inc., 1991.

[Bo *et al.*, 1997] Z. Q. Bo, A. T. Johns, and R. K. Aggarwal. A Novel Fault Locator Based On The Detection of Fault Generated High Frequency Transients. *Developments in Power System Protection*, March 1997.

[Boel, 1983] Simon Boel. *Signalanalyse 1.* Laboratoriet for Akustik, DTU, 1983.

[Dommel, 1981] Hermann W. Dommel. *EMTP Theory Book.* Bonneville Power Administration, Portland Oregon 97208–3621, USA, 1981.

[Gunnarsson, 1998] Helgi Gunnarsson. *Detection of Ground Faults in Distribution Networks.* Master's thesis, Departement of Mathematical Modeling, Technical University of Denmark, 1998.

[Haykin, 1994] Simon Haykin. *Neural Networks.* Macmillian Publishing Company, 1994.

[Haykin, 1996] Simon Haykin. *Adaptive Filter Theory.* Prentice–Hall, Inc., third edition, 1996.

[Høg, 1994] Rasmus Høg. *Analyse af Transienter i El–distributionsnetværk, Projektrapport + Appendiks.* Master's thesis, The Electronics Institute, Technical University of Denmark, March 1994.

[Jensen *et al.*, 1998] Kåre Jean Jensen, Steen M. Munk, and John Aasted Sørensen. Feature Extraction Method for High Impedance Ground Fault Localization in Radial Power Distribution Networks. *International Conference on Acoustics, Speech and Signal Processing*, 1998.

[Jespersen, 1994] Jesper Kring Jespersen. *Analyse af Transienter i El–distributionsnetværk, Detektion.* Master's thesis, The Electronics Institute, Technical University of Denmark, September 1994.

[Krauss, 1991] John D. Krauss. *Electromagnetics.* mcgraw–Hill, inc., fourth edition, 1991.

[Lakervi and Holmes, 1989] E. Lakervi and E. J. Holmes. *Electricity Distribution Network Design.* Peter Peregrinus Ltd., London, 1989.

[Leu, 1987] Leuven EMTP Center. *ATP Rule Book*, 1987.

[Madsen, 1996] Mads K. Madsen. *Analyse af Transienter i El–distributionsnetværk, Wavelet Packets og Klassifikation ved hjælp af Neurale Netværk + Appendiks.* Master's thesis, The Electronics Institute, Technical University of Denmark, 1996.

[Mähönen *et al.*, 1996] Pentti Mähönen, Markku Moisio, Tapio Hakola, and Harri Kuisti. The Rogowski Coil and the Voltage Divider in Power System Protection and Monitoring. *Cigré*, pages 34–103, 1996.

[Matti, 1992] Lehtonen Matti. *Transient Analysis for Ground Fault Distance Estimation in Electrical Distribution Networks.* PhD thesis, Technical Research Center of Finland, 1992.

[Munk and Sørensen, 1997] Steen M. Munk and John Aasted Sørensen. A Consept for Monitoring Radial Distribution Networks Based on Very Few Measurements. *Proceedings of the 4th International Workshop on Systems, Signals and Image Processing*, May 1997.

[Munk, 1995] Steen M. Munk. *Centralized Monitoring of 10 kV Cable Based Radial Distribution Networks.* PhD thesis, The Electronics Institute, Technical University of Denmark, August 1995.

[Nielsen, 1995] Jacob Thymann Nielsen. *Analyse af Transienter i El–distributionsnetværk, Waveletanalyse + Appendiks.* Master's thesis, The Electronics Institute, Technical University of Denmark, August 1995.

[NKT, 1992] NKT Engineering, Vibeholms Allé 22, 2605 Brønshøj, Denmark. *NKT Katalog*, 1992.

[Noda *et al.*, 1996] T. Noda, N. Nagaoka, and A. Ametani. Phase Domain Modeling of Frequency–Dependent Transmission Lines by Means of an ARMA Model. *IEEE Transactions on Power Delivery*, pages 401–411, January 1996.

[Oppenheim, 1989] Schafer Ronal W. Oppenheim, Alan V. *Discrete–time Signal Processing.* Prentice–Hall, Signal Processing Series, 1989.

[Strang and Nguyen, 1996] Gilbert Strang and Truong Nguyen. *Wavelets and Filter Banks.* Wellesley–Cambridge Press, 1996.

[Vetterli and Kovačević, 1995] Martin Vetterli and Jelana Kovačević. *Wavelets and Subband Coding.* Prentis–Hall International, Inc., 1995.

[Zhu *et al.*, 1997] Jun Zhu, David L. Lubkeman, and Adly A. Girgis. Automated Fault Location and Diagnosis On Electric Power Distribution Feeders. *IEEE Transactions on Power Delivery*, 12:801–807, April 1997.

Index