Signal processing for distribution network monitoring

Jensen, Kåre Jean

Publication date: 1999

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.
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
Contents

Preface vi

Abstract xi

Résumé (Abstract in Danish) x

1 Introduction
1.1 Background .............................................
1.2 The NESA Distribution Network ......................
1.3 The Centralized Monitoring Concept ............
1.4 Ground Fault Localization ........................

2 Normal Operation Feeder Activity
2.1 The Combi-Sensors ....................................
2.2 The DISMO-PC ........................................
2.3 Customers at the Feeder ............................
2.4 Normal Operation Data ................................

3 Mathematical Network Analysis ..........................
3.1 Symmetrical Components ..........................
3.2 Distributed Parameter Transmission Line ........
3.3 Lumped Element Cable Model ......................
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Numerical Simulations of Network Signals</td>
<td>21</td>
</tr>
<tr>
<td>4.1 Cable Model</td>
<td>22</td>
</tr>
<tr>
<td>4.1.1 Choice of Cable Model</td>
<td>23</td>
</tr>
<tr>
<td>4.1.2 Parameters for the Cable Model</td>
<td>24</td>
</tr>
<tr>
<td>4.1.3 Distributed Ground Resistance</td>
<td>25</td>
</tr>
<tr>
<td>4.1.4 Three Phase II-section</td>
<td>26</td>
</tr>
<tr>
<td>4.1.5 II-section with Ground Resistance</td>
<td>28</td>
</tr>
<tr>
<td>4.2 Transformer Models</td>
<td>31</td>
</tr>
<tr>
<td>4.3 Petersen Coil</td>
<td>35</td>
</tr>
<tr>
<td>4.4 Network Model</td>
<td>36</td>
</tr>
<tr>
<td>4.5 Impulse Response Generation</td>
<td>37</td>
</tr>
<tr>
<td>4.6 Ground Fault Simulations</td>
<td>39</td>
</tr>
<tr>
<td>4.7 Precision of ATP Simulations</td>
<td>39</td>
</tr>
<tr>
<td>4.7.1 Valid Frequency Range</td>
<td>40</td>
</tr>
<tr>
<td>4.7.2 Model Bandwidth</td>
<td>42</td>
</tr>
<tr>
<td>4.7.3 Step Frequency</td>
<td>43</td>
</tr>
<tr>
<td>4.7.4 Summary</td>
<td>45</td>
</tr>
<tr>
<td>5 Ground Fault Localization</td>
<td>47</td>
</tr>
<tr>
<td>5.1 Existing Methods</td>
<td>47</td>
</tr>
<tr>
<td>5.2 Impulse Response Model of Feeder</td>
<td>48</td>
</tr>
<tr>
<td>5.3 The Deconvolution Approach</td>
<td>50</td>
</tr>
<tr>
<td>5.4 Deconvolution in the Frequency Domain</td>
<td>53</td>
</tr>
<tr>
<td>5.5 Ground Fault Model</td>
<td>55</td>
</tr>
<tr>
<td>5.6 Ground Fault Localization Estimation</td>
<td>56</td>
</tr>
<tr>
<td>5.7 Simulation Results</td>
<td>58</td>
</tr>
<tr>
<td>5.7.1 Two Different Load Conditions</td>
<td>59</td>
</tr>
<tr>
<td>5.7.2 Impulse Response Model</td>
<td>39</td>
</tr>
<tr>
<td>5.7.3 Ground Faults on the Load 1 Condition</td>
<td>59</td>
</tr>
<tr>
<td>5.7.4 Ground Faults on the Load 2 Condition</td>
<td>61</td>
</tr>
<tr>
<td>5.8 Fault Resistance</td>
<td>65</td>
</tr>
<tr>
<td>6 Full Scale Ground Fault Experiment</td>
<td>6</td>
</tr>
<tr>
<td>6.1 Laboratory Network Overview</td>
<td>7</td>
</tr>
<tr>
<td>6.2 Experimental Outline</td>
<td>7</td>
</tr>
<tr>
<td>6.3 Ground Fault Equipment</td>
<td>7</td>
</tr>
<tr>
<td>6.4 Acquisition Equipment</td>
<td>7</td>
</tr>
<tr>
<td>6.5 The Experiment Procedure</td>
<td>7</td>
</tr>
<tr>
<td>6.6 Acquired Data</td>
<td>8</td>
</tr>
<tr>
<td>6.6.1 Phase Measurements</td>
<td>8</td>
</tr>
<tr>
<td>6.6.2 Faulted Phase for All Locations</td>
<td>8</td>
</tr>
<tr>
<td>6.6.3 Ground Fault Current</td>
<td>8</td>
</tr>
<tr>
<td>6.6.4 Modeling the Experiment Data</td>
<td>8</td>
</tr>
<tr>
<td>7 Conclusion and Future Work</td>
<td>9</td>
</tr>
<tr>
<td>7.1 Conclusions</td>
<td>9</td>
</tr>
<tr>
<td>7.2 Suggestions for Future Work</td>
<td>9</td>
</tr>
<tr>
<td>7.2.1 Localization Algorithm</td>
<td>9</td>
</tr>
<tr>
<td>7.2.2 Network Element Models</td>
<td>9</td>
</tr>
<tr>
<td>7.2.3 Ground Fault Current</td>
<td>9</td>
</tr>
<tr>
<td>8 Signal Processing Algorithms</td>
<td>9</td>
</tr>
<tr>
<td>A.1 Iterative Deconvolution Algorithm</td>
<td>9</td>
</tr>
<tr>
<td>A.2 Design of Digital Integrator</td>
<td>10</td>
</tr>
<tr>
<td>A.2.1 Integration of a periodic signal</td>
<td>10</td>
</tr>
<tr>
<td>A.2.2 Low-pass filter</td>
<td>10</td>
</tr>
<tr>
<td>A.2.3 IIR filter designed using bilinear transformation</td>
<td>10</td>
</tr>
<tr>
<td>A.2.4 Discussion</td>
<td>10</td>
</tr>
<tr>
<td>B Transfer Function for II-section</td>
<td>10</td>
</tr>
<tr>
<td>C ATP Network Models</td>
<td>11</td>
</tr>
<tr>
<td>C.1 The Primary Input File</td>
<td>11</td>
</tr>
<tr>
<td>C.2 The Main Input File</td>
<td>11</td>
</tr>
<tr>
<td>C.3 Ground Fault Simulation Model</td>
<td>11</td>
</tr>
<tr>
<td>C.4 Node Naming Conventions</td>
<td>11</td>
</tr>
<tr>
<td>Contents</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td></td>
</tr>
<tr>
<td>C.5 File Naming Conventions</td>
<td>119</td>
</tr>
<tr>
<td>C.6 Cable Network Model</td>
<td>119</td>
</tr>
<tr>
<td>C.6.1 Input File Syntax for makenet</td>
<td>120</td>
</tr>
<tr>
<td>C.6.2 Network Data at A12</td>
<td>123</td>
</tr>
<tr>
<td>C.6.3 Network Data at the 10 kV Laboratory</td>
<td>123</td>
</tr>
<tr>
<td>C.7 Conversion of the ATP output file</td>
<td>123</td>
</tr>
<tr>
<td>C.8 The Complete Impulse Response Model</td>
<td>126</td>
</tr>
<tr>
<td>D Matlab Functions</td>
<td>129</td>
</tr>
<tr>
<td>D.1 Variable Naming Conventions</td>
<td>131</td>
</tr>
<tr>
<td>D.2 Ground Fault Localization Algorithm</td>
<td>131</td>
</tr>
<tr>
<td>D.3 Visualization of the Localization Algorithm Output</td>
<td>135</td>
</tr>
<tr>
<td>D.4 Ground Fault Model</td>
<td>137</td>
</tr>
<tr>
<td>D.5 ATP File Name Extractor</td>
<td>138</td>
</tr>
<tr>
<td>D.6 ATP Source Generation</td>
<td>140</td>
</tr>
<tr>
<td>D.7 ATP Cable Model Generator</td>
<td>141</td>
</tr>
<tr>
<td>D.8 The DISMO-Toolbox</td>
<td>148</td>
</tr>
<tr>
<td>D.8.1 The Main Menu</td>
<td>149</td>
</tr>
<tr>
<td>D.8.2 The Analysis Menu</td>
<td>150</td>
</tr>
<tr>
<td>D.8.3 Implementation</td>
<td>155</td>
</tr>
<tr>
<td>D.9 Experiment Data Extractor</td>
<td>158</td>
</tr>
<tr>
<td>D.10 Experiment Data Browser</td>
<td>159</td>
</tr>
<tr>
<td>D.10.1 Overview</td>
<td>159</td>
</tr>
<tr>
<td>D.10.2 User Interface</td>
<td>161</td>
</tr>
<tr>
<td>E C++ Utility Programs</td>
<td>163</td>
</tr>
<tr>
<td>E.1 Network Model Generation</td>
<td>164</td>
</tr>
<tr>
<td>E.2 Data Conversion</td>
<td>165</td>
</tr>
<tr>
<td>E.2.1 ATP Output Conversion</td>
<td>165</td>
</tr>
<tr>
<td>E.2.2 Acquisition Data Conversion</td>
<td>165</td>
</tr>
<tr>
<td>E.3 Signal Down Sampling</td>
<td>167</td>
</tr>
<tr>
<td>E.4 Command Execution Timer</td>
<td>168</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>F Ground Fault Experiment Data</td>
</tr>
<tr>
<td>F.1 Data Storage</td>
</tr>
<tr>
<td>F.1.1 Acquisition System Overview</td>
</tr>
<tr>
<td>F.1.2 Original Acquisition Data</td>
</tr>
<tr>
<td>F.1.3 Acquisition Errors</td>
</tr>
<tr>
<td>F.1.4 Extracted Acquisition Data</td>
</tr>
<tr>
<td>F.2 Conversion Factors</td>
</tr>
<tr>
<td>F.2.1 Conversion Factors for All Systems</td>
</tr>
<tr>
<td>F.2.2 Conversion Factors for DISMO-PC</td>
</tr>
<tr>
<td>F.3 Comparison of the Different Locations</td>
</tr>
<tr>
<td>F.4 Simulation of the Experiment Data</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bibliography</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Index</th>
</tr>
</thead>
</table>
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 Distribution Network Monitoring, and is a project in monitoring medium voltage power distribution networks. This Ph.D. project directly followed the Ph.D. project “Centralized Monitoring of 10 kV Cable Based Radiation 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), Jørgen Aasted Sørensen (IMM), Henrik Weldingh (DEFU), and Morten Lin (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 the 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 II-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:

<table>
<thead>
<tr>
<th>Text symbols</th>
<th>Mathematical symbols</th>
</tr>
</thead>
<tbody>
<tr>
<td>trademark</td>
<td>$x$</td>
</tr>
<tr>
<td>program name</td>
<td>\text{ProgName}</td>
</tr>
<tr>
<td>file type</td>
<td>\text{EXT}</td>
</tr>
<tr>
<td>file name</td>
<td>\text{/dir/name.ext}</td>
</tr>
<tr>
<td>geographical location</td>
<td>\text{Geographical Location}</td>
</tr>
<tr>
<td>ATP routine</td>
<td>\text{ATP ROUTINE}</td>
</tr>
<tr>
<td>ATP variable</td>
<td>\text{ATPVAR}</td>
</tr>
</tbody>
</table>

**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 10kV 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), Bernt 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 the 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
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 exists. The method is verified successfully on simulated data. Experimental data collected 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 of 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**


Der blev udført forsøg med jordfejl på et mellemhændings-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 højpræcisionsskontakt, og spændinger og strømme blev opsat ved de forsyndende transformer ved hjælp af både måletransformere, probe 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 market.

In this new marked of competition it is essential to optimize resources 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. Through
Introduction

out 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, 1993] 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 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 5,800 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 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.

1The DSO-project, 1991–1994, was a project on control and monitoring of distribution networks accomplished as a cooperation between several Danish power distribution utilities.
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 bandwidth 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 50kV/10kV transformer station at Glentegården. Glentegården is a main transformer station located at Buddinge just North of Copenhagen. This 30kV/10kV 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 10kV feeder A12 by the transformer station busbar. The sensors are from ABB in Finland and are high bandwidth, compact, combined voltage and current.
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 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 16bit 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 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...
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.

2.3 Customers at the Feeder

A database at NESA was searched for information on the customers at A12. This database contains information on how much electricity each customer uses on a long term basis. Care has been taken to normalize the 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
seen from the figure that the period with the largest activity (the largest changes in the load) is between 5\textsuperscript{00} and 8\textsuperscript{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\textsuperscript{00}, 7\textsuperscript{00}, 10\textsuperscript{00}, 12\textsuperscript{00}, and 18\textsuperscript{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), 16bit precision, and a sampling frequency of 20kHz, 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, [Hög, 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 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, 1993] 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\textsuperscript{00} hours, so this supports the conclusion made on the rate.
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.

Figure 2.9: Detected transients of unknown origin.

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. This transformation is defined as [Lakervi and Holmes, 1989]

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

(3.1)

This is a frequency domain representation of the transformation, and is also most often used when considering sinusoidal excitation. A transformation to the time domain would require two all-pass filters with the transfer function \( e^{-j \frac{2\pi}{3}} \) and \( e^{-j \frac{4\pi}{3}} \). By applying these two filters to the
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 \(\gamma\) and the characteristic impedance \(Z_0\) are defined in Table 3.1 under Modal 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

\[
V_x = \frac{1}{2}(V_1 + Z_0 I_1)e^{-\gamma x} + \frac{1}{2}(V_1 - Z_0 I_1)e^{\gamma x} \tag{3.4}
\]

\[
I_x = \frac{1}{2} \frac{V_1}{Z_0} + I_1 \text{e}^{-\gamma x} + \frac{1}{2} \frac{V_1}{Z_0} - I_1 \text{e}^{\gamma x}
\]

or expressed in terms of hyperbolic functions

\[
V_x = V_1 \cosh(\gamma x) - Z_0 I_1 \sinh(\gamma x) \tag{3.5}
\]

\[
I_x = \frac{V_1}{Z_0} \sinh(\gamma x) + I_1 \cosh(\gamma x)
\]

For \(x = L\), this equation gives the relation between the voltage and current.

---

**Table 3.1**: Definition of transmission line parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary parameters</td>
<td></td>
</tr>
<tr>
<td>Series resistance per meter</td>
<td>(r)</td>
</tr>
<tr>
<td>Series inductance per meter</td>
<td>(i)</td>
</tr>
<tr>
<td>Shunt conductance per meter</td>
<td>(g)</td>
</tr>
<tr>
<td>Shunt capacitance per meter</td>
<td>(c)</td>
</tr>
<tr>
<td>Length of transmission line</td>
<td>(L)</td>
</tr>
<tr>
<td>Phase parameters</td>
<td></td>
</tr>
<tr>
<td>Series impedance</td>
<td>(z = r + j\omega l)</td>
</tr>
<tr>
<td>Shunt admittance</td>
<td>(y = g + j\omega c)</td>
</tr>
<tr>
<td>Total series impedance</td>
<td>(Z = zL)</td>
</tr>
<tr>
<td>Total shunt admittance</td>
<td>(Y = yL)</td>
</tr>
<tr>
<td>Modal parameters</td>
<td></td>
</tr>
<tr>
<td>Wave propagation constant</td>
<td>(\gamma = \sqrt{\frac{y}{z}})</td>
</tr>
<tr>
<td>Characteristic impedance</td>
<td>(Z_0 = \sqrt{\frac{2}{y}})</td>
</tr>
</tbody>
</table>
of the two ends of the cable in Figure 3.2.

\[ V_2 = V_1 \cosh(\gamma L) - Z_0 I_1 \sinh(\gamma L) \]
\[ I_2 = \frac{V_1}{Z_0} \sinh(\gamma L) - I_1 \cosh(\gamma L) \]  

(3.6)

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

\[ I_1 = V_1 \frac{1}{Z_0} \frac{1}{\tanh(\gamma L)} - V_2 \frac{1}{Z_0} \frac{1}{\sinh(\gamma L)} \]
\[ I_2 = -V_1 \frac{1}{Z_0} \frac{1}{\sinh(\gamma L)} + V_2 \frac{1}{Z_0} \frac{1}{\tanh(\gamma L)} \]  

(3.7)

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

\[ \frac{1}{\tanh(x)} = \tanh\left(\frac{x}{2}\right) + \frac{1}{\sinh(x)} \]  

(3.8)

and the characteristic impedance in two ways as

\[ Z_0 = \frac{Z}{\gamma L} = \frac{\gamma L}{Y} \]  

(3.9)

Using Equation 3.9 following equalities can be derived

\[ \frac{1}{Z_0 \sinh(\gamma L)} = \frac{1}{Z} \frac{\gamma L}{\sinh(\gamma L)} \]
\[ \frac{\tanh(\frac{\gamma L}{2})}{Z_0} = \frac{Y \tanh(\frac{\gamma L}{2})}{2} \]  

(3.10)

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

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

(3.11)

This is the admittance matrix for an ideal two wire transmission line with \textit{constant} distributed parameters. The term \textit{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 over 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 \( \gamma \).

### 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 \( \Pi \)-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

\[ \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{Z} + \frac{Y}{2} & -\frac{1}{Z} + \frac{Y}{2} \\ -\frac{1}{Z} - \frac{Y}{2} & \frac{1}{Z} - \frac{Y}{2} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \]  

(3.12)

If this matrix is compared to Equation 3.11 it is clear that for a given angular frequency \( \omega \) and corresponding \( Z, Y, \) and \( \gamma \) the \( \Pi \)-equivalent
valid if

\[ Z_H = Z \frac{\sinh(\gamma L)}{\gamma L} \quad \text{and} \quad Y_H = Y \frac{\tanh(\gamma L)}{\gamma L} \]

\[(3.13)\]

where \( Z_H \) and \( Y_H \) 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 \( H \)-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 8th 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/~jang/index2-e.html 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 1999 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 II-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 II-section includes the loss, but is generally only valid for a 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 II-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.
4.1.2 Parameters for the Cable Model

The cascaded II-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 II-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 mm² copper APB cable. The capacitance is constant 0.29 μF. This frequency dependence is a function of properties 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 assumption 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 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 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 it is a common reference point for all voltages in the model. When a number of II-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 II-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.
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, $R$, $L$, and $C$, are computed by CABLE CONSTANTS.

The current $i_{kR}$ going into the II-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

$$i_{kR} = C_g \frac{\partial v_{kR}}{\partial t} + C_p \frac{\partial (v_{kR} - v_{kS})}{\partial t} + C_p \frac{\partial (v_{kR} - v_{kT})}{\partial t} + i_{kmR}$$

$$= (C_g + 2 C_p) \frac{\partial v_{kR}}{\partial t} - C_p \frac{\partial v_{kS}}{\partial t} - C_p \frac{\partial v_{kT}}{\partial t} + i_{kmR} \quad (4.3)$$

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

$$C = 2 \begin{bmatrix} C_g + 2 C_p & -C_p & -C_p \\ -C_p & C_g + 2 C_p & -C_p \\ -C_p & -C_p & C_g + 2 C_p \end{bmatrix} \quad (4.4)$$

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

$$v_{kR} - v_{mR} = L_p \frac{\partial i_{kmR}}{\partial t} + L_m \frac{\partial i_{kmS}}{\partial t} + L_m \frac{\partial i_{kmT}}{\partial t}$$

$$+ R_p i_{kmR} + R_m i_{kmS} + R_m i_{kmT} \quad (4.5)$$

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

\[
L = \begin{bmatrix}
L_p & L_m & L_m \\
L_m & L_p & L_m \\
L_m & L_m & L_p
\end{bmatrix} \quad R = \begin{bmatrix}
R_p & R_m & R_m \\
R_m & R_p & R_m \\
R_m & R_m & R_p
\end{bmatrix}
\] (4.6)

### 4.1.5 II-section with Ground Resistance

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

4.1 Cable Model

Now the current and voltage vectors are given by

\[
\begin{align*}
\mathbf{i}_k &= \begin{bmatrix} i_{kR} \\ i_{kS} \\ i_{kT} \end{bmatrix}, \quad \mathbf{i}_m = \begin{bmatrix} i_{mR} \\ i_{mS} \\ i_{mT} \end{bmatrix}, \quad \mathbf{i}_{km} = \begin{bmatrix} i_{kmR} \\ i_{kmS} \\ i_{kmT} \end{bmatrix} \\
\mathbf{v}_k &= \begin{bmatrix} v_{kR} \\ v_{kS} \\ v_{kT} \end{bmatrix}, \quad \mathbf{v}_m = \begin{bmatrix} v_{mR} \\ v_{mS} \\ v_{mT} \end{bmatrix}
\end{align*}
\] (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

\[
i_{kR} = C_g \frac{\partial (v_{kR} - v_{kG})}{\partial t} + C_p \frac{\partial (v_{kR} - v_{kS})}{\partial t} + C_p \frac{\partial (v_{kR} - v_{kT})}{\partial t} + i_{kmR}
\] (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 \( i_{kmG} \):

\[
i_{kG} = C_g \frac{\partial (v_{kG} - v_{kR})}{\partial t} + C_g \frac{\partial (v_{kG} - v_{kS})}{\partial t} + C_g \frac{\partial (v_{kG} - v_{kT})}{\partial t} + i_{kmG}
\]

\[
i_{kG} = -C_g \frac{\partial v_{kR}}{\partial t} - C_g \frac{\partial v_{kS}}{\partial t} - C_g \frac{\partial v_{kT}}{\partial t} + 3 C_g \frac{\partial v_{kG}}{\partial t} + i_{kmG}
\] (4.9)

If these expressions are compared to Equation 4.1 the matrix \( C' \) can be
identified as
\[
\mathbf{C}' = 2 \begin{bmatrix}
C_g + 2C_p & -C_p & -C_p & -C_p \\
-C_p & C_g + 2C_p & -C_p & -C_p \\
-C_p & -C_p & C_g + 2C_p & -C_p \\
-C_g & -C_g & -C_g & 3C_g
\end{bmatrix}
\] (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
\[
\mathbf{C}' = \begin{bmatrix}
\mathbf{C} & -2C_g \\
-2C_g & -2C_g \\
-2C_g & -2C_g & -2C_g & 6C_g
\end{bmatrix}
\] (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
\[
\mathbf{L}' = \begin{bmatrix}
\mathbf{L} & 0 \\
0 & 0 & 0
\end{bmatrix}, \quad \mathbf{R}' = \begin{bmatrix}
\mathbf{R} & 0 \\
0 & 0 & 0 & R_g
\end{bmatrix}
\] (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
\[
C_g = \frac{c_{11} + 2c_{21}}{2}
\] (4.13)

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

### 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. 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 lower RMS voltage.

<table>
<thead>
<tr>
<th>Open circuit test</th>
<th>Short circuit test</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_e )</td>
<td>Excitation loss</td>
</tr>
<tr>
<td>( I_e )</td>
<td>Excitation current</td>
</tr>
<tr>
<td>( U_e )</td>
<td>Excitation voltage</td>
</tr>
<tr>
<td>( P_s )</td>
<td>Short circuit loss</td>
</tr>
<tr>
<td>( I_s )</td>
<td>Short circuit current</td>
</tr>
<tr>
<td>( U_s )</td>
<td>Short circuit voltage</td>
</tr>
</tbody>
</table>

**Table 4.1: Transformer test variables**
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

\[ S_N = 3U_N I_N \]  

(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

\[ U_p = \sqrt{3} U_l \]  

(4.15)

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

\[ S_N = \sqrt{3} U_N I_N \]  

(4.16)

Rearranging this expression, the nominal current can be written as

\[ I_N = \frac{S_N}{\sqrt{3} U_N} \]  

(4.17)

Table 4.2 lists the variables needed to run the ATP supporting routine *BCTRAN*. \( IEXPOS \) is the normalized excitation current for open circuit test at nominal voltage, so it is defined as

\[ IEXPOS = I_o \cdot 100\% = \frac{\sqrt{3} U_N I_o}{S_N} \cdot 100\% \]  

(4.18)

4.2 Transformer Models

Table 4.2: *BCTRAN* input variables

<table>
<thead>
<tr>
<th>ATP variable</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEXPOS</td>
<td>Normalized excitation current for open circuit test</td>
<td>%</td>
</tr>
<tr>
<td>SPOS</td>
<td>Power base</td>
<td>kVA</td>
</tr>
<tr>
<td>LEXPOS</td>
<td>Normalized open circuit loss</td>
<td>kW</td>
</tr>
<tr>
<td>P12</td>
<td>Normalized short circuit loss</td>
<td>kW</td>
</tr>
<tr>
<td>ZPOS12</td>
<td>Normalized short circuit impedance</td>
<td>%</td>
</tr>
</tbody>
</table>

\( IEXPOS \) is the active power at nominal voltage so it is defined as

\[ IEXPOS = P_o \frac{U_N}{U_o} \]  

(4.19)

\( P12 \) is the short circuit active power at nominal current.

\[ P12 = P_s \frac{I_N}{I_s} = \frac{P_s S_N}{\sqrt{3} U_N I_s} \]  

(4.20)

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

\[ ZPOS12 = \frac{Z_s}{\sqrt{3} U_N I_s} \cdot 100\% \]  

(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...
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_{m,o}$</td>
<td>Open circuit active power</td>
<td>0.648</td>
<td>kW</td>
</tr>
<tr>
<td>$P_{m,o}$</td>
<td>Open circuit reactive power</td>
<td>1.74</td>
<td>kVA</td>
</tr>
<tr>
<td>$P_{m,s}$</td>
<td>Short circuit active power</td>
<td>1.55</td>
<td>%</td>
</tr>
<tr>
<td>$S_{m,s}$</td>
<td>Short circuit apparent power</td>
<td>4.58</td>
<td>%</td>
</tr>
</tbody>
</table>

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.

$$S_{m,o} = \sqrt{P_{m,o}^2 + Q_{m,o}^2} = \sqrt{3}U_N I_{m,o}$$  \hspace{1cm} (4.22)

This is used in Equation 4.18 and IEXPOS is

$$IEXPOS = \frac{\sqrt{P_{m,o}^2 + Q_{m,o}^2}}{S_N} \cdot 100 \%$$

$$= \frac{\sqrt{0.648 \text{ kV} \cdot 1.74 \text{kVar}}}{400 \text{kVA}} \cdot 100 \% = 0.464 \% \hspace{1cm} (4.23)$$

The active power is given directly so

$$LEXPOS = P_{m,o} = 0.648 \text{kW} \hspace{1cm} (4.24)$$

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

$$P_{12} = \frac{P_{m,s} S_N}{100 \%} = \frac{400 \text{kVA} \cdot 1.55 \%}{100 \%} = 6.20 \text{kW} \hspace{1cm} (4.25)$$

The short circuit impedance $Z_{POS12}$ 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.

$$Z_{POS12} = \frac{U_s I_N}{I_N U_N} \cdot 100 \% = \frac{S_N}{S_N} \cdot 100 \% = S_{m,s} = 4.58 \% \hspace{1cm} (4.26)$$

### 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 \textit{arc suppression coil}. The Petersen coil is connected to the star point of 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

$$L_P = \frac{1}{\omega^2 C_T}, \hspace{1cm} \omega = 2\pi 50 \text{ Hz}$$  \hspace{1cm} (4.27)
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 Pi-sections with ground resistance, each Pi-section representing 40 m of cable. Furthermore, 20 loads and a Petersen coil is included in the model.

4.5 Impulse Response Generation

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 or 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:

<table>
<thead>
<tr>
<th>Terminal output</th>
</tr>
</thead>
<tbody>
<tr>
<td>MakeNet version 3.5</td>
</tr>
<tr>
<td>Totals are:</td>
</tr>
<tr>
<td>length of network : 8880 m.</td>
</tr>
<tr>
<td>number of Pi sections : 212</td>
</tr>
<tr>
<td>symmetrical resistance : 2.22 Ω</td>
</tr>
<tr>
<td>symmetrical inductance : 2.23 mH</td>
</tr>
<tr>
<td>symmetrical capacitance : 2.36 μF</td>
</tr>
<tr>
<td>zero resistance : 11.4 Ω</td>
</tr>
<tr>
<td>zero inductance : 41.7 mH</td>
</tr>
<tr>
<td>zero capacitance : 1.05 μF</td>
</tr>
</tbody>
</table>

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

\[
L_P = \frac{1}{(100\pi)^2 \cdot 1.05 \mu F} = 9650 \text{ mH}
\]

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, 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
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.

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 II-sections in the network model. This ground is not the global reference node TERRA but the ground node contained in the II-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...
Numerical Simulations of Network Signals

here consists of cables so the first two issues deals with the II-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 \( Y \) that gives the relation between the voltages and currents for this two port is given by

\[
\begin{bmatrix}
I_1 \\
I_2
\end{bmatrix} = Y \begin{bmatrix}
V_1 \\
V_2
\end{bmatrix}
\]

(4.29)

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

\[
T_d = \begin{bmatrix}
\frac{Z}{\tan(h/\gamma L)} + \frac{\gamma L}{2} & \frac{\gamma L}{2} \\
\frac{\gamma L}{2} & \frac{\gamma L}{\tan(h/\gamma L) + \frac{\gamma L}{2}}
\end{bmatrix}
\]

(4.30)

where \( Z \) is the series impedance, \( Y \) is the shunt admittance, \( \gamma \) 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

\[
\gamma = \sqrt{(r + j\omega)(g + j\omega c)}
\]

(4.31)

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

\[
T_l = \begin{bmatrix}
\frac{1}{Z} + \frac{Y}{2} & -\frac{1}{Y} \\
-\frac{1}{Z} & \frac{1}{Z} + \frac{Y}{2}
\end{bmatrix}
\]

(4.32)

It is obvious that these two matrices are equal if the hyperbolic sine and tangent terms in \( T_d \) are equal to 1. To get an estimate of the frequency range for which this is true, let \( \varepsilon \) be some small number defined by

\[
\varepsilon > \frac{\tanh(x)}{x} - 1 \quad \text{and} \quad \varepsilon > 1 - \frac{x}{\sinh(x)}
\]

(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

\[
\varepsilon > \frac{\tanh(x)}{x} - 1 = \frac{x + \frac{x^3}{3} + \frac{x^5}{5} + \cdots}{x(1 + \frac{x^2}{2} + \frac{x^4}{4} + \cdots)} - 1
\]

\[
\approx -\frac{2x^2}{6 + 3x^2}
\]

(4.34)

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

\[
\gamma^2 = -\omega^2 L c, \quad \omega = 2\pi f
\]

(4.35)

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

\[
f < \frac{1}{2\pi L} \sqrt{\frac{6\varepsilon}{(2 + 3\varepsilon)c}} \approx \frac{1}{\pi L} \sqrt{\frac{3\varepsilon}{L c}}, \quad \varepsilon \ll 1
\]

(4.36)

Similarly the second inequality of Equation 4.33 can be approximated by

\[
\varepsilon > 1 - \frac{x}{\sinh(x)} = 1 - \frac{x}{x + \frac{x^3}{6} + \cdots} \approx \frac{x^2}{6 + x^2}
\]

(4.37)

Again by rearranging Equation 4.37, substituting \( x \) with \( \gamma L \), and \( \gamma^2 \) from Equation 4.35, the frequency limit corresponding to the second inequality of Equation 4.33 can be approximated by

\[
f < \frac{1}{2\pi L} \sqrt{\frac{6\varepsilon}{(1 - \varepsilon)c}} \approx \frac{1}{2\pi L} \sqrt{\frac{6\varepsilon}{L c}}, \quad \varepsilon \ll 1
\]

(4.38)
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

\[ f_{\text{lim}} = \frac{1}{2\pi L} \sqrt{\frac{6\varepsilon}{lc}}, \quad \varepsilon \ll 1 \]  

(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 mm² 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 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 \( Z_{\text{in}} \), as shown in the figure. The transfer function of the model in Figure 4.12 is given by

\[ H(j\omega) = \frac{Z_{\text{in}}}{Z + Z_{\text{in}}(1 + YZ)} \]  

(4.40)

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

\[ Z_{\text{in}} = \frac{Z}{2} + \sqrt{\frac{Z^2}{4} + \frac{Z}{Y}} \]  

(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 mm² 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.
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 to 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 0Hz to 5kHz. First ensure that no signal on the model contains frequency components higher than 5kHz. To avoid aliasing, the sampling frequency of the final signal must be 10kHz. This is the Nyquist frequency. To get an error less than 1% in the ATP simulations, the step frequency must be 20 times larger than the required signal bandwidth at 5kHz, which is 100kHz.

### 4.7 Precision of ATP Simulations

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 25kHz. 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 6.7·10$^{-8}$ 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 0–25kHz.

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. Furthermore, 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 Ω 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 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.
Ground Fault Localization

illustrates how the impulse response is generated. A single phase impulse current source is connected between location \( i \) and ground, and the impulse response \( h_i \) is measured on the same phase at the observation point \( O \) by the substation busbar. This means that \( h_i \) is the impulse response of the network from the \( i \)th grid point to the observation point. The impulse response \( h_i \) 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

\[
h_i = [h_i(0) h_i(T) \ldots h_i(mT) \ldots h_i((M-1)T)]^T \tag{5.1}
\]

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

\[
H = [h_0 h_1 \ldots h_i \ldots h_{N-1}] \tag{5.2}
\]

where \( h_i \) is the \( i \)th column in an \( M \) by \( N \) matrix. The length of an impulse response for the A/1/2 feeder described in Section 4.4 is a few milliseconds, so if 3ms 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 40m between the grid points, so in that case \( H \) 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 \( y_F \).

If \( x_F \) 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

\[
y_F = x_F * h_F \tag{5.3}
\]

where \( F \) is the unknown index that we want to find, and \( y_F \) is the transient measured at the observation point. Both \( x_F \) and \( y_F \) are \( M \) length column vectors defined analogous to \( h_i \) in Equation 5.1. An estimate of the transient at the fault point can therefore be found by deconvolution as

\[
\hat{x}_{F,i} = y_F * h_i^{-1}, \quad i = 0, \ldots, N-1 \tag{5.4}
\]

where \( h_i^{-1} \) is the inverse impulse response defined by

\[
h_i * h_i^{-1} = \delta \tag{5.5}
\]

The impulse vector \( \delta \) 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

\[
\hat{x}_{F,i} = x_F * h_F * h_i^{-1}, \quad i = 0, \ldots, N-1 \tag{5.6}
\]

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,

\[
e_{F,i} = h_F * h_i^{-1}, \quad i = 0, \ldots, N-1 \tag{5.7}
\]

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

\[
e_{F,F} = \delta \tag{5.8}
\]

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.
5.4 Deconvolution in the Frequency Domain

To describe this, Figure 5.2 shows six deconvolutions of the same ground fault transient $y_F$. 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 $x_{F,i}$ 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 filter step function with some high frequency content. The high-pass filter is applied to the measurement $y_F$ and the step function is due to the switch that is used to model the ground fault. The high frequency content 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 $h_i^{-1}$ 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

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

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

\[ H_i(k) = \sum_{m=0}^{M-1} h_i(mT) e^{-j\frac{2\pi km}{M}}, \quad k = 0, \ldots, M-1 \]  

(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 \( 0 \leq k < \frac{M}{2} \).

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 \( h_i(t) \).

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

\[ Y_F(k) = X_F(k) H_F(k) \]  

(5.12)

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

\[ X_{F,i}(k) = \frac{Y_F(k)}{H_i(k)} \]  

(5.13)

and the spectrum of the mismatch function \( e_{F,i} \) in Equation 5.7 as

\[ E_{F,i}(k) = \frac{H_F(k)}{H_i(k)} \]  

(5.14)

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

\[ E_{F,i}(k) = \frac{Y_F(k)}{H_i(k) X_F(k)} \]  

(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

\[ E_{F,F}(k) = 1, \quad \text{for all } k \]  

(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 \( x_F \) may be derived directly from the observed transient \( y_F \). 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 \( y_F \) and using the difference as the size of a step function. If this difference is called \( \Delta_F \), it can be expressed as

\[ \Delta_F = \max_m(y_F(mT)) - \min_m(y_F(mT)), \quad m = 0, \ldots, M-1 \]  

(5.17)

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

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

(5.18)

where \( u(t) \) is the unit step function, and \( m_F \) is a delay that aligns the step with the ground fault transient \( y_F \). Index \( F \) of \( g_F \) denotes that this model is derived from the measured transient \( y_F \).

Figure 5.3 shows a simulation of the ground fault transient \( y_F \) 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 \( y_F \) and shown in Figure 5.4. Appendix D.4 gives a Matlab function which performs this task.
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

$$E_{F,i}(k) = \frac{Y_F(k)}{H_i(k)G_F(k)} \quad k = 0, \ldots, \frac{M}{2} - 1$$

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

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, $\hat{X}_i, \hat{X}_f, \hat{X}_m,$ and $\hat{X}_p$ (the hat is omitted in the legend of the figure). For comparison, $G_F$ is shown together with the deconvolutions.

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

$$E_{F,i} = \frac{1}{\log_{10} 2} \sum_{k=0}^{\frac{M}{2} - 1} \left| \frac{Y_F(k)}{H_i(k)G_F(k)} \right|$$

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 $E_{F,F}$ of zero. This means that the minimum value of the error measure in Equation 5.20 is an estimate of the fault location $F$.

$$\hat{F} = \left\{ \hat{i} \mid \min_i (E_{F,i}) \right\}$$
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, 1993] as a normal daily 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 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, 1993] as a normal daily 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.
The error measure $E_{F,i}$ 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 $E_{F,i}$. 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 $E_{F,i}$ 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 fault location for all $N$ ground fault simulations. Each point on the curve is 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.
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 alphabetical 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 where 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...
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 is a dark blue diagonal line from the lower left corner to the upper right corner 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 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 resistance $R_F$, 1 Ω, 100 Ω, and 10 kΩ. For each
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 $\Omega$ 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 $\Omega$ and a ground fault of 1 $\Omega$ (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.
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.
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 are 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...
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 were 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 were 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 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 improvements.

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.
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 synchronize 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 contains 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 200 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 can influence 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 zero 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 the 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 $2 \times 19.6 \, \text{k\Omega}$ and $2 \times 1 \, \text{k\Omega}$. The 19.6 k\Omega resistors are in fact series connections of four 4.9 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 is 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 k\Omega resistors and in the right side of the picture the two 1 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 synchronization and timing equipment is seen, in the middle the compressor and in the right side the generator is seen. Behind the synchronization unit is the HZA breaker.
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, there are three voltage probes 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 plastic 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 PEARSE 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
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.

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.
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 kΩ, 0.5 kΩ, 1 kΩ, 2 kΩ, and 20 kΩ, and the closing angle was varied in six different steps: 0°, 30°, 60°, 90°, 120°, and 150°. At location 1 the above range of resistances was expanded with a fault resistance of 10 kΩ. 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 100kHz 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° is an exception because, as one might expect, the fault is clearly seen in the 50Hz time signal but nothing is seen in the spectrum above 2kHz. 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
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 0Ω 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 $u_0$.

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 2kHz. 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 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 transient also computed using a high-pass filter with a cutoff frequency of 2kHz. 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 2ms for a signal bandwidth of 2–50 kHz.

Note that the transient $i_T$ in 6.17 is the signal referred to as the measured transient $y_T$ in Chapter 5. The spectrum of the voltage and current signals are computed by taking 10ms 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 6.20: 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

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 50kHz (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 spectrums.

The result is shown in Figure 6.18 and 6.19. The absolute level of the vertical axis in decibel is relative to 1V and 1A 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 15–20kHz whereas the voltage signals are well defined up to 25kHz. It may also be that the current signals does not have a frequency contents within the dynamic range for frequencies above 15kHz. 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 20kHz all three 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 10–20kHz, 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 50m apart and their spectras are also very much alike, so these two locations would be difficult to distinguish in the

---

\(^1\)A Hanning window or a raised cosine is defined by

\[ h_n = \frac{1}{2} \left(1 - \cos \left(2\pi \frac{n-0.5}{N} \right)\right), \quad n = 0, \ldots, N - 1 \]
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 Section 5.5. The figure shows that the two spectras have the same general shape (−20dB/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 2kHz and for 20kHz. 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 18 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 6.28: Voltage, location 7, configuration 2.

Figure 6.29: Current, location 7, configuration 2.

6.6 Acquired Data

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 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 II-section 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 database. 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 II-sections. A method for inclusion of a distributed ground resistance (impedance) in these II-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 on 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 trainin
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.

7.2 Suggestions for Future Work

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 II-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 II-section as described in Chapter 7.1 on page 21. The problem with the ground fault experiment data in this respect is that the network at the test facility is composed of loops comi
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 60-66dB (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.

7.2 Suggestions for Future Work

The experimental data show in some cases an extensive noise component around 2 kHz. This may be caused by serial resonance circuits 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, \( y \) that is defined by

\[
y = x * h
\]

where vector \( y \) and \( h \) are known, and that we want to find \( x \) by deconvolution. This cannot 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 \( x \). If we have an initial guess \( x_0 \) an update of the estimate, \( \hat{x}_i \) can be found by applying the method of steepest descent [Haykin, 1996]

\[
\hat{x}_i = \hat{x}_{i-1} - \lambda \frac{\partial \| y - x_{i-1} * h \|^2}{\partial x_{i-1}}
\]

where \( \lambda \) 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 \( \lambda \).
If we define an error vector as
\[ \mathbf{e} = \mathbf{y} - \mathbf{x} \ast \mathbf{h} \] (A.3)
we have that the squared error can be written as
\[ \mathbf{e}^T \mathbf{e} = ||\mathbf{y} - \mathbf{x} \ast \mathbf{h}||^2 \] (A.4)

The gradient to the squared error with respect to \( \mathbf{x} \) can then be written as
\[ \frac{\partial \mathbf{e}^T \mathbf{e}}{\partial \mathbf{x}} = -2(\mathbf{h} \ast \mathbf{e}^{rev})^{rev} \\
= -2(\mathbf{h} \ast (\mathbf{y} - \mathbf{x} \ast \mathbf{h})^{rev})^{rev} \] (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 \( \mathbf{x}^{rev} = [x_4 \ x_3 \ x_2 \ x_1] \).

If Equation A.5 is inserted into Equation A.2 the iteration update is found as
\[ \hat{x}_i = \hat{x}_{i-1} + 2\lambda(\mathbf{h} \ast (\mathbf{y} - \mathbf{x}_{i-1} \ast \mathbf{h})^{rev})^{rev} \] (A.6)

All that is needed by Equation A.6 is an initial guess for \( \mathbf{x}, \mathbf{x}_0 \) and a step size \( \lambda \). 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 \(-20\,\text{dB} \) per decade and a constant argument of \(-90^\circ \). 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.
\[ y(n) = \frac{1}{L} \sum_{k=0}^{L-1} x(n-k) \]
\[ = y(n-1) + \frac{1}{L}[x(n) - x(n-L)] \quad (A.7) \]

The Z-transform of Equation A.7 gives the transfer function in Equation A.8.
\[ Y(z) = \frac{1}{L} X(z) (z^{-L-1} + \ldots + z^{-2} + z^{-1} + 1) \]
\[ = Y(z) z^{-L-1} + X(z) \frac{1}{L} [1 - z^{-L}] \quad (A.8) \]
\[ H_1(z) = \frac{1}{L} \frac{1 - z^{-L}}{1 - z^{-1}} \]

The frequency response of Equation A.8 can be found by substituting \( z \) with \( e^{j2\pi fT} \). This gives the expression in Equation A.9 where \( T \) is the sampling time.
\[ H_1(e^{j2\pi fT}) = \frac{1}{L} \frac{1 - e^{-j2\pi fLT}}{1 - e^{-j2\pi fT}} \]
\[ = \frac{1}{L} \frac{e^{j\pi fLT}}{e^{j\pi fT}} - e^{-j\pi fLT} e^{-j\pi fT} \\
= \frac{1}{L} \frac{\sin(\pi fLT)}{\sin(\pi fT)} e^{-j\pi f(L-1)T} \quad (A.9) \]

Figure A.1 shows the magnitude and argument of \( H_1(e^{j2\pi fT}) \) 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 50Hz and at all
higher harmonics. The argument seems to be linear with jumps from $-180^\circ$ to $0^\circ$ 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:

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

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

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

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

$$\hat{Y}(z) = \hat{Y}(z)z^{-1} + X(z)$$
$$\hat{Y}(z) = \frac{1}{1 - z^{-1}}X(z) \quad (A.12)$$

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

$$Y(z) = \hat{Y}(z) - \frac{1}{L} \hat{Y}(z)[1 + z^{-1} + z^{-2} + \ldots + z^{-(L-1)}]$$
$$= \hat{Y}(z) \left\{ 1 - \frac{1}{L} \left[ 1 + z^{-1} + z^{-2} + \ldots + z^{-(L-1)} \right] \right\} \quad (A.13)$$

and the overall Z-transform can be found by combining Equation A.11 and A.12.
A.2 Design of Digital Integrator

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.

\[ H(s) = \frac{\frac{s}{\omega_0}}{1 + \frac{s}{\omega_0}^2} \]  \( \omega_0 = 2\pi f_0 \)  \hspace{1em} (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.

\[ H(z) = \frac{\omega_0}{(\omega_0 + 1)^2} \left( 1 - \frac{1}{z + 1} \right) \left( 1 - \frac{1}{\omega_0 + 1} + \frac{1 - z^{-2}}{(\omega_0 + 1)^2} z^{-2} \right) \]  \hspace{1em} (A.16)

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

\[ y(n) = \frac{\omega_0}{(\omega_0 + 1)^2} (x(n) - x(n - 2)) - \frac{\omega_0 - 1}{\omega_0 + 1} y(n - 1) - \left( \frac{\omega_0 - 1}{\omega_0 + 1} \right)^2 y(n - 2) \]  \hspace{1em} (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 \( \frac{d\varphi}{df} \) where \( \varphi \) is the argument of \( H_3(e^{j2\pi fT}) \) in Figure A.4. An approximation to this is calculated numerically as given.
by Equation A.18 and the result is shown in Figure A.5.

\[
d(n) = \frac{\varphi(n) - \varphi(n-1)}{2\pi (f(n) - f(n-1))}
\]

where \( \varphi(n) = \text{arg}(H_3(e^{2\pi j f(n)T})) \) (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 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 -20dB 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 \(-90^\circ\) 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 ex
actually −20 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 $T = 0.05$ 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 end of the model the input impedance $Z_{\text{in}}$ for the section must be equal to the load impedance as shown by Figure B.1.

The impedance $Z_{\text{in}}$ can be found by solving following equation for $Z_{\text{in}}$:

$$Z_{\text{in}} = Z + \frac{1}{Y} \parallel Z_{\text{in}} = Z + \frac{1}{Y + Z_{\text{in}}} = Z + \frac{Z_{\text{in}}}{1 + YZ_{\text{in}}}$$

$$Z_{\text{in}} = \frac{Z(1 + YZ_{\text{in}}) + Z_{\text{in}}}{1 + YZ_{\text{in}}} = \frac{Z + (ZY + 1)Z_{\text{in}}}{1 + YZ_{\text{in}}}$$

$$\Leftrightarrow Z_{\text{in}}(1 + YZ_{\text{in}}) = Z + (ZY + 1)Z_{\text{in}}$$

$$\Leftrightarrow Z_{\text{in}}^2 - ZZ_{\text{in}} - \frac{Z}{Y} = 0$$

$$\Rightarrow Z_{\text{in}} = \frac{-(-1) \pm \sqrt{(-1)^2 - 4 \cdot 1 \cdot (-\frac{Z}{Y})}}{2} = \frac{Z \pm \sqrt{Z^2 + \frac{Z}{Y}}}{2}$$
Transfer Function for II-section

\[ H(j\omega) = \frac{V_2}{V_1} = \frac{Z + \frac{Z_{in}}{1 + Y_{\infty}}} {Z + \frac{Z_{in}}{1 + Y_{\infty}}} = \frac{\frac{Z_{in}}{1 + Y_{\infty}}}{\frac{Z_{in}}{1 + Y_{\infty}}} \]

where

\[ Z = L(r + j\omega) \]
\[ Y = L(g + j\omega) \]  \hspace{1cm} (B.2)

When the length \( L \) of the cable that the II-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

\[ \lim_{L \to 0} Z_{in} = \sqrt{\frac{Z}{Y}} \] \hspace{1cm} (B.3)

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

\[ H(j\omega) = \frac{V_2}{V_1} = \frac{1}{Z + \frac{Y_{\infty}}{1 + Y_{\infty}}} = \frac{\frac{Z_{in}}{1 + Y_{\infty}}}{Z + \frac{Z_{in}}{1 + Y_{\infty}}} \]

\[ = \frac{1}{Z(1 + Y_{\infty}) + Z_{in}} = \frac{Z \cdot \frac{Z_{in}}{1 + Y_{\infty}}}{Z + Z_{in}(1 + Y)} \] \hspace{1cm} (B.4)

Figure B.1: Model of II-equivalent.

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 II-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 II section. This frequency is computed for different lengths \( L \) of the II-section and 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:

```plaintext
$INCLUDE, iai2.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 iai2.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 iai2.atp for the A12 feeder is listed below:

```plaintext
KARD 6 7
KARG 1 1
KBGD 3 3
KEND 7 7
KTEX 1 1
BEGIN NEW DATA CASE
POWER FREQUENCY 50.00
C DELT < TMAX < EPSIL < CULMAT < TSTART
2.0E-06 0.004
C Source: low-pass filtered impulse source at phase r
C Node: defined by $INCLUDE parameter.
C Direction: from phase to ground.
/SOURCE
C (n b) < start > < stop >
LMOD_R-1
LMOD_G-1
$INCLUDE, /phd/kjn/matlab/downsmpl/h10.dat
/SWITCH
C (n b) < n z > < Tclose > < Top/Tde > < Le > < VL/CLDP > < type >
GLN1_SIA000S
```

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. The iai2.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 10th of the half sampling rate ($\frac{1}{2}$). 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 included 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 Ohms 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.

c/./4 No de 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

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 symmetric.

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 lowercase, 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 II-sections, so editing the network input file by hand is not a good solution. Therefore a program makenet was written to generate the input files. Appendix E.1 on page 164 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 II-section as described in Section 4.1.5 on page 28.

SetPiLength: The length of each II-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 II-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 cable will be connected (with the newcablesession 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 \( \cos(\phi) \). 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 II-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.

```plaintext
% Definition of the A12 feeder at Glentegaarden
SetGroundResistance 0.000625
SetPiLength 40
% Use cable parameters computed at 10kHz
SetCableDir /phd/kjm/matlab/cabledef/10.0000
DefineCable a0a.lis Al240PEX A2P
DefineCable a0a.lis Al240APB A2A
DefineCable a5a.lis Al150PEX A1P
DefineCable a5a.lis Al150PEX A1S
DefineCable c5a.lis Cu60APB C1S
DefineCable c5a.lis Cu50APB C9S
DefineCable c5a.lis Cu50APB C9S
```
In line 2 the resistance in the ground phase is set to 0.625 mΩ/m and in line 3 the length of the II-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 mm² aluminum core.

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 II-section of this cable subsection. Line 29 makes a branch at the junction of cable subsection CB and DA.

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 a 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 experiment 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 a12.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
### ATP Network Models

<table>
<thead>
<tr>
<th>Section</th>
<th>Name</th>
<th>Length (m)</th>
<th>Cable Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>636-1237</td>
<td>AA</td>
<td>70</td>
<td>Al 240 PEX</td>
</tr>
<tr>
<td>636-1237</td>
<td>AB</td>
<td>654</td>
<td>Cu 65 APB</td>
</tr>
<tr>
<td>636-1237</td>
<td>AC</td>
<td>114</td>
<td>Cu 150 APB</td>
</tr>
<tr>
<td>636-1237</td>
<td>AD</td>
<td>363</td>
<td>Cu 65 APB</td>
</tr>
<tr>
<td>1927-1645</td>
<td>BA</td>
<td>246</td>
<td>Cu 65 APB</td>
</tr>
<tr>
<td>1927-1645</td>
<td>BB</td>
<td>181</td>
<td>Al 150 PEX</td>
</tr>
<tr>
<td>1964-1997</td>
<td>CA</td>
<td>181</td>
<td>Al 150 PEX</td>
</tr>
<tr>
<td>1964-1997</td>
<td>CB</td>
<td>306</td>
<td>Al 150 APB</td>
</tr>
<tr>
<td>1979-0992</td>
<td>DA</td>
<td>31</td>
<td>Al 150 APB</td>
</tr>
<tr>
<td>1979-0992</td>
<td>DB</td>
<td>17</td>
<td>Cu 65 APB</td>
</tr>
<tr>
<td>1979-0992</td>
<td>DC</td>
<td>52</td>
<td>Al 150 APB</td>
</tr>
<tr>
<td>0992-1974</td>
<td>EB</td>
<td>10</td>
<td>Cu 150 APB</td>
</tr>
<tr>
<td>0992-1974</td>
<td>EC</td>
<td>821</td>
<td>Cu 65 APB</td>
</tr>
<tr>
<td>1974-3719</td>
<td>FA</td>
<td>260</td>
<td>Cu 65 APB</td>
</tr>
<tr>
<td>3719-1926</td>
<td>FB</td>
<td>73</td>
<td>Al 65 APB</td>
</tr>
<tr>
<td>1928-2090</td>
<td>GA</td>
<td>71</td>
<td>Al 65 APB</td>
</tr>
<tr>
<td>2090-1106</td>
<td>GB</td>
<td>457</td>
<td>Cu 60 APB</td>
</tr>
<tr>
<td>1105-3408</td>
<td>IA</td>
<td>215</td>
<td>Cu 60 APB</td>
</tr>
<tr>
<td>1105-3408</td>
<td>JB</td>
<td>57</td>
<td>Al 65 APB</td>
</tr>
<tr>
<td>5408-0925</td>
<td>KA</td>
<td>58</td>
<td>Al 65 APB</td>
</tr>
<tr>
<td>0925-0633</td>
<td>KB</td>
<td>267</td>
<td>Cu 50 APB</td>
</tr>
<tr>
<td>0925-0633</td>
<td>LA</td>
<td>110</td>
<td>Cu 60 APB</td>
</tr>
<tr>
<td>0925-0633</td>
<td>LB</td>
<td>7</td>
<td>Al 65 APB</td>
</tr>
<tr>
<td>0925-0633</td>
<td>LC</td>
<td>53</td>
<td>Cu 50 APB</td>
</tr>
<tr>
<td>0925-0633</td>
<td>LD</td>
<td>6</td>
<td>Al 65 APB</td>
</tr>
<tr>
<td>0925-0633</td>
<td>LE</td>
<td>124</td>
<td>Cu 50 APB</td>
</tr>
<tr>
<td>0925-0633</td>
<td>LF</td>
<td>4</td>
<td>Al 65 APB</td>
</tr>
<tr>
<td>0925-0633</td>
<td>LG</td>
<td>51</td>
<td>Cu 50 APB</td>
</tr>
<tr>
<td>1974-3138</td>
<td>MA</td>
<td>476</td>
<td>Cu 65 APB</td>
</tr>
<tr>
<td>518-2312</td>
<td>NA</td>
<td>481</td>
<td>Cu 65 APB</td>
</tr>
<tr>
<td>2162-2841</td>
<td>OA</td>
<td>458</td>
<td>Cu 65 APB</td>
</tr>
<tr>
<td>1979-0648</td>
<td>PA</td>
<td>88</td>
<td>Al 65 APB</td>
</tr>
<tr>
<td>1979-0648</td>
<td>PB</td>
<td>452</td>
<td>Cu 50 APB</td>
</tr>
<tr>
<td>0648-2074</td>
<td>QA</td>
<td>181</td>
<td>Cu 60 APB</td>
</tr>
<tr>
<td>0648-2074</td>
<td>QB</td>
<td>23</td>
<td>Al 65 APB</td>
</tr>
<tr>
<td>2074-1090</td>
<td>RA</td>
<td>22</td>
<td>Al 65 APB</td>
</tr>
<tr>
<td>2074-1090</td>
<td>RB</td>
<td>178</td>
<td>Cu 50 APB</td>
</tr>
<tr>
<td>1090-1315</td>
<td>SA</td>
<td>599</td>
<td>Cu 60 APB</td>
</tr>
<tr>
<td>1090-1344</td>
<td>TA</td>
<td>360</td>
<td>Cu 60 APB</td>
</tr>
</tbody>
</table>

Table C.1: Cable data for A12.

### C.7 Conversion of the ATP output file

<table>
<thead>
<tr>
<th>Section</th>
<th>Name</th>
<th>Length (m)</th>
<th>Cable Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>6464-6466</td>
<td>AA</td>
<td>34</td>
<td>Al 95 PEX</td>
</tr>
<tr>
<td>6464-6466</td>
<td>AB</td>
<td>4</td>
<td>Al 150 APB</td>
</tr>
<tr>
<td>6464-6466</td>
<td>AC</td>
<td>381</td>
<td>Cu 50 APB</td>
</tr>
<tr>
<td>6464-6466</td>
<td>AD</td>
<td>965</td>
<td>Cu 25 OH</td>
</tr>
<tr>
<td>6464-6466</td>
<td>AE</td>
<td>340</td>
<td>Cu 35 OH</td>
</tr>
<tr>
<td>6465-6464</td>
<td>BA</td>
<td>308</td>
<td>Cu 35 OH</td>
</tr>
<tr>
<td>6465-6464</td>
<td>BB</td>
<td>270</td>
<td>Cu 50 OH</td>
</tr>
<tr>
<td>6465-6464</td>
<td>BC</td>
<td>900</td>
<td>Cu 50 OH</td>
</tr>
<tr>
<td>6465-6464</td>
<td>BD</td>
<td>250</td>
<td>Cu 50 APB</td>
</tr>
<tr>
<td>6465-6464</td>
<td>BE</td>
<td>193</td>
<td>Al 95 PEX</td>
</tr>
<tr>
<td>6464-6464</td>
<td>CA</td>
<td>181</td>
<td>Al 95 PEX</td>
</tr>
<tr>
<td>6464-6464</td>
<td>CB</td>
<td>30</td>
<td>Al 150 PEX</td>
</tr>
<tr>
<td>6464-6464</td>
<td>CC</td>
<td>612</td>
<td>Al 240 APB</td>
</tr>
<tr>
<td>6467-6464</td>
<td>DA</td>
<td>612</td>
<td>Cu 60 APB</td>
</tr>
<tr>
<td>6467-6464</td>
<td>DB</td>
<td>239</td>
<td>Al 95 PEX</td>
</tr>
<tr>
<td>6464-6464</td>
<td>EA</td>
<td>759</td>
<td>Al 240 PEX</td>
</tr>
<tr>
<td>6464-6464</td>
<td>EB</td>
<td>149</td>
<td>Al 95 PEX</td>
</tr>
<tr>
<td>6465-6465</td>
<td>FA</td>
<td>20</td>
<td>Cu 60 APB</td>
</tr>
<tr>
<td>6465-6465</td>
<td>FB</td>
<td>165</td>
<td>Al 95 PEX</td>
</tr>
<tr>
<td>6465-6465</td>
<td>FC</td>
<td>5</td>
<td>Al 150 APB</td>
</tr>
<tr>
<td>6465-6465</td>
<td>FD</td>
<td>635</td>
<td>Cu 95 APB</td>
</tr>
</tbody>
</table>

Table C.2: Cable data for Kyndby network.
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-pass filtered 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 $1.1is
4 rm $1.1is
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 5 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 background 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, therefore 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.

**GFERR**: implementation of the ground fault algorithm derived in Chapter 5. D.2, 131
**SHOWERR**: visualization of the output from GFERR. D.3, 135
**GFMODEL**: computes a ground fault current model from a measured transient. D.4, 137
**GETAMF**: retrieves file names of impulse response or ground fault data in a given directory. D.5, 138
**WRTESRC**: writes a given time function to a user defined (type-1) ATP source. D.6, 140
**MKCABLE**: computes impedance and admittance matrices for Π-section cable models for a set of different cable types and a range of frequencies. This function generates input and batch files for ATP which then computes the matrices. In addition, phase and modal parameters are computed and saved in LaTeX format. D.7, 141
**DISMUB**: the DISMO-toolbox for analysis and transient detection of power systems signals. D.8, 148
**EXTDATA**: extract down-sampled data from the large ground fault experiment data material. D.9, 158
**KYVDATA**: experiment data browser. D.10, 159

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

Line 256 to 266 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
256 if bSaveData != 1
257    eval(['save ' strGferrFile ' dErr strImpNode strGfNode cConnect'])
258 end
```

In line 324 ComputeAllError is defined as

```
gferr.m
263 function [Fftgfm,Fftcim] = ComputeAllError(strGfName,nGf,strGfFile,...
264    strImpFile,dImp,...
265    dImpFft,freq,findx,hpFir,NhpFir,Nfft,...
266    bPlotImp,nFigImp,bPlotFft,nFigFft,bPlotModel,nFigModel)
267    dErr (:,nGf) = [sum(abs(Fftgfm.Fftcim))/size(Fftgfm,1)]';
```

D.2 Ground Fault Localization Algorithm

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
```

Most of the parameters to this function are used to control the graphic 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.
Matlab Functions

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.

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.

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**.

**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 62 respectively.
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 itself 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 32-33 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.

```matlab
function dhp = gfmodel2(hpgf, hp)

GFMODEL Computes a ground fault model.

% dhp = gfmodel2(hpgf, hp)
% Kaare Jean Jensen, 1998-06-22

iDim = isvector(hpgf);
if iDim == 0, error('hpgf must be a vector'), end

NFilter = length(hp);
NHPDelay = fix(NFilter/2);
Nmdl = length(hpgf) + NFilter - 1;

% D.4 Ground Fault Model

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.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.
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.

```matlab
function writesrc(src, file)
% WRITESRC Writes user defined source to file
% writesrc(src, file)

% Kaare Jean Jensen, 1998-08-26
% file = fopen(file, 'wt');
% fprintf(fid,'/PLOT/n');
for n = 1:length(src)
    if src(n) > 0
        fprintf(fid, '%s %s', real2str(src(n), 7), ...
        real2str(-src(n), 8));
    else
        fprintf(fid, '%s %s', real2str(src(n), 8), ...
        real2str(-src(n), 7));
    end
end
fprintf(fid, '99999n');
fclose(fid);
else
    error(['file ' can not be opened'])
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.

```matlab
mkcable.m

iAtpFreq = [50, 100, 200, 500, 1000, 2000,...
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.

```matlab
mkcable.m

% Character number 1, 4, 5 and length(strCabName(n,:)) - 2 will
% % be used for the ATP file name so they must be non-blank.
% strCabName = str2mat(['Al 240 APB', 'Al 150 APB', 'Cu 150 APB',
% 'Al 95 APB', 'Cu 95 APB', 'Cu 50 APB',
% 'Al 240 PEX', 'Al 150 PEX', 'Al 95 PEX',
% '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

```matlab
mkcable.m

% Al APB geometrical dimensions
AlApbGeoDim = [200, 150, 95;...
5.2, 5.2, 5.2; % Core-core distance
4, 4, 4; % Core-pipe distance
2.0, 1.9, 1.7; % Pipe thickness
52.5, 45, 39; % Outer pipe diameter
0.1, 0.1, 0.1; % Pipe insulation thickness
2.0, 1.9, 1.7; % Cross section

% Cu APB geometrical dimensions
CuApbGeoDim = [50, 35, 25;...
5.2, 5.2, 5.2; % Core-core distance
4, 4, 4; % Core-pipe distance
1.9, 1.7, 1.6; % Pipe thickness
45, 39, 33; % Outer pipe diameter
0.1, 0.1, 0.1; % Pipe insulation thickness
5.2, 5.2, 5.2; % Cross section

% Al PEX geometrical dimensions
AlPexGeoDim = [200, 150, 95;...
5.2, 5.2, 5.2; % Core-core distance
4, 4, 4; % Core-pipe distance
2.0, 1.9, 1.7; % Pipe thickness
52.5, 45, 39; % Outer pipe diameter
0.1, 0.1, 0.1; % Pipe insulation thickness
2.0, 1.9, 1.7; % Cross section

% Cu PEX geometrical dimensions
CuPexGeoDim = [50, 35, 25;...
5.2, 5.2, 5.2; % Core-core distance
4, 4, 4; % Core-pipe distance
1.9, 1.7, 1.6; % Pipe thickness
45, 39, 33; % Outer pipe diameter
0.1, 0.1, 0.1; % Pipe insulation thickness
5.2, 5.2, 5.2; % Cross section

% Cu OHL (overhead line) geometrical dimensions
CuOHLGeoDim = [50, 35, 25;...
25, 25, 25; % Ground wire cross section
25, 25, 25; % Vertical distance to 1st phase
9.27, 9.27, 9.27; % Vertical distance to 2nd phase
9.27, 9.27, 9.27; % Vertical distance to 3rd phase
4.95, 4.95, 4.95; % Horizontal distance to 1st phase
0, 0, 0; % Horizontal distance to 2nd phase
-0.75, -0.75, -0.75; % Horizontal distance to 3rd phase
```
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

<table>
<thead>
<tr>
<th>% Material parameters for copper</th>
</tr>
</thead>
<tbody>
<tr>
<td>CuMatPar = [1.78E-8,... % core rho</td>
</tr>
<tr>
<td>0.999991]; % core my_r</td>
</tr>
<tr>
<td>% Material parameters for aluminum</td>
</tr>
<tr>
<td>AlMatPar = [2.38E-8,... % core rho</td>
</tr>
<tr>
<td>1.00002]; % core my_r</td>
</tr>
<tr>
<td>% Material parameters for iron (only used for overhead line ground wire)</td>
</tr>
<tr>
<td>FeMatPar = [9.78E-8,... % core rho</td>
</tr>
<tr>
<td>1.0]; % core my_r</td>
</tr>
<tr>
<td>% Material parameters for APB cables</td>
</tr>
<tr>
<td>ApbMatPar = [2.2E-7,... % pipe rho</td>
</tr>
<tr>
<td>0.999983,... % pipe my_r</td>
</tr>
<tr>
<td>3.8,... % pipe epsilon1 (insulator inside pipe)</td>
</tr>
<tr>
<td>4.0,... % pipe epsilon2 (insulator surrounding pipe)</td>
</tr>
<tr>
<td>1,... % core insulator my</td>
</tr>
<tr>
<td>3.5]; % core insulator epsilon</td>
</tr>
</tbody>
</table>

% Material parameters for PEX cables |
| PexMatPar = [CuMatPar(1),... % pipe rho (Cu)|
| CuMatPar(2), % pipe my_r (Cu) |
| 1,... % pipe epsilon1 (insulator inside pipe) |
| 2.3,... % pipe epsilon2 (insulator surrounding pipe) |
| 1,... % rho for semi conducting material |
| 1,... % my_r for semi conducting material |
| 1,... % core insulator 2 (PEX) my |
| 2.0]; % 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 50Hz, 2kHz, and 100kHz 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 Cu95APB will be called c95a.1is, an Al150PEX will be called a15p.1is, and a Cu35OHL will be called c35o.1is. Remember that ATP input are ATP files and output are LD files. The phase and modal parameters are written to two files called capar1.tex and capar2.tex and could look the examples in Tables D.1 and D.2. These parameters are computed at 1kHz.

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

- cdvar.tex: description of MKCABLE and ATP variable names.
- cdapbgeo.tex: geometrical dimensions for APB cables.
- cdapbatp.tex: ATP dimensions for APB cables.
- cpexgeo.tex: geometrical dimensions for PEX cables.
- cpexatp.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 these files through LATEX to get an overview of all parameters for all cables.
D.7 ATP Cable Model Generator

### 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.

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

As the last action, the sequence and modal parameters for all frequencies are saved to the file name in `strPosVar` and `texRun` is run on the TEX files to generate a document with all computed data.
### 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 intention that new features developed in the DISMO project should be implemented 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 cannot 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 (freq-), 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 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 on page 100.

If the Frequency estimation check box is checked the instantaneous frequency is estimated and saved to the file name given in the respective field at the bottom of the window. If the box is unchecked the power frequency is assumed to be 50Hz and no frequency estimation is saved.
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`, `-freq`, 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 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 frames 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
residual signal computed by the fundamental estimation algorithm (a -res file). If the fundamental component is not removed 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 Kovacevic, 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.

### Implementation

D.8 The DISMO-Toolbox

An outline of the graphical user interface implementation of the DISMO-toolbox is discussed in this section. The reader is 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 DISMTB 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 control 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.

<table>
<thead>
<tr>
<th>file name</th>
<th>menu item name</th>
</tr>
</thead>
<tbody>
<tr>
<td>mdeffile.m</td>
<td>Default File Names</td>
</tr>
<tr>
<td>msetdefs.m</td>
<td>Set Default Parameters</td>
</tr>
<tr>
<td>msavepar.m</td>
<td>Save All Parameters at Exit</td>
</tr>
<tr>
<td>mvalidtb.m</td>
<td>Validate Toolbox</td>
</tr>
<tr>
<td>mfundest.m</td>
<td>Fundamental Estimation</td>
</tr>
<tr>
<td>mtrandet.m</td>
<td>Transient Detection</td>
</tr>
<tr>
<td>mwavelet.m</td>
<td>Wavelet Analysis</td>
</tr>
<tr>
<td>mcombine.m</td>
<td>Combined analysis</td>
</tr>
</tbody>
</table>

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.

D.8 The DISMO-Toolbox

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:
   ```matlab
   elseif strcmp(alias, 'mymenu') label = 'My New Menu Item';
   ```
   It is a good idea to keep the alias seven characters long so that 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 mymymenu.m) that defines the controls 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:
   ```matlab
   uimenu(fadmenu,...
   'label', stralias('mymenu')....
   'callback',[hide, mymymenu('show')]);
   ```

4. Add a line to the initializing and exiting section of dismo.m. These two lines would look like:
   ```matlab
   mymymenu('init')
   ```
   and
   ```matlab
   mymymenu('exit')
   ```

5. If the menu item contains default parameters following lines must be added to the exit action of mymymenu.m:
if strcmp(msavepar('getstatus'),'on')
message('('mymenu') 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

The following lines must be added to savepars.m and the following lines must be added to movepars.m:
clear
message('moving mymenu.mat', 'debug')
eval(['load ' mdeffile('getoldtmppath') '/mymenu'])
eval(['delete ' mdeffile('getoldtmppath') '/mymenu.mat'])
eval(['save ' mdeffile('gettmppath') '/mymenu'])

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 the voltage probe. A full list of these names are given in Table F.2 on page 175.

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 Kyunday which has the acronym KY 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
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 Experiment Data Browser

#### 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 10kΩ 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, KYVATA 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 KYVATA with a high-pass FIR filter applied to the time domain signal. The filter cut-off frequency is 2kHz 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:

makenet: generates network models for ATP ............. E.1, page 16
atp2mat: converts ATP output to MAT files ............ E.2.1, page 16
dat2mat: converts DISMO-PC acquisition files .......... E.2.2, page 16
b2mat: converts BAKKER acquisition files ............ E.2.2, page 16
E.1 Network Model Generation

This program is called makenet and generates a full network model of H-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.

<table>
<thead>
<tr>
<th>Terminal output</th>
</tr>
</thead>
<tbody>
<tr>
<td>makenet version 3.5</td>
</tr>
<tr>
<td>usage: makenet [-dhks] NetDefFile</td>
</tr>
<tr>
<td>-d Print debug information</td>
</tr>
<tr>
<td>-h Display this help</td>
</tr>
<tr>
<td>-k List valid keywords</td>
</tr>
<tr>
<td>-s Run silent</td>
</tr>
</tbody>
</table>

The `-d` option makes makenet print messages of all actions which can be useful for trouble-shooting. Option `-s` makes 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.

<table>
<thead>
<tr>
<th>Terminal output</th>
</tr>
</thead>
<tbody>
<tr>
<td>makenet version 3.5</td>
</tr>
<tr>
<td>Following keywords are valid:</td>
</tr>
<tr>
<td>setgroundresistance [resistance in ohms/m] (default: 0.025)</td>
</tr>
<tr>
<td>setcablelength [length of Pi-equivalent in meters] (default: 10)</td>
</tr>
<tr>
<td>setcabledir [dir] (must precede definecable keyword)</td>
</tr>
<tr>
<td>definecable [file] [cable type] [node name, 4ch]</td>
</tr>
<tr>
<td>newcablesection [node name, 2ch] [length] [cable type]</td>
</tr>
<tr>
<td>addcable [node name, 2ch] [length] [cable type]</td>
</tr>
<tr>
<td>addsplit [node name, 2ch] [length] [cable type]</td>
</tr>
<tr>
<td>usesplit [node name, 2ch]</td>
</tr>
<tr>
<td>load [kW] [cos(φ)]</td>
</tr>
<tr>
<td>externmode [node name, 5ch]</td>
</tr>
<tr>
<td>Note: Ich following node names means that the mandatory length of the node name is I characters.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Terminal output</th>
</tr>
</thead>
<tbody>
<tr>
<td>atp2mat, version 1.1</td>
</tr>
<tr>
<td>usage: atp2mat [-dhks] atpout.lis</td>
</tr>
<tr>
<td>-d Print debug information</td>
</tr>
<tr>
<td>-h Display this help</td>
</tr>
<tr>
<td>-k List valid keywords</td>
</tr>
<tr>
<td>-s Run silent</td>
</tr>
</tbody>
</table>

The `-d` option will make atp2mat print messages for all actions. This is useful for trouble-shooting. The `-s` 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, das2mat, b2mat, and w2mat. These conversion programs have very similar behavior and accept 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.

```
<table>
<thead>
<tr>
<th>Terminal output</th>
</tr>
</thead>
<tbody>
<tr>
<td>dat2mat, version 1.4</td>
</tr>
<tr>
<td>Converts binary output from GlobalLab (DAT) to MAT files</td>
</tr>
<tr>
<td>Usage: dat2mat [-cdhmprs] [-o matdir] infile</td>
</tr>
<tr>
<td>Options:</td>
</tr>
<tr>
<td>-c do not write data, Check only</td>
</tr>
<tr>
<td>-d print Debug information</td>
</tr>
<tr>
<td>-h display this Help</td>
</tr>
<tr>
<td>-m write Output to directory matdir/infile</td>
</tr>
<tr>
<td>-o matdir write Output to directory matdir/infile</td>
</tr>
<tr>
<td>-p Print dat header</td>
</tr>
<tr>
<td>-r Recurse through sub directories</td>
</tr>
<tr>
<td>-s run Silent</td>
</tr>
<tr>
<td>Description:</td>
</tr>
<tr>
<td>- If infile is a directory and option r is on, it will be searched recursively</td>
</tr>
<tr>
<td>- If a TXT file is found with the same name as the DAT file, it will be searched for a time and date string. This will be used to compose a directory name. The format for this file is:</td>
</tr>
<tr>
<td>TIME=16:47</td>
</tr>
<tr>
<td>DATE=981130</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>
```

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 filenames 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 of 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, `fr_z`, 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 `0000m001, 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 `fr_z`. 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 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 \( \text{dec} \).

Four options can be passed to `downsmpl` on the command line. The \(-h\) option makes `downsmpl` print following help to the terminal.

\[
\begin{array}{|c|}
\hline
\text{downsmpl version 1.2} \\
\text{Equivalent to Matlab's \texttt{FILTER}} \\
\text{usage: downsmpl [-hs] [-o Outfile] Infile.mat Filter.mat} \\
\text{-h } \text{Display this help} \\
\text{-o File Write output to file Outfile} \\
\text{-n No decimation} \\
\text{-s Silent} \\
\hline
\end{array}
\]

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@1999.5.01`.

If `timer` is executed with the option \(-h\) following output is printed to the terminal:

\[
\begin{array}{|c|}
\hline
\text{usage: timer [-options] [commandfile]} \\
\text{options: -b: Make a beep at execution} \\
\text{-h: Display this help} \\
\text{-l: Write a log file} \\
\hline
\end{array}
\]

As described by the above output, other options can be given to `timer`. The \(-1\) 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 \(-s\) 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 were discussed. For the sake of completeness, this chapter also presents transient spectra 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 information.
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 wWK520 acquired a three phase voltage and current signal using probes and Rogowski coils at a sampling frequency of 1 MHz. The wWK524 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 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 Data Storage

<table>
<thead>
<tr>
<th>Signal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ur, us, ut</td>
<td>voltage on phase R, S, and T</td>
</tr>
<tr>
<td>ir, is, it</td>
<td>current on phase R, S, and T</td>
</tr>
<tr>
<td>u0p</td>
<td>zero system voltage from probe</td>
</tr>
<tr>
<td>u0t</td>
<td>zero system voltage from measurement transformer</td>
</tr>
<tr>
<td>i0r</td>
<td>zero system current from Rogowski coil</td>
</tr>
<tr>
<td>i0t</td>
<td>zero system current from measurement transformer</td>
</tr>
</tbody>
</table>

### Table F.2: Description of signal names.

<table>
<thead>
<tr>
<th>Channel</th>
<th>Signal</th>
<th>Channel</th>
<th>Signal</th>
<th>Channel</th>
<th>Signal</th>
<th>Channel</th>
<th>Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ur</td>
<td>1</td>
<td>us</td>
<td>1</td>
<td>ur</td>
<td>1</td>
<td>ur</td>
</tr>
<tr>
<td>2</td>
<td>ir</td>
<td>2</td>
<td>is</td>
<td>2</td>
<td>i0p</td>
<td>2</td>
<td>us</td>
</tr>
<tr>
<td>3</td>
<td>us</td>
<td>3</td>
<td>ut</td>
<td>3</td>
<td>igf</td>
<td>3</td>
<td>ut</td>
</tr>
<tr>
<td>4</td>
<td>is</td>
<td>4</td>
<td>it</td>
<td>4</td>
<td>i0r</td>
<td>4</td>
<td>ir</td>
</tr>
<tr>
<td>5</td>
<td>ut</td>
<td>5</td>
<td>ur</td>
<td>5</td>
<td>i0t</td>
<td>5</td>
<td>is</td>
</tr>
<tr>
<td>6</td>
<td>it</td>
<td>6</td>
<td>ir</td>
<td>6</td>
<td>u0t</td>
<td>6</td>
<td>it</td>
</tr>
</tbody>
</table>

### Table F.3: Acquisition system channel identification.

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
Ground Fault Experiment Data

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 m2mat 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 wukec520 system and from Table F.3 we find that the phase R current at wukec520 is channel 4. The root directory is therefore wukec520 and the file name is 0004001.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: /wukec520/nesa02.800/077.inx/0004001.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 file containing all channels. To retrieve the channel number we find the signal name from Table F.2 as 10t and from Table F.3 we find that 10t 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 name 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 in the form lxry_mn.dat where x is the location number, y represents the resistance, and mn 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 is the first example is /dismo/conf2/1l1r1_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, even though 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/nesa0.303 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 technician from the producer of the transient recorder came up from Switzerland to trouble shoot. 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**: 2kΩ resistance.
- **Location 3**: all resistances.
- **Location 6**: 0.5kΩ and 1kΩ 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, were build into the main circuit 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 BARKER) 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 KYVDA described in Section D.10 on page 159.

The terminal output from EXTDATA describes 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.
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 period 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 crucial 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 77 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 A 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 A 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 V/\(\sqrt{3}\) 0.11 kV and with an additional factor 3.50 applied.

CT, PC current transformer mounted in the Petersen coil with a sensitivity of 2V/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 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 elliptical 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 elliptical 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.

As seen from Figure F.1 the filter magnitude is very close to zero dB (unity) for frequencies below 1 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

---

**F.2 Conversion Factors**

![Figure F.1: Magnitude of anti-aliasing filter response.](image1)

![Figure F.2: Argument of anti-aliasing filter response.](image2)

![Figure F.3: Magnitude of amplifier response.](image3)

![Figure F.4: Argument of amplifier response.](image4)
Table F.5: Amplifier gain for the DISMO-PC.

<table>
<thead>
<tr>
<th>phase</th>
<th>voltage</th>
<th>current (a)</th>
<th>current (b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>5.552</td>
<td>9.972</td>
<td>5.154</td>
</tr>
<tr>
<td>S</td>
<td>5.562</td>
<td>10.19</td>
<td>5.127</td>
</tr>
<tr>
<td>T</td>
<td>5.548</td>
<td>9.998</td>
<td>5.149</td>
</tr>
</tbody>
</table>

Table F.6: Calibration factors for combi-sensors.

<table>
<thead>
<tr>
<th>phase</th>
<th>calibration factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>1.0090</td>
</tr>
<tr>
<td>S</td>
<td>1.0072</td>
</tr>
<tr>
<td>T</td>
<td>1.0092</td>
</tr>
</tbody>
</table>

Table F.7: Conversion factors for DISMO-PC.

<table>
<thead>
<tr>
<th>phase</th>
<th>voltage</th>
<th>current (a)</th>
<th>current (b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>0.5497</td>
<td>0.04940</td>
<td>0.03186</td>
</tr>
<tr>
<td>S</td>
<td>0.5487</td>
<td>0.04824</td>
<td>0.03197</td>
</tr>
<tr>
<td>T</td>
<td>0.5501</td>
<td>0.04929</td>
<td>0.03190</td>
</tr>
</tbody>
</table>

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 spectra 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 8 for more details.
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 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.
Ground Fault Experiment Data

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.

Figure F.15: Current, location 1, configuration 1.

Figure F.16: Current, location 5, configuration 1.

Figure F.17: Current, location 3, configuration 1.

Figure F.18: Current, location 5, configuration 1.

F.4 Simulation of the Experiment Data
Ground Fault Experiment Data

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.

F.4 Simulation of the Experiment Data
Ground Fault Experiment Data

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.

F.4 Simulation of the Experiment Data
Bibliography


## Index

A/D converter, 181
A12, 7, 50
ABB, x, 7, 69, 74
aliasing, 44
anti-aliasing filter, 176, 180, 181
APB, 25, 143
apparent power, 32
arc suppression coil, 35
ARMA, 22, 23
ATP, 21, 30, 32, 43, 94
ATP, 163, 185
ATV, vii, 22

**BCTRAN**, 31, 32
busbar, 3, 50, 71

**C++ programs**
- atp2mat, 123, 163, 165
- atpinput, 120, 126, 127
- b2mat, 158, 163, 165, 167, 174
- dat2mat, 158, 163, 165–167, 174
- downsamp, 126, 158, 163, 164, 167, 168
- makenet, 37, 116, 119–121, 123, 126, 127, 141, 163, 164
- timer, 9, 10, 163, 164, 166, 168, 169
- w2mat, 158, 163–165, 167, 174

**CABLE CONSTANTS**, 24, 30
**CABLE PARAMETERS**, 25
centralized monitoring, 4
cepstrum, 94
characteristic impedance, 16
closing angle, 72, 81, 174–176
combi-sensor, 8, 77, 162, 173, 182

**DATA BASE MODULE**, 115
DEFU, vii, 69
delta, 32
deregulation, 1
DISMO, vii, 148
DISMO-PC, 9, 77, 81, 162, 163
distribution network, 1
DTU, vii, 22
Dy, 32
ELTEK, x
Index

EMTP, 21
fault resistance, 72, 78, 81
FFT, 84
FIR, 162

Glentegården, 7
ground fault, 4, 69

Hannings window, 84
high voltage, 2
HV, 2
HZA, 74

IAU, vii
IIR, 9, 23
IMM, vii
integrator, 8, 100
inverse sequence, 15

JMARTI SETUP, 23
Kyndby, 159

Linux, 21, 163
low voltage, 2
LV, 2, 35, 70, 71, 74

Matlab, 129
Matlab functions
MKCABLE, 143
DISMOTB, 130, 155
EXTDATA, 130, 158, 159, 176, 177
FFT, 134
FILTER, 167
FIR1, 162
GETAMP, 130, 132, 138
GETCTRLH, 155
GFERR, 130, 131, 135, 137
GPMODEL2, 130, 134, 137
HIDECTRL, 155
KYVDATE, 130, 159-162, 174, 177
KYVFACT, 160, 161, 178
KYVNAME, 160, 161, 174, 175
MENUS, 155
MESSAGE, 158
MKCABLE, 130, 141, 142, 144, 145, 147
MTRANDET, 156
REAL2STR, 141
SETCTRLH, 155
SHOWERR, 130, 131, 135, 136
STRALIAS, 155
UICONTROL, 155
WHAT, 139
WRITESRC, 115, 130, 140

medium voltage, 2
modal, 18, 23, 130, 145, 147
MV, 2, 4, 35, 37, 69-72, 74, 77, 80

NESA, 22
NODA SETUP, 23
Nyquist, 44
observation point, 4, 50, 55, 60, 61, 63, 115, 118

Outage, 4

Petersen coil, 3, 4, 35, 36, 71, 91, 116, 159, 161, 172, 180
PEX, 25, 143
positive sequence, 15, 24, 31
power base, 32
primary substation, 2, 35, 47, 50, 91, 92, 115, 118
proximity effect, 24
Rulebook, 21, 32, 113
secondary substation, 3
SEMLYEN SETUP, 23
skin effect, 24
Star, 32
TERRA, 25, 28, 39, 115
Theorybook, 21, 23
transmission network, 1
winding, 31
zero sequence, 15, 31
Zy, 35, 71, 116