

DIPLOMARBEIT

Modellierung von Kommunikationsabläufen im intelligenten Niederspannungsnetz

ausgeführt zur Erlangung des akademischen Grades
eines Diplom-Ingenieurs unter der Leitung von

O.Univ.Prof. Dipl.-Ing. Dr.techn. DIETRICH Dietmar
Projektass. Dipl.-Ing. FASCHANG Mario
Projektass. Dipl.-Ing. DIMITRIOU Pavlos

am

Institut für Computertechnik (E384)
der Technischen Universität Wien

durch

BRÄUER Rainer, BSc
Matr.Nr. 0626369

Wien, am 22. September 2014

Kurzfassung

Bei der Forschung und Entwicklung im Bereich der intelligenten Niederspannungsnetze, oft Smart Grids genannt, ist unter anderem die Analyse der Kommunikationsabläufe zwischen den kommunizierenden Geräten von großer Bedeutung. Eine der Herausforderungen dabei ist, dass die zur Informationsübertragung genutzten Energieübertragungsnetze naturgemäß nicht für Informationsübertragung optimiert sind. Es treten daher z.B. Reflexionen an nicht abgeschlossenen Leitungsenden, Interferenzen durch angeschlossene Geräte sowie inhomogenen Leitungstypen auf. Daraus ergeben sich ein geringeres Signal-to-Noise-Ratio, relativ geringe Bandbreiten und hohe Verlusten. Ein Power Line Communication Protokoll, welches mit diesen Eigenschaften umgehen kann, ist das Automated Metering and Information System (AMIS)-Powerline Protokoll. Zur Analyse dieses Protokolls wird im Rahmen eines Forschungsprojektes vom Institut für Computertechnik der Technischen Universität Wien ein Kommunikationssimulator entwickelt. Dieser wird durch Messungen von realen Testnetzen an diese Testnetze angepasst. Das Ziel dieser Diplomarbeit war es, diesen Kommunikationssimulator so zu erweitern, dass eine Simulation von nicht gemessenen Niederspannungsnetzen anhand allgemeiner Daten über die Netzstruktur, wie z.B. die Leitungslänge, Anzahl der Abzweigungen, Anzahl der Sammelschienen, durchgeführt werden kann. Dazu wurde ein MATLAB Programm implementiert, mit welchem zuerst Daten aus mehreren verschiedenen Datenquellen zu einer Datenbasis zusammengeführt werden. Anschließend können mit der erstellten Datenbasis verschiedene Analysen durchgeführt werden. Trotz eingehender Analyse der erstellten Datenbasis, zuerst mit einem grafischen, später mit einem heuristischen Ansatz, konnte mit den vorliegenden AMIS-Messdaten und den Daten über die Netzstruktur kein Kommunikationsmodell entwickelt werden, welches die Kommunikationsabläufe zufriedenstellend beschreibt. Deshalb wurde das Ziel abgeändert, es sollte nun eine Verbesserung der Anpassung des Kommunikationssimulators an gemessene Netze erfolgen. Dies geschah durch Erweiterung des bereits für die Analyse implementierten MATLAB Programms derart, dass damit eine detailliertere statistische Auswertung der vorhandenen AMIS-Messdaten möglich gemacht wurde. Im Vergleich zur bisher vom Kommunikationssimulator verwendeten, über die gesamten AMIS-Messdaten eines Testnetzes gemittelte, Auswertung ist nun eine Berechnung der statistischen Kommunikationsparameter für jedes im Testnetz vorhandene AMIS-Gerät einzeln möglich. Damit wird eine wesentliche Verbesserung des Kommunikationssimulators erreicht. Das dazu im Zuge dieser Diplomarbeit implementierte MATLAB Programm kann weiters als Basis für weitere, neue oder noch detailliertere Analysen von AMIS-Testnetzen verwendet werden.

Abstract

At science and development in the area of intelligent low power grid networks, which are often called Smart Grids, there is the need to analyse the communication behaviour between the communicating devices. One challenge is that the power grid networks, that are used for information transmission, are not designed and optimized for information transmission. So there arise effects like reflections on not properly closed wire endpoints, interferences from connected devices, and inhomogenous wire types. These result in a low Signal-to-Noise-Ratio, low bandwidth (some kb/s) and high loss rates. One of the protocols that can deal with this properties is the Automated Metering and Information System (AMIS)-Powerline Protokoll. To be able to analyse this protocol a communication simulator is developed at the Institute of Computer Technology (ICT) of the Vienna University of Technology (TU-Wien). This communication simulator is fitted to real power grid networks by measurements of AMIS parameters done in this networks. The aim of this master thesis was to enhance this communication simulator in a way that a simulation of power grids, from which no measurements have been taken, is possible. This should be possible by having only information about the power grid itself, like distance of the wires, number of connections and number of bus bars. There was a MATLAB program implemented which first merges the data from several different datasources to a single datastructure, and then analyses this datastructure. Despite exhaustive analysis of the datastructure, beginning with a graphical analysis and after that with a heuristic approach, the development of a communication model that provided satisfactory results, could not be achieved. Because of that the aim was redefined, now the fitting to measured power grid networks should be enhanced. That was done by enhancing the MATLAB program, which was already implemented for the analysis, so that the calculation of the statistical communication parameters of the AMIS measurements could be done more accurate. Compared to the older statistical analysis of the AMIS measurements, which resulted in averaged values of the communication parameters for all AMIS measurements of a complete low voltage power grid network, it is now possible to calculate the statistical parameters for each AMIS device separately. This enhancement is a considerably improvement of the communication simulator. The MATLAB program, which was implemented for this master thesis, can also be used as a basis for new analysis or further enhancements and more detailed analysis of the AMIS measurements.

Abkürzungsverzeichnis und Glossar

AG	Aktiengesellschaft
AMI	Advanced Metering Infrastructure
AMIS	Automated Metering and Information System
AMIS-CX1	Automated Metering and Information System Compatibly/Consistently Extendable Transport Profile V.1
BEWAG	Burgenländischen Elektrizitätswirtschaft Aktiengesellschaft
CO2	Kohlenstoffdioxid
csv	Comma Separated Values
CSG	Compact Switch Gear, Kompaktschaltanlage
CX1	Compatibly/Consistently Extendable Transport Profile V.1
DA	Diplomarbeit
DC	Data Concentrator
DG	Distributed Generation
dgs	DIgSILENT-GIS-SCADA
EGDA	Express Grid Data Access
GA	Genetic Algorithm
GA-Solver	Genetic Algorithm Solver
GIS	Geografische Informations Systeme
GmbH	Gesellschaft mit beschränkter Haftung
GPRS	General Packet Radio Service
ICT	Institut für Computertechnik
ID	Identifier
IT	Information Technology
JAVA	Objektorientierte Programmiersprache
kb/s	Kilobit pro Sekunde
LV	Low Voltage, Niederspannung
MS	Microsoft
Niederspannung	Netzspannung zwischen 230V/400V (einphasig/dreiphasig)
OLTC	On Load Tap Changer
PLC	Power Line Communication
SCADA	Supervisory Control and Data Acquisition
SD-Karte	Secure Digital Memory Card
SGTS	Smart Grids Topologie Simulator
sic	sic erat scriptum → „wirklich so“
TU	Technische Universität

UML	Unified Modeling Language
UMTS	Universal Mobile Telecommunications System
USA	United States of America
xlsx	Excel Datei ohne Makros
xml	Extensible Markup Language

Inhaltsverzeichnis

1	Einleitung	1
1.1	Smart Grid - Das intelligente Stromnetz	1
1.2	Aufgabenstellung	3
1.3	Methodik und Aufbau der Arbeit	4
2	Umfeldanalyse	5
2.1	State of the Art	5
2.1.1	Digitale Repräsentation von Niederspannungsnetzen	5
2.1.2	Das Automated Metering and Information System Protokoll	7
2.2	Verwandte Arbeiten	9
2.2.1	Kommunikationssimulator „DG DemoNet Smart LV Grid“	10
2.2.2	Bisherige Analyse des Automated Metering and Information System Protokolls am Institut für Computertechnik	12
2.2.3	Bachelor Arbeit „Smart Grids Topologie Simulator“	15
2.2.4	Paper „Study on Signal Attenuation Characteristics in Power Line Communications“	16
3	Datenaufbereitung	18
3.1	Gemessene Daten	19
3.1.1	Testnetze	19
3.1.2	Datensatz „AMIS-Messdaten“	19
3.2	Digitale Beschreibung der gemessenen Niederspannungsnetze	21
3.2.1	Datensatz „Netzstruktur“ (Variante 1)	22
3.2.2	Datensatz „Netzstruktur“ (Variante 2)	23
3.2.3	Datensatz „Netzstruktur“ (Variante 3)	25
3.2.4	Datensatz „Smart Grids Topologie Simulator“	26
3.2.5	Datensatz „Abzweigungen/Sammelschienen“	27
3.3	Zusammenführen der Datensätze	29
3.3.1	Zusammenführen mit Datensatz „Netzstruktur“ (Variante 1)	30
3.3.2	Zusammenführen mit Datensatz „Netzstruktur“ (Variante 2)	32
3.3.3	Zusammenführen mit Datensatz „Netzstruktur“ (Variante 3)	34
3.4	Implementierung des MATLAB Programms (Datenaufbereitungsteil)	34
3.4.1	Verwendete Datenstrukturen	36
3.4.2	Erstellung der Datenbasis für die Analyse	40
3.4.3	Struktur des MATLAB Programms	42
4	Analyse der erstellten Datenbasis	50
4.1	Grafische Analyse der Datenbasis	50
4.1.1	Verwendete Diagramme	50
4.1.2	Implementierung des MATLAB-Programms (Analyseteil)	56
4.1.3	Ergebnis der grafischen Analyse	62
4.2	Heuristische Analyse der Datenbasis	67

4.2.1	Allgemeine Beschreibung der Heuristik	67
4.2.2	Implementierung in MATLAB	71
4.2.3	Ergebnis der heuristischen Analyse	73
4.3	Verbesserung des Kommunikationssimulators	73
4.3.1	Aufbau der Input Datei	73
4.3.2	Erstellung der Input Datei	74
4.4	Ergebnis der Verbesserung des Kommunikationssimulators	75
5	Diskussion	76
5.1	Diskussion der Datenaufbereitung	76
5.2	Diskussion der Analyse	77
5.3	Diskussion der Verbesserung des Kommunikationssimulators	78
6	Zusammenfassung und Ausblick	79

1 Einleitung

Spätestens seit Mitte der Neunziger Jahre bildet sich weltweit bei viele Menschen, Organisationen und Staaten das Bewusstsein, dass die Energieerzeugungs- und Energieverteilungssysteme effizienter und nachhaltiger werden müssen, um die von der Menschheit verursachte Beeinflussung auf die Erde, z.B. auf das Klima, zu reduzieren. Derzeit erfolgt ein großer Teil der Energieerzeugung, z.B. im Bereich der Wärmeerzeugung, bei Verbrennungskraftmaschinen und speziell im Bereich der elektrischen Energieerzeugung, durch fossile Energieträger, wie z.B. Braun- und Steinkohle, Erdgas und Erdöl. All diesen fossilen Energieträgern ist gemein, dass die Umwandlung der in den Brennstoffen enthaltenen Energie in die neue, gewünschte Form mit der Erzeugung von CO_2 einhergeht. Aufgrund der Wirkung von CO_2 als Treibhausgas wird eine Verringerung des CO_2 Ausstoßes mittlerweile von vielen Wissenschaftlern und Politikern als wünschenswert bzw. notwendig erachtet.

Ein wichtiger Meilenstein dazu ist das Kyoto-Protokoll, welches im Wesentlichen eine Vereinbarung zur Verringerung des CO_2 Ausstoßes in Industrieländern zu völkerrechtlich verbindlichen Zielwerten ist. Auch wenn dieses Übereinkommen nicht durchgehend den gewünschten Erfolg brachte, weil z.B. die USA als weltweit führende Nation des CO_2 Ausstoßes dem Kyoto-Protokoll nie beigetreten sind und einige Länder (darunter auch Österreich) die vereinbarten Ziele bei Weitem nicht erreicht haben, wird die Setzung von Zielen von verschiedensten Organisationen als sinnvoll erachtet. Deshalb hat auch die Europäische Union das Kyoto-Protokoll ratifiziert. Das Kyoto-Protokoll ist im Jahre 2012 ausgelaufen, die Europäische Union hat sich mittlerweile mit dem Projekt 20-20-20 auf ein Richtlinien- und Zielpaket geeinigt, welches im Wesentlichen Folgendes beinhaltet [EuZ]:

Es gelten bis zum Jahr 2020 die folgenden europaweiten Vorgaben:

- * 20 % weniger Treibhausgasemissionen als 2005*
- * 20 % Anteil an erneuerbaren Energien*
- * 20 % mehr Energieeffizienz*

Wie man an dieser Aufzählung erkennen kann ist die Optimierung der Erzeugung, Übertragung und Verteilung von elektrischer Energie ein Ziel bzw. eine Methode, welche sehr hohe Priorität auf der Agenda der Europäischen Union hat. Um die vorgenommenen Ziele zu erreichen wird es notwendig sein, die Energieerzeugungssysteme und Energieübertragungsnetze an die neuen Gegebenheiten, wie z.B. eine stärker schwankende Energieerzeugung durch einen steigenden Anteil von Wind- und Photovoltaikanlagen, anzupassen.

1.1 Smart Grid - Das intelligente Stromnetz

Die notwendigen Anpassungen, welche das derzeitige elektrische Versorgungssystem fit für die zukünftigen Herausforderungen machen sollen, beinhalten, dass unsere elektri-

schen Energieerzeugungssysteme dezentraler und unsere Energieübertragungsnetze intelligenter werden. Der für diese intelligenten Übertragungsnetze meistens gebrauchte Begriff lautet Smart Grid. Wie so oft gibt es auch bei diesem Begriff keine einheitliche Definition. Eine oft verwendete, sehr allgemein gehaltene Definition ist jene der europäischen Technologieplattform „Smart Grids“ [Smab]:

The concept of SmartGrids ... concerns an electricity network that can intelligently integrate the actions of all users connected to it - generators, consumers and those that do both - in order to efficiently deliver sustainable, economic and secure electricity supplies. A smart grid employs innovative products and services together with intelligent monitoring, control, communication, and self-healing technologies in order to:

- * *Better facilitate the connection and operation of generators of all sizes and technologies*
- * *Allow consumers to play a part in optimising the operation of the system*
- * *Provide consumers with more information and better options choosing their energy supplier*
- * *Significantly reduce the environmental impact of the whole electricity supply system*
- * *Maintain and improve the existing high levels of system reliability, quality, and security of supply*
- * *Maintain and improve the existing services efficiently*
- * *Foster the development of an integrated European market.*

Eine etwas andere Definition kommt von der nationalen österreichischen Technologieplattform Smart Grids [Smaa]:

Smart Grids sind Stromnetze, welche durch ein abgestimmtes Management mittels zeitnaher und bidirektionaler Kommunikation zwischen Netzkomponenten, Erzeugern, Speichern und Verbrauchern einen energie- und kosteneffizienten Systembetrieb für zukünftige Anforderungen unterstützen.

Der für diese Diplomarbeit wichtigste Teil dieser Definition ist *ein abgestimmtes Management mittels zeitnaher und bidirektionaler Kommunikation*. Um ein Smart Grid zu realisieren ist es also unbedingt notwendig, dass die Netzteilnehmer (oft auch Knoten genannt) bidirektional miteinander kommunizieren können. Dies ist ein fundamentaler Unterschied zu klassischen elektrischen Energieverteilungsnetzen. Bei diesen erfolgt eine Regelung der Energieeinspeisung durch eine implizite Regelung über die Netzfrequenz. Eine Kommunikation mit Netzteilnehmern zu Überwachungs- und Steuerungszwecken erfolgt bis jetzt schwerpunktmäßig auf der Hochspannungsebene, wohingegen laut [Smaa] in den niedrigeren Spannungsebenen derzeit überwiegend ohne Systemmanagement und direkte Anbindung an zentrale Steuerungs- und Überwachungseinrichtungen

gearbeitet wird.

Um eine bidirektionale Kommunikation zwischen den Netzteilnehmern zu ermöglichen stehen technisch unterschiedlichste Lösungen zur Verfügung:

- * Modemverbindung
- * Breitbandverbindung (Kupfer- oder Glasfaser)
- * Funkverbindung (WiMax, WiFi, ZigBee, proprietäre Systeme, ...)
- * Mobilfunkverbindung (GPRS, UMTS, ...)
- * Power Line Communication (PLC)

Da jedes dieser Kommunikationssysteme Vor- und Nachteile hat, können in einem Smart Grid mehrerer dieser Kommunikationssysteme vorhanden sein.

1.2 Aufgabenstellung

Am Institut für Computertechnik (ICT) (Technische Universität (TU) Wien) wird von der Energy&IT Group bereits seit einiger Zeit im Zuge von verschiedenen Projekten, auch in Zusammenarbeit mit anderen etablierten Partnern aus Forschung und Industrie, im Themenbereich Smart Grid geforscht. Eines dieser Projekte beschäftigt sich mit der Analyse und Vorhersage von Kommunikationsparametern bei PLC-Kommunikation im Niederspannungsnetz. Dabei soll ein Simulator entwickelt werden, welcher verschiedene Kommunikationsparameter bei Verwendung eines bestimmten PLC-Kommunikationsprotokolls, berechnen kann. Das verwendete Protokoll ist das von SIEMENS entwickelte AMIS Powerline Protokoll. Mit Smart Metern, welche das AMIS Powerline Protokoll verwenden, werden verschiedenste Netz- und AMIS-Kommunikationsparameter in realen Niederspannungsnetzen zu Testzwecken gemessen. Eine erste, statistische Analyse des Protokolls wurde von der Energy&IT Group am ICT der TU Wien bereits durchgeführt. Dabei wurden die in den realen Testnetzen gemessenen AMIS Kommunikationsparameter analysiert, um damit eine statistische Verteilung zu gewinnen und in weiterer Folge eine Vorhersage der Kommunikation mit dem zu entwickelnden Simulator durchführen zu können. Mit der bestehenden Analyse ist jedoch nur eine statistische Vorhersage der AMIS-Kommunikationsparameter derart möglich, dass über alle im Testnetz gemessenen AMIS-Kommunikationsparameter gemittelt wird, wobei die Netzstruktur des Niederspannungsnetzes unberücksichtigt bleibt. Diese Vereinfachung ist deshalb notwendig, da in den vorhandenen AMIS-Messdaten keine direkte Information über die physikalische oder logische Netzstruktur vorhanden ist, und es auch keine Möglichkeit der Zuordnung von AMIS-Messdaten zu den Daten über die Netzstruktur gibt.

Das Ziel dieser Diplomarbeit ist es deshalb, ein Modell für die Vorhersage von AMIS-Kommunikationsparametern in Niederspannungsnetzen zu entwickeln, welches vom am ICT der TU-Wien entwickelten Kommunikationssimulator verwendet werden kann. Dieses Modell soll anhand von Daten über die Netzstruktur (Leitungslänge, Anzahl der

Abzweigungen, Anzahl der Sammelschienen) den auftretenden *hopCount* vorhersagen können, ohne dass für diese Testnetze AMIS-Messdaten vorliegen müssen.

1.3 Methodik und Aufbau der Arbeit

Um das eben erläuterte Ziel zu erreichen ist es unter anderem notwendig, über die Funktionsweise von Smart Grids ein gewisses Grundwissen und spezifischere Informationen über das in den Testnetzen verwendete AMIS-PLC-Protokoll zu haben. Die benötigten Informationen dazu werden in Kapitel 2 bereitgestellt. Weiters erfolgt in dem eben genannten Kapitel eine genauere Beschreibung der bisher am ICT durchgeführten Analysen des AMIS-Protokolles und eine übersichtsmäßige Beschreibung, wie die Netzstruktur der Niederspannungsnetze digital gespeichert und dargestellt werden kann. Da auch die für diese Diplomarbeit zur Verfügung stehende digitale Beschreibung der Netzstruktur nicht einheitlich ist, wird in Kapitel 3 der genaue Aufbau der unterschiedlichen Datensätze beschrieben. Anschließend erfolgt eine Erläuterung, wie das Zusammenführen der verschiedenen Datensätze zu einem aggregierten Datensatz erfolgt. Wie die Analyse dieses aggregierten Datensatzes erfolgt und die dabei gewonnenen Erkenntnisse werden in Kapitel 4 beschrieben. Das eben genannte Kapitel enthält auch die Beschreibung, wie der am ICT entwickelte Kommunikationssimulator durch die in dieser Diplomarbeit durchgeführte Analyse des AMIS-Protokolls verbessert wurde. Im Kapitel 5 erfolgt eine eingehende Diskussion über die Ergebnisse dieser Diplomarbeit. Im abschließenden Kapitel 6 erfolgt eine Zusammenfassung dieser Diplomarbeit und ein Ausblick, wie die Entwicklung in diesem zukunftssträchtigen Forschungsgebiet, speziell mit Bezug auf die in dieser Diplomarbeit eingehend behandelten Aspekte, verlaufen könnte.

2 Umfeldanalyse

Um den Lösungsansatz leichter zu verstehen wird zuerst im Kapitel 2.1 der derzeit existierende Stand der Technik von technischen Systemen, welche in dieser Diplomarbeit verwendet werden bzw. damit in engem Zusammenhang stehen, beschrieben. Danach werden im Kapitel 2.2 Forschungsarbeiten, welche ebenfalls mit dieser Diplomarbeit in, teilweise sehr engem, Zusammenhang stehen, beschrieben.

2.1 State of the Art

Es folgt nun eine Beschreibung des von Forschungseinrichtungen und Unternehmen im Energiesektor verwendete Standes der Technik. Dabei erfolgt eine Einschränkung auf jene Bereiche, welche für diese Diplomarbeit relevant sind.

2.1.1 Digitale Repräsentation von Niederspannungsnetzen

Energieübertragungsunternehmen verwenden auf der Niederspannungsebene verschiedenste digitale Beschreibungen für die Abbildung der Netzstruktur. Auch heute nicht wegzudenken sind grafische Ortspläne nach dem Muster von Abbildung 1. Diese stellen eine gute Übersicht über die vorhandene Netzstruktur dar und enthalten relevante Detailinformationen, wie z.B. diverse Nummern von Zählern und Netzanschlüssen oder die Kabeltypen der verwendeten Kabel.

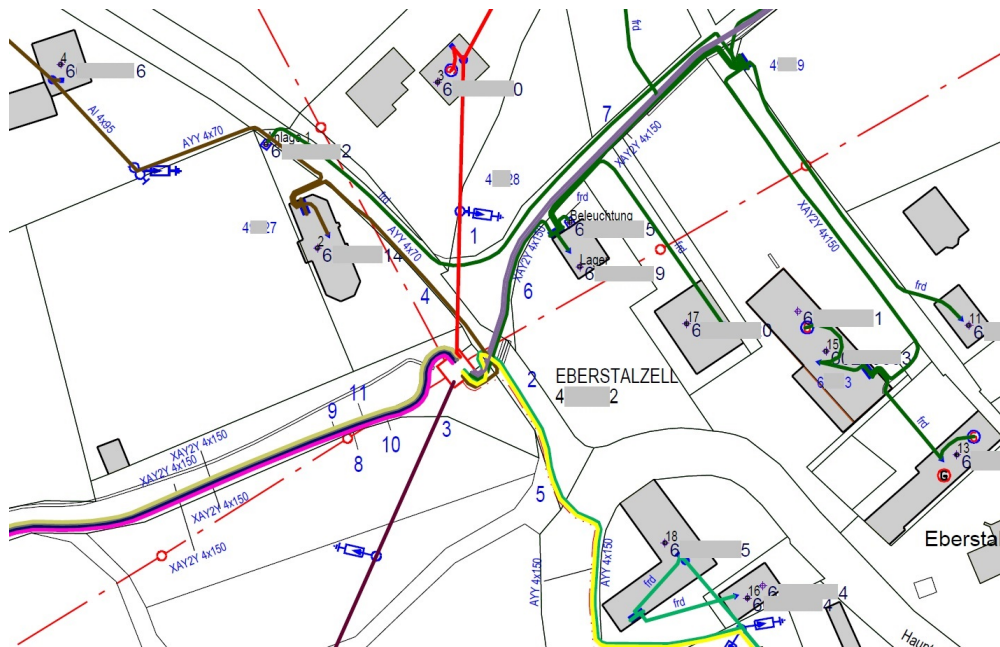


Abbildung 1: Ausschnitt eines Ortsnetzplans; Quelle: Netz Oberösterreich GmbH

Diese und andere Daten, welche die Netzstruktur beschreiben, liegen heutzutage teilweise auch für Niederspannungsnetze in digitaler Form vor. Eine mögliche digitale Datenrepräsentation sind klassische Comma Separated Values (csv) Dateien, wie z.B. die in Abbildung 2 dargestellte Beispieldatei (geöffnet mit Microsoft Excel wegen besserer Übersichtlichkeit). Diese, für manche vielleicht betagt wirkende Speicherart von Daten, hat nach wie vor ihre Berechtigung, da sie von sehr vielen Programmen unterstützt wird (weil sie quasi den kleinsten Nenner für einen Datenaustausch darstellt) und die Daten auch relativ effizient gespeichert werden. Aber auch Extensible Markup Language (xml) Dateien wie in Abbildung 3 werden mittlerweile verwendet, vorwiegend natürlich zum einfachen Datenaustausch zwischen verschiedenen Programmen. Natürlich gibt es auch noch die verschiedensten proprietären Datenformate, z.B. zur Speicherung von Netzplänen. Am ICT verwendet wird z.B. das DIgSILENT-GIS-SCADA (dgs) Format des Programms PowerFactory vom Hersteller DIgSILENT, welches laut [dig] zur Modellierung, Analyse und Simulation von Energiesystemen, verwendet werden kann.

	A	B	C	D	E	F	G
1	ON 01245 ORTSNAME						
2	Stationsnr neu	Stationsname	Strangnr alt	Stangnr neu	Anschlussobjektnr	Funktionsklasse	Serialnr Hersteller
3	45678	ORTSNAME	123456789	45678-01	10123456	AMIS	G01-123-C.01/987-01234
4						AMIS	G01-123-C.01/987-01235
5						AMIS	G01-123-C.01/987-01236
6					10123457	AMIS	G01-123-C.01/987-01237
7					10123458	AMIS	G01-123-C.01/987-01238
8					10123459	AMIS	G01-123-C.01/987-01239
9					10123460	AMIS	G01-123-C.01/987-01240
10			987654321	45678-02	10123461	AMIS	G01-123-C.01/987-01241
11					10123462	AMIS	G01-123-C.01/987-01242
12						AMIS	G01-123-C.01/987-01243
13						AMIS	G01-123-C.01/987-01244

Abbildung 2: Zuordnung zu Ortsnetzplan - Beispieldatei (Inhalt fiktiv)

```
<?xml version="1.0" encoding="UTF-8"?>
<DGS xsi:noNamespaceSchemaLocation="ON2-01471_EBERSTALZELL_Reduced.xsd" xmlns:xsi="http://www.
format="DIgSILENT (R) DGS Export V1.2.1" date="2012-07-17T10:39:53" application="DIgSILENT (C) Powe
- <ElmCoup>
  <ID>2</ID>
  <loc_name>LS/TR Schalter1_0147110/0147111</loc_name>
  <fold_id>471</fold_id>
  <typ_id/>
  <on_off>0</on_off>
  <aUsage>sdc</aUsage>
  <chr_name/>
</ElmCoup>
- <ElmCoup>
  <ID>3</ID>
  <loc_name>LS/TR Schalter2_0147110/0147111</loc_name>
  <fold_id>471</fold_id>
  <typ_id/>
  <on_off>0</on_off>
  <aUsage>sdc</aUsage>
  <chr_name/>
</ElmCoup>
```

Abbildung 3: Zuordnung zu Ortsnetzplan mit XML; Quelle: Netz Oberösterreich GmbH

Durch die voranschreitende Digitalisierung der Netzpläne, auch auf der Niederspannungsebene, ist es heute einfacher möglich Analysen bzw. Simulationen der Niederspannungsverteilstnetze durchzuführen, als dies in Zeiten ohne vorhandene digitale Daten über die Energieverteilnetze der Fall war.

2.1.2 Das Automated Metering and Information System Protokoll

Wie bereits in der Einleitung erwähnt, wird in dieser Diplomarbeit das Automated Metering and Information System (AMIS) Powerline Protokoll genauer analysiert. Das AMIS Powerline Protokoll wurde von Siemens speziell für Smart Grid PLC Anwendungen entwickelt. Es ist nicht als eine alleinstehende Entwicklung zu sehen, vielmehr ist es ein Teil des AMIS Konzeptes von Siemens, welches wiederum eine konkrete Implementierung des Advanced Metering Infrastructure (AMI) Ansatzes ist. Abbildung 4 zeigt ein Schichten-

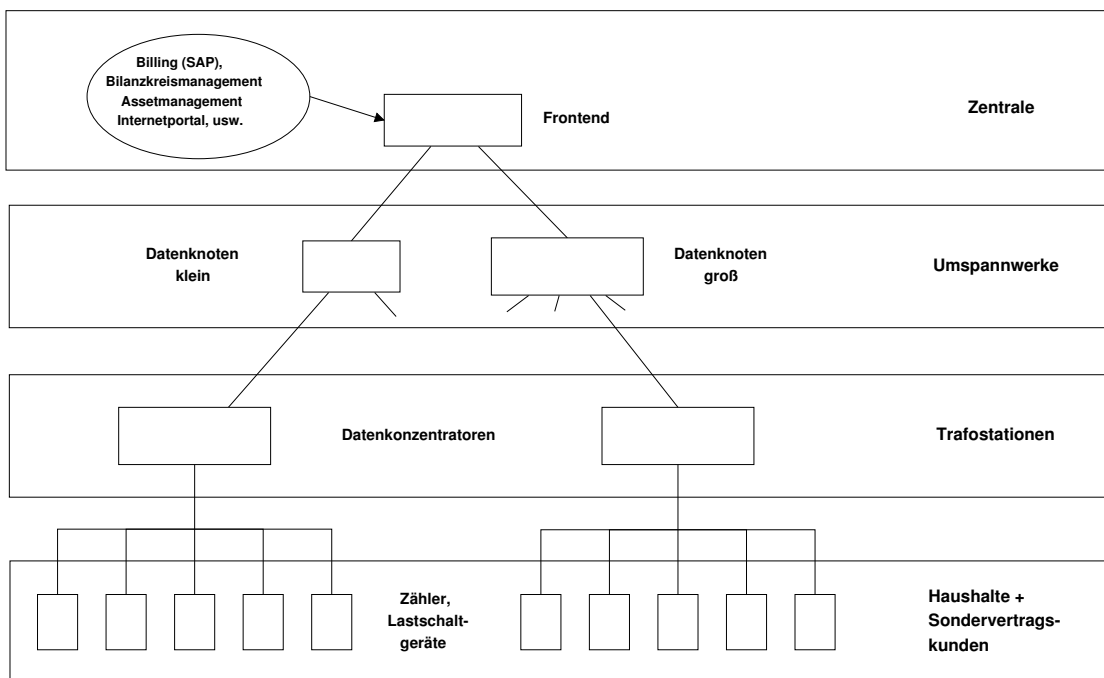


Abbildung 4: AMIS Systemüberblick (vereinfacht); Quelle: [AMI06]

modell des AMIS Konzeptes. Für diese Diplomarbeit wichtig sind die zwei untersten Schichten, da dort zwischen den Knoten mittels PLC kommuniziert wird.

Bevor die für diese Diplomarbeit wichtigsten Punkte zusammengefasst werden, erfolgt ein Auszug aus der Spezifikation des Automated Metering and Information System Compatibly/Consistently Extendable Transport Profile V.1 (AMIS-CX1) Powerline Protokolls, da der gewählte Abschnitt das Protokoll für Personen, welche mit der Materie vertraut sind, relativ schnell und einfach beschreibt [AMI11, S. 22-24]:

Überblick

„Das Kommunikationsprotokoll gemäß AMIS CX1-Profil (Master/Slave) wird im System AMIS zur Ankopplung von AMIS Endgeräten (Zähler, Lastschaltgerät, Fremdgeräte Gateway) an den Datenkonzentrator AMIS DC eingesetzt. Die Kommunikation erfolgt über öffentliche Niederspannungsnetze. Anwendungsgebiete sind die Verbrauchsdatenerfassung und das Management von Verteilnetzen als Komplettlösung AMIS.

Als Gemeinschaftsverkehr wird ein serielles Kommunikationsprotokoll bezeichnet bei dem eine Zentralstation mit einer oder mehreren Unterstationen über eine Kommunikationsverbindung in einer Linien- oder Sternkonfiguration verbunden sind. Der Datenverkehr wird durch die Zentralstation gesteuert. Bei Gemeinschaftsverkehr wird eine Datenübertragung immer nur von der Zentralstation (MASTER) eingeleitet.

Im Gemeinschaftsverkehr wird eine unsymmetrische Übertragungsprozedur verwendet. Das bedeutet das die Zentralstation als Primärstation alle Nachrichtenübertragungen veranlasst, während die Unterstationen die Sekundärstationen sind nur übertragen dürfen, wenn sie aufgerufen sind.

Funktionsübersicht

Kommunikation einer Zentrale (AMIS DC) mit einem oder mehreren Endgeräten (AMIS).

- Unbalanced Multi-Point (Gemeinschaftsverkehr) gemäß AMIS Protokollfestlegungen für Datenübertragung auf der Niederspannungsleitung gemäß AMIS CX1-Profil.

- * Unterstützte Funktionalität entsprechend AMIS Protokollfestlegungen
- * Datenerfassung durch Abfragen (Stationsabfrage)
- * Automatische Suche von neuen Endgeräten
- * Erfassen von Ereignissen (übertragen sendebereiter Daten)
- * Uhrzeitsynchronisation
- * Befehlsübertragung
- * AMIS-Datenübertragung
 - „Time Credit“ Verfahren (Bandbreitenbegrenztes Übertragungsverfahren)
 - Optimierte BROADCAST Behandlung mit Quittungsverfahren
 - Überwachungsfunktion für Transaktionsserver
 - Master-Wechsel durch das Endgerät
 - Datenübertragung durch „Simultanes Forwarding“

Technische Daten

Verkehrsabwicklung

- * *Gemeinschaftsverkehr: Master - Slave (Halbduplex)*
- * *Reichweitenerhöhung durch „Simultanes Forwarding“*
- * *8 Hop-Ebenen*

Physikalische Schnittstelle

- * *240V Netzanschluss*

Anzahl der Stationen

- * *Gemeinschaftsverkehr: 2000 AMIS Endgeräte*
- * *Reichweitenerhöhung durch „Simultanes Forwarding“*
- * *8 Hop-Ebenen*

Modulation (CX1-Profil Layer 1)

- * *DBPSK (Differential Binary Phase Shift Keying)*
- * *DQPSK (Differential Quadrature Phase Shift Keying)*

unterstützte Baudraten

- * *600, 900, 1200, 1500, 1800, 2400, 3000*

Anhand dieses Auszugs aus der Spezifikation lässt sich erkennen, dass das AMIS Powerline Protokoll für eine sehr einfache und robuste (im Sinne von „widerstandsfähig gegenüber Ausfällen und äußeren Einflüssen, welche die Übertragung verschlechtern“) Kommunikation ausgelegt wurde. Weiters wurde Wert darauf gelegt, dass sich das System nach Änderungen, z.B. dem Hinzufügen bzw. Entfernen von Knoten oder dem Umschalten bzw. Ausfall von Leitungen, automatisch an die neuen Gegebenheiten anpasst und möglichst wenig durch manuelle (menschliche) Eingriffe rekonfiguriert werden muss.

2.2 Verwandte Arbeiten

In diesem Kapitel erfolgt eine Beschreibung von Forschungsarbeiten, welche mit dieser Diplomarbeit in, teilweise sehr engem, Zusammenhang stehen. Der Zusammenhang lässt sich leicht damit erklären, da einige der genannten Arbeiten ebenso wie diese Diplomarbeit am ICT der TU Wien verfasst wurden und den gleichen Themenbereich behandeln. Teilweise baut diese Diplomarbeit auch direkt auf den beschriebenen Arbeiten auf bzw. erweitert die bestehenden Arbeiten um neue Funktionen oder Aspekte.

2.2.1 Kommunikationssimulator „DG DemoNet Smart LV Grid“

Wie in [DGDa] ersichtlich ist, wird im Projekt „DG DemoNet Smart LV Grid“ von den beteiligten Projektpartner an Lösungen für die zukünftigen Herausforderungen, welche auf das Niederspannungsnetz zukommen, gearbeitet. Ein Ziel dabei ist laut [DGDb] die Maximierung erneuerbarer Energien. Dabei soll die bereits vorhandene Infrastruktur weitestgehend genutzt werden, um damit eine kommunikationsbasierende, automatische Regelung von Niederspannungsnetzen zu entwickeln und bei Praxistest deren Tauglichkeit zu evaluieren. Um diese umfangreichen Aufgaben durchführen zu können sind eine Reihe von namhaften Partnern aus Industrie und Forschung (z.B. Siemens AG Österreich, Netz Oberösterreich GmbH, TU Wien – Institut für Elektrische Anlagen und Energiewirtschaft, TU Wien – Institut für Computertechnik) an dem Projekt beteiligt.

Laut [FDLK12, S. 5] ist der Projektpartner am ICT die Energy & IT Group, welche eine Analyse der bidirektionalen Einflüsse zwischen dem elektrischen Stromversorgungsnetz auf der einen Seite und dem Kommunikationssystem auf der anderen Seite durchführt. Zu diesem Zweck wurde eine Simulationsumgebung entwickelt, deren Architektur in Abbildung 5 abgebildet ist. Dieser Architektur liegt die Hypothese zu Grunde, dass es einfacher ist, für jede Domäne (elektrisches Stromversorgungssystem, Kommunikationssystem) einzelne, spezialisierte Simulatoren zu verwenden und diese zu koppeln („Coupled Simulation“) und nicht einen einzelnen Simulator zu verwenden, welcher die verschiedenen Modelle zur Berechnung der verschiedenen Domänen beinhaltet.

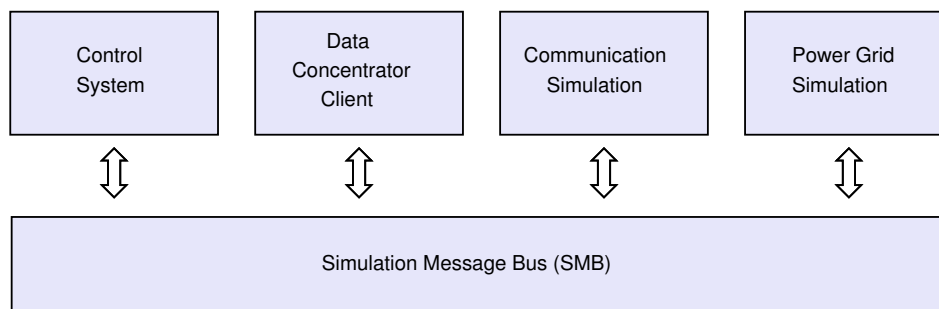


Abbildung 5: Architektur der „Coupled Simulation“; Quelle: [FDLK12]

Aus der eben beschriebenen Hypothese ergibt sich zwangsläufig ein modularer Aufbau der Simulationsumgebung. Es erfolgt nun eine kurze Beschreibung der einzelnen Komponenten um die anschließende, für diese Diplomarbeit wichtige und deshalb detailliertere, Beschreibung der „Communication Simulation“ besser verstehen zu können.

Simulation Message Bus

Der Simulation Message Bus verbindet alle anderen Komponenten miteinander. Laut [FDLK12, S. 5] kann er als die Middleware zwischen den einzelnen Modulen gesehen

werden. Um eine Hardware In The Loop Emulation sowie einen späteren Feldeinsatz zu erleichtern ist es wichtig, dass das Verhalten der Kommunikation zur Datenverteilung möglichst ähnlich zu dem von Hardware Komponenten in realen Einsatzszenarien ist. Laut [FDLK12, S. 7] braucht der Simulation Message Bus jedoch kein Wissen über die zu transportierenden Daten, sondern führt lediglich Routing Operationen zwischen den Komponenten aus.

Power Grid Simulation

Wie in [FDLK12, S. 5ff] beschrieben ist, wird in der Power Grid Simulation das Verteilnetz für die elektrische Energie simuliert. Dazu wird die Software DigSILENT PowerFactory benutzt, welche stationäre und transiente Analysen durchführen kann. Es wurde auch eine Erweiterung der in der Software vorhandenen Standardnetzwerkkomponenten durchgeführt (z.B. Erweiterung des Types Niederspannungstransformators mit einem On Load Tap Changer (OLTC)).

Data Concentrator Client

Laut [FDLK12, S. 18] wird im Data Concentrator Client das Verhalten eines AMIS Data Concentrator (DC), welcher im AMIS Powerline Protokoll eine zentrale Rolle spielt, simuliert. Der DC kommuniziert mit den Smart Metern via AMIS Powerline Protokoll, mit anderen (simulierten) Hardware Komponenten (z.B. OLTC) jedoch via ModBus. Um eine realistische Repräsentation des Verhaltens aller (simulierten) Hardware Komponenten zu erhalten wurden deshalb zwei Clients implementiert: ein Express Grid Data Access (EGDA) Client für Smart Meter und ein ModBus Client für z.B. OLTC.

Control System

Im Control System werden die Messdaten des Verteilnetzes, welche über den DC verfügbar sind, analysiert. In [FDLK12, S. 6] wird beschrieben, dass auf Basis dieser Analyse etwaige System-Aktuatoren (z.B. OLTC) angesteuert werden.

Communication Simulation

Laut [FDLK12, S. 6] werden in der Communication Simulation die Kommunikationsabläufe im Verteilnetz simuliert. Das verwendete Kommunikationsprotokoll ist das AMIS Powerline Protokoll.

In [FDLK12, S. 12] wird beschrieben, dass es aufgrund der Charakteristik des AMIS Powerline Protokolls nicht möglich war, die Kommunikationsabläufe in bereits existierenden Kommunikationssimulatoren für Netzwerk abzubilden. Deshalb wurde eine eigenes Communication Simulation (Kommunikationssimulation) Modul entwickelt. In diesem Modul wird der Kommunikationskanal auf Basis von gemessenen, statistisch ausgewerteten AMIS Packet log Daten, modelliert.

Wie in [FDLK12, S. 13] beschrieben wird, kann es wegen der erheblichen Einflüsse des Verteilnetzes und der Umwelt auf die PLC-Kommunikation passieren, dass Pakete während der Übertragung verloren gehen bzw. verzerrt werden. Um dies zu beschreiben werden zwei statistische Parameter verwendet: die Verlustwahrscheinlichkeit (P_{loss}) und die (approximierte und gauß-verteilte) Verzögerungszeit (T_{delay}). Diese Parameter werden berücksichtigt indem die Parameterwerte (P_{loss} und T_{delay}) in der Co-Simulation ("Coupled Simulation") durch das Communication Simulation Modul entsprechend den Kanalparameterwerten (Anzahl der retransmissions/hops) gesetzt werden. Um den Kommunikationskanal zu simulieren wurden echte AMIS Packet log-Daten von den zugehörigen Verteilnetzen verwendet.

2.2.2 Bisherige Analyse des Automated Metering and Information System Protokolls am Institut für Computertechnik

Laut [FD11, S. 3] soll das im Projekt Smart LV-Grid vorgesehene Regelungssystem über das AMIS-System mit Sensoren und Aktuatoren direkt oder indirekt kommunizieren. Dazu ist es notwendig, die Funktionsweise des AMIS-Powerline-Kommunikationssystems zu modellieren. Das entworfene Modell wird anschließend auch in einen Simulator integriert.

Wie in [FD11, S. 4] beschrieben erfolgt die Kommunikation nach dem AMIS-Protokoll auf einem gemeinsam genutzten Kanal nach dem Master/Slave-Prinzip. Der DC fungiert dabei als Master und die einzelnen Stationen (z.B. Zähler, Lastschaltgeräten) als Slaves. In Abbildung 6 ist die Kommunikation zwischen einem DC und einem Slave schematisch dargestellt. Die Zahlen bezeichnen die IDs der Zwischenstationen zwischen dem DC und dem adressierten Slave. Die zu sendende Nachricht wird vom DC an alle verbundenen Stationen mit einem Hinweis auf die aktuelle und die maximale Wiederholungsanzahl (max. Hops) versandt. Die Nachricht wird von allen Empfängern simultan so oft wiederholt (also an alle mit dem Empfänger verbundenen Stationen in der gleichen Weise weitergeleitet), bis die maximale Wiederholungsanzahl erreicht wurde („Simultanes Forwarding“). Hierbei stellt jeder Slave einen Repeater für dahinter liegende Slaves dar. Nachdem der Ziel-Slave die Nachricht erhalten hat muss dieser in einer vorgegebene maximale Zeit (in der Regel 40 ms) die empfangenen Daten verarbeiten und daraus eine Antwort generieren. Die Antwort wird vom Ziel-Slave immer, also auch wenn sich der Slave in einer niedrigeren als der maximalen Hop-Ebene befindet, erst dann zurückgesendet, wenn die vorgegebene Hop-Anzahl (=Wiederholungen) erreicht wurde.

Die Übertragung der Antwort zum DC erfolgt laut [FD11, S. 4] nach demselben simultanen Übertragungsprinzip und mit derselben Anzahl an Wiederholungen wie die ursprüngliche Anfrage. Bei dieser Kommunikation wird eine unsymmetrische Übertragungsprozedur verwendet. Das bedeutet, dass jede Datenübertragung vom DC angeregt wird und die Slave-Stationen nur dann senden, wenn sie vom DC dazu aufgefordert werden. Durch diese Vorgehensweise kann eine effiziente Kommunikation zwischen dem Master und den Slaves sichergestellt werden. Die Wiederholungsanzahl wird durch den Master

festgelegt und kann von diesem auch verändert werden. Die maximale Anzahl der Wiederholungen einer Nachricht (also der Hops) beträgt acht.

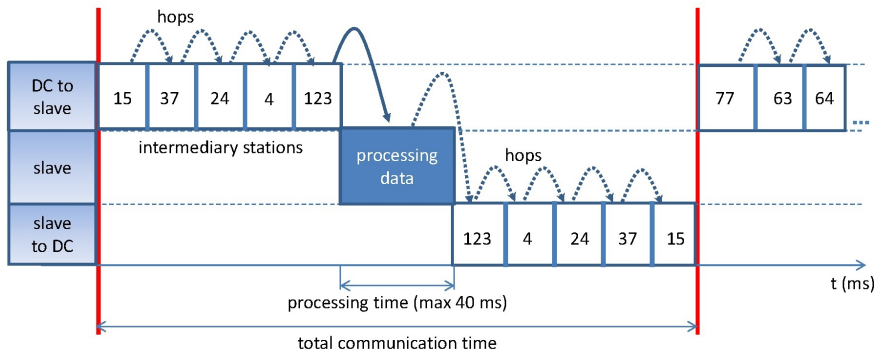


Abbildung 6: Schema der AMIS Kommunikation; Quelle: [FD11]

Zur Erhebung der Übertragungskennwerte der AMIS-Kommunikation wurde eine Analyse eines bestehenden AMIS-Kommunikationssystems anhand eines Datenbestandes von zwei Millionen Datensätzen durchgeführt. Ein solcher Datensatz enthält unter anderem folgende Informationen:

- * Zeitstempel
- * Laufzeit
- * ID der adressierten Station
- * Quittung
- * Prioritätsnummer
- * Phase
- * Modus
- * Signal-to-Noise-Ratio
- * Hop-Anzahl
- * Bit-Error-Rate

Laut [FD11, S. 6] wurden diese Daten in fünf unabhängigen Netzabschnitten gesammelt. Analysiert wurden die Parameter Paketlaufzeit und Paketverlust in Abhängigkeit von der Wiederholungsanzahl (Hops) mittels MATLAB. Attribute wie Tageszeit, Wochentag, Prozesstype, Übertragungsmodus und Phase wurden bei der Analyse nicht berücksichtigt. Die Ergebnisse der Analyse sind in Abbildung 7 und Abbildung 8 dargestellt. Aus Abbildung 7 ist eindeutig die Korrelation zwischen der Verzögerungszeit (Paketlaufzeit) und der Hop-Anzahl ersichtlich. Die am besten annäherbare, lineare Steigungsgleichung lautet (Best-fit Matlab):

$$f(x) = p1 * x + p2 \text{ wobei } p1 = 30.08, p2 = 23.12$$

In Abbildung 7 ebenfalls ersichtlich ist eine Zunahme der maximalen Streubreite der Verzögerungszeit um deren Mittelwert mit steigender Hop-Anzahl. Laut [FD11, S. 7] ergibt die statistische Analyse der Paketdaten hinsichtlich der Paketverluste die in Abbil-

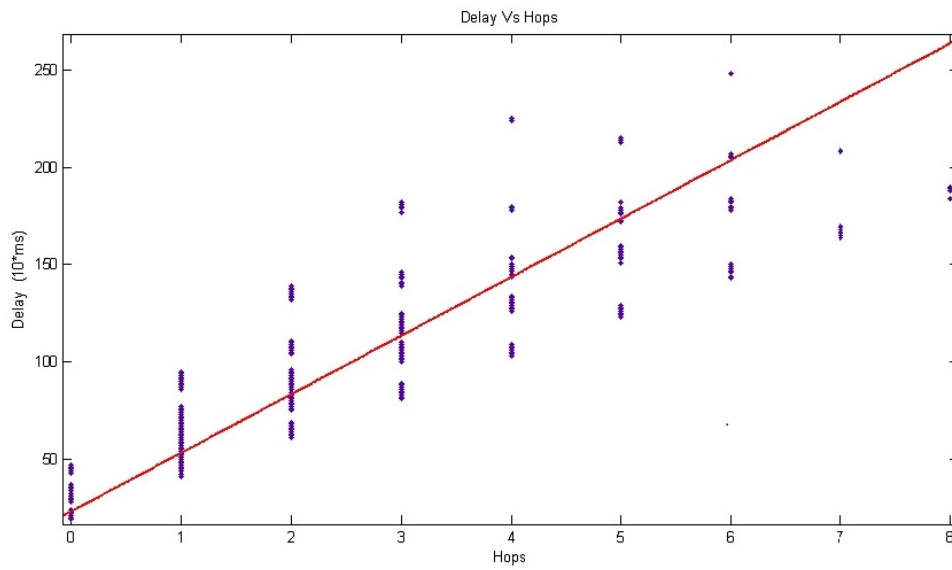


Abbildung 7: Statistische Auswertung der Paketlaufzeit; Quelle: [FD11]

Abbildung 8 dargestellten Daten. Diese repräsentieren die Paketverluste für die zwei-Wegen-Kommunikation (DC-Slave-DC) in Abhängigkeit von der Hop-Anzahl. Für die Hop-Anzahl 6,7 und 8 befinden sich keine Messwerte in den Daten.

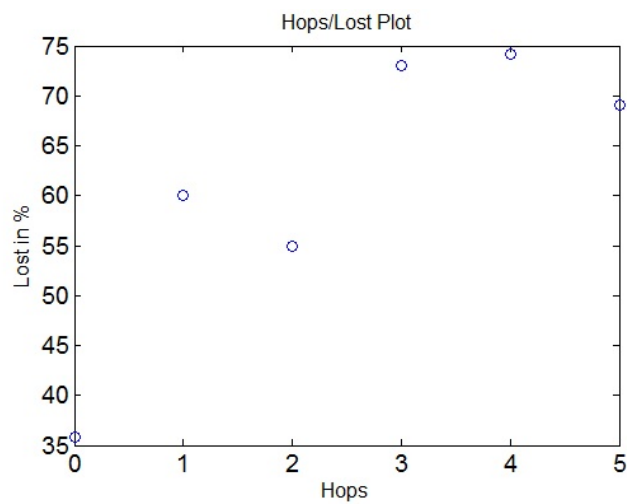


Abbildung 8: Statistische Auswertung der Paketverluste; Quelle: [FD11]

2.2.3 Bachelor Arbeit „Smart Grids Topologie Simulator“

Die Bachelor Arbeit „Smart Grids Topologie Simulator“ wurde am ICT der TU-Wien durchgeführt. Es folgt nun eine Beschreibung jener Teile, welche für diese Diplomarbeit relevant sind.

Aufgabenstellung

Laut [Zil, S. 11] soll mit dieser Bachelorarbeit das Verhalten des AMIS-Protokolls beleuchtet werden. In einem ersten Schritt werden reale oder fiktive Pläne von Niederspannungsnetzen in das zu erstellende Computerprogramm eingelesen. Die Komponenten im Netzplan werden dabei so abstrahiert, dass Knoten und Leitungen im Programm gespeichert werden. Der dahinterliegende, topologische Aufbau wird durch eine symmetrische Matrix nach dem Schema von Abbildung 9 dargestellt.

	DC	Knoten 2	Knoten 3	Knoten 4
DC	0	Leitung 1	0	0
Knoten 2	Leitung 1	0	Leitung 2	Leitung Z
Knoten 3	0	Leitung 2	0	Leitung 3
Knoten 4	0	Leitung Z	Leitung 3	0

Abbildung 9: (Ergebnis)Matrix eines Niederspannungsnetzes; Quelle: [Zil]

Einlesen des Netzplans und Erstellen der Matrix

Laut [Zil, S. 22] ist der erste Schritt das Einlesen eines Netzplans im NEPLAN-Format. Danach wird die topologische Information der eingelesenen Daten in eine Matrix transformiert. Diese Darstellung der Topologie dient anschließend, unter Berücksichtigung des Verhaltens des AMIS-CX1 Profils, zur Berechnung der Hop-Ebenen.

Wie in [Zil, S. 22] beschrieben konzentriert sich diese Bachelorarbeit beim Einlesen von Netzplänen auf das Programm NEPLAN. Für den Smart Grids Topologie Simulator (SGTS) und somit für den Algorithmus, welcher das AMIS-CX1 Profil abbildet, sind nur die Knoten, Leitungen und die Verbindung der Knoten miteinander wichtig.

Aus den eingelesenen Informationen der NEPLAN-Exportdateien wird laut [Zil, S. 25ff] eine Matrix erstellt, welche die Topologie des Netzplans enthält. Die Elemente der Matrix sind die Leitungen und jede Zeile bzw. Spalte repräsentiert einen Knoten, siehe Abbildung 9. Diese Matrix ist im Programm mit den jeweiligen Knoten und Leitungsobjekten gefüllt. Das bedeutet, dass mit der Matrix alleine, die gesamte benötigte Information zur Verfügung steht.

Berechnung der Hop-Ebenen

Das erstellte Programm SGTS berechnet mittels eines einfachen, rekursiven Algorithmus die Anzahl der Hop-Ebenen zwischen einem Sender und den Empfängern (allen anderen Netzteilnehmern).

Laut [Zil, S. 27] wird dazu die maximale Anzahl der zu bestimmenden Hop-Ebenen, die maximale Metrik und von welchem Knoten aus begonnen werden soll, benötigt. Der Algorithmus startet in der Zeile des Startknotens. Jedes Element ungleich Null in der Matrix stellt eine Verbindungsleitung zu einem folgenden Knoten dar. Das Programm folgt diesen Verbindungen und berechnet nach jeder Leitung die Metrik. Sobald der maximale Wert der Metrik überschritten wird, ist der Knoten nicht mehr erreichbar.

Resümee

Wie in [Zil, S. 32] beschrieben hat sich im Zuge dieser Bachelorarbeit durch Studium der einschlägigen Literatur unter anderem herausgestellt, dass die Berechnung der Dämpfung pro Längeneinheit zu kurz greift. Es ist zwar korrekt, dass durch die elektromagnetische Abstrahlung die Energie des Signals reduziert wird, dieser Effekt macht aber im Vergleich zu den auftretenden Reflexionen an allen Arten von Verbindungsstellen und Anschlüssen nur einen geringen Anteil aus. Um das Programm in einem ersten Schritt zu verbessern ist somit bei jedem Knoten eine zusätzliche Dämpfung anzunehmen.

Die aktuelle Wahl der Vorgehensweise entspricht laut [Zil, S. 32] bestenfalls einer Heuristik und keineswegs einem adäquaten nachrichtentechnischen Modell. Dieser Ansatz (Wahl der dämpfenden Elemente und der Wirkung) kann nun durch Versuche in realen Netzen verbessert werden. Durch Messung der auftretenden Signaldämpfung über verschiedene Leitungstypen, wie z.B. Erdkabel oder Freileitungen, können gute Werte ermittelt und ins Programm übernommen werden.

2.2.4 Paper „Study on Signal Attenuation Characteristics in Power Line Communications“

Laut [MK06, S. 2] ist es in PLC-Systemen oft ein Problem, dass die Möglichkeiten einer praktischen Anwendung nicht im Vorhinein evaluiert werden können. Deshalb wurde in dem genannten Paper versucht ein mathematisches Modell zu entwickeln, mit welchem sich die Dämpfungscharakteristik auf PLC-Signale schätzen lässt. Das mathematische Modell sollte dabei auf Basis von realen Messdaten aufgestellt werden. Stromleitungen sind darauf ausgelegt elektrische Energie zu übertragen, üblicherweise mit einer Frequenz von 50 Hz bzw. 60 Hz. Deshalb ist eine Übertragung von Kommunikationssignalen oft mit Problemen bzw. Herausforderungen verbunden. So werden z.B. die Dämpfungseigenschaften von Stromleitungen stärker durch Kabeltypen, Verkabelungsmethoden und Belastungen beeinflusst wie jene von z.B. Koaxialkabeln oder verdrehten Leitungen. Reflexionen am Ende von Nebenleitungen von Stromleitungen sind einer der

Hauptfaktoren für eine steigende Dämpfung der Kommunikationssignale. Da in Stromleitungen meist unterschiedliche Kabeltypen bei Haupt- und Nebenleitungen verwendet werden haben diese sehr oft unterschiedliche Impedanzen und es liegt eine Fehlanpassung vor. Auch die Lasten im Stromnetz haben unterschiedliche, dynamische Impedanzen und werden sehr oft auch offen gelassen, also im Leerlauf betrieben. Daher kommt es zu Reflexionen an den Verbindungspunkten zu Geräten oder Verbindungspunkten zwischen verschiedenen Kabeltypen.

Eine Erkenntnis des genannten Papers ist laut [MK06, S. 6], dass herausgefunden wurde, dass die Signaldämpfung an Bedeutung gewinnt wenn die Stromleitungen länger werden und/oder mehr Abzweigungen enthalten. Außerdem wurde erkannt, dass die Anzahl der Abzweigungen einen größeren Einfluss auf die Signaldämpfung hat als die Leitungslänge. Obwohl die Tendenz der Signaldämpfung ähnlich war bei gleichen Bedingungen (Leitungslänge und Anzahl der Abzweigungen) des Stromnetzes wurden jedoch auch Differenzen der Signaldämpfung festgestellt. Um das Modell zu verbessern wurde angedacht, weitere Umweltbedingungen (Zeit, elektrische Lasten, ...) in die Berechnung einzubeziehen.

3 Datenaufbereitung

Um ein Modell zur Vorhersage der in realen Netzen auftretenden Kommunikationsparameter (z.B. Hop Anzahl, Delay, ...) für Automated Metering and Information System (AMIS)-basierte Kommunikation in Niederspannungsnetzen erstellen zu können ist es erforderlich, eine Analyse gemessener AMIS Kommunikationsdaten bekannter Niederspannungsnetze durchzuführen. Da die AMIS Kommunikationsdaten nur Metadaten über die AMIS Kommunikation selbst (z.B. Hop Anzahl und Delay zwischen Slave Station und DC), jedoch keine Information über den (physikalischen) Aufbau des Niederspannungsnetzes enthalten (z.B. Leitungslänge zwischen Data Concentrator (DC) und AMIS-Endgeräten), ist es für eine Analyse erforderlich zuerst eine erweiterte bzw. aggregierte Datenbasis zu erstellen. Da die Datensätze über den Aufbau der Testnetze in verschiedenen Dateien und Datenformaten (z.B. als csv, Excel Datei ohne Makros (xlsx) oder xml-Datei) vorliegen ist es zuerst erforderlich all diese Datensätze zu einem aggregierten Datensatz (→ Datenbasis für die Analyse) zusammenzuführen. Der schematische Vorgang dazu ist in Abbildung 10 abgebildet. In den folgenden einzelnen Unterkapiteln erfolgt eine detailliertere Beschreibung der eben genannten Abbildung und des Zusammenführens. Dabei wird auch einige Male auf diese schematische Abbildung zurückgegriffen.

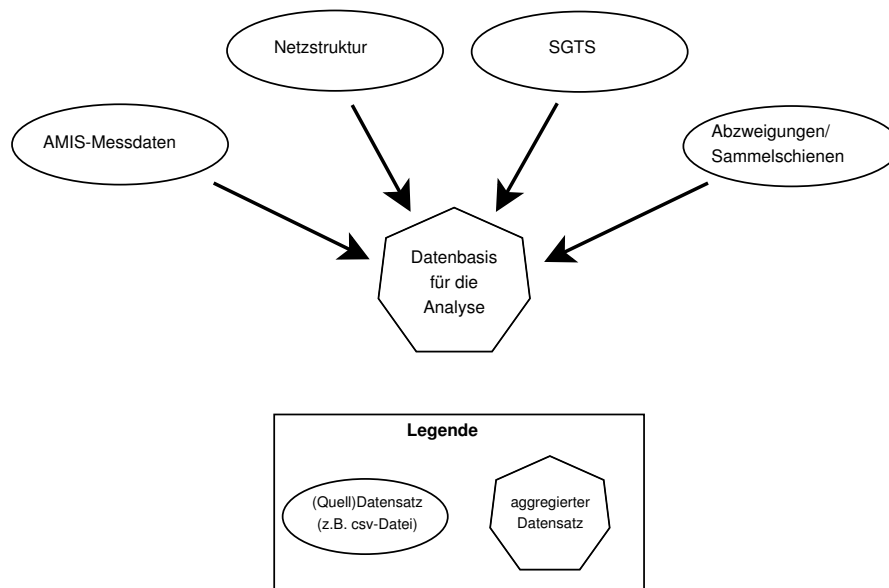


Abbildung 10: Schema des Zusammenführens der einzelnen Datensätze

Zum Zeitpunkt der Erstellung dieser Diplomarbeit befanden sich einige der Systeme bzw. Geräte, welche zur Erstellung der einzelnen Datensätze verwendet werden, erst in der Entwicklungs- bzw. in der Testphase. Daher enthielten die zur Verfügung stehenden Datensätze teilweise Fehler, wie z.B. fehlerhafte Daten oder Zeilenumbrüche in den

csv-Dateien. Weiters sind die Datensätze bzw. die Darstellung der einzelnen Datensätze (Dateiformat, vorhandene Spalten bzw. Bezeichnung der Spalten in csv-Dateien, ...), welche für die Testnetze vorlagen, nicht einheitlich. Da die zu erstellende Datenbasis für die Analyse die vorhandenen Fehler bzw. Unterschiede natürlich nicht enthalten soll gestaltet sich das Erstellen derselbigen aus den einzelnen Datensätzen durchaus anspruchsvoll.

In den folgenden Unterkapiteln erfolgt zuerst eine Beschreibung der vorhandenen Datensätze:

- * Datensatz „AMIS-Messdaten“
- * Datensatz „Netzstruktur“ (liegt in drei verschiedenen Varianten vor)
- * Datensatz „Smart Grids Topologie Simulator (SGTS)“
- * Datensatz „Abzweigungen/Sammelschienen“

Anschließend wird das Zusammenführen der einzelnen Datensätze zu einer aggregierten Datenbasis, welche für die Analyse benötigt wird, beschrieben.

3.1 Gemessene Daten

Wie bereits im Kapitel 2.2.1 (Seite 10) erwähnt, wurden im Rahmen des Projektes „DG DemoNet Smart LV Grid“ Testnetze mit AMIS Komponenten (Smart Meter, DC, Compact Switch Gear, Kompaktschaltanlage (CSG)) bestückt und damit Feldtests der Komponenten durchgeführt. Die für diese Diplomarbeit zur Verfügung stehenden AMIS-Messdaten werden in den DC (befindet sich in der Trafostation) der jeweiligen Testnetze auf jeweils einer Secure Digital Memory Card (SD-Karte) gespeichert. Auf diesen SD-Karten werden auch noch andere, für diese Diplomarbeit jedoch nicht benötigte, Daten (z.B. Firmwareversion, Fehlerlogs, Diagnosedaten, ...) gespeichert.

3.1.1 Testnetze

Die für diese Diplomarbeit zur Verfügung stehenden AMIS-Messdaten stammen aus sechs verschiedenen Ortsnetzen, welche von der Netz OÖ GmbH betrieben werden:

- * ATTNANG-PUCHHEIM (zwei Netze, Datensatz Netzstruktur Variante 1)
- * VÖCKLABRUCK (zwei Netze, Datensatz Netzstruktur Variante 1)
- * EBERSTALZELL (ein Netz, Datensatz Netzstruktur Variante 2 und 3)
- * LITTRING (ein Netz, Datensatz Netzstruktur Variante 2 und 3)

Die AMIS-Messdaten der ersten beiden Testnetze bzw. DC (ATTNANG-PUCHHEIM und VÖCKLABRUCK) stammen von einem früheren Zeitraum als jene AMIS-Messdaten der beiden letzteren Testnetze (EBERSTALZELL und LITTRING).

3.1.2 Datensatz „AMIS-Messdaten“

Wie bereits in Kapitel 3.1 beschrieben werden die AMIS-Messdaten in jedem DC auf jeweils einer SD-Karte gespeichert. Auf diesen besagten SD-Karten befindet sich eine

definierte Datei- und Ordnerstruktur, in welcher die aufgezeichneten Daten abgelegt werden. Für diese Diplomarbeit sind folgende zwei Datenbereiche relevant:

1. Alle csv-Dateien im Verzeichnis \DIAG\DLC\

2. Die Datei LOG-DLC.CSV im Hauptverzeichnis

Aus diesen Daten/Dateien wird in weiterer Folge der Datensatz „AMIS-Messdaten“ gebildet. Dabei ist zu beachten, dass die vom DC (Master) einem AMIS Endgeräte (Slave) zugeordnete Stationsnummer vom Master dynamisch vergeben wird. Deshalb kann im Laufe der Zeit ein AMIS Endgeräten eine andere Stationsnummer erhalten bzw. eine Stationsnummer einem anderen AMIS Endgeräte zugeordnet werden. Der genaue Vorgang des Zusammenführens der csv-Dateien zum Datensatz „AMIS-Messdaten“ und die Implementierung des dazu geschriebenen MATLAB Programms wird im Kapitel 3.4.3.1 (Seite 45) beschrieben.

csv-Dateien in \DIAG\DLC

In Abbildung 11 sind der Header und einige Zeilen einer csv-Beispieldatei abgebildet. Eine Beschreibung der Spalten dieser Datei beinhaltet Tabelle 1.

	A	B	C	D	E	F	G	H	I	J	K
1	Uhrzeit	Stat-Nr	Prior-Nr	Mode	Anzahl Hop	Laufzeit n*10ms	Quittung	Phase	SL	SNR	BER
2	12.12.2008 11:59:20:162	71	4	1	1	123	NEIN	L3	-1	-1	0
3	12.12.2008 11:59:21:391	72	4	6	2	88	JA	L3	11	69	38
4	12.12.2008 11:59:22:273	73	4	1	1	71	JA	L2	13	72	47
5	12.12.2008 11:59:22:981	74	4	6	2	89	JA	L2	13	70	39
6	12.12.2008 11:59:23:872	75	4	9	2	82	JA	L3	14	74	50
7	12.12.2008 11:59:24:692	76	4	9	1	51	JA	L2	14	71	38
8	12.12.2008 11:59:25:202	77	4	11	2	88	NEIN	L3	-1	-1	0
9	12.12.2008 11:59:26:080	78	4	9	2	78	JA	L3	13	75	46
10	12.12.2008 11:59:26:861	79	4	9	2	80	JA	L3	12	74	40
11	12.12.2008 11:59:27:662	1023	0	6	8	310	NEIN	L2	-1	-1	0

Abbildung 11: Header und Beispieldaten einer csv-Datei aus dem Verzeichnis \DIAG\DLC; Quelle: Netz Oberösterreich GmbH

Spaltenname	Beschreibung
Stationsnummer	ID der adressierten Slave Station, wird vom DC dynamisch vergeben
Status	Aktueller Status der Slave Station
Uhrzeit	Zeitstempel des AMIS Telegramms, Uhrzeit im DC beim Aussenden [AMI11, S. 54]
Ablauf-Zeit	Timeout für die der Slave Station zugeordnete Stationsnummer
Zieladresse	nicht verwendet/bekannt
Serial-Nr	Seriennummer welche vom Gerätehersteller vergeben wird
Serien-Nr	Seriennummer welche von der Netz Oberösterreich GmbH vergeben wird
FW-Typ nicht ladbar	nicht verwendet/bekannt
Revision nicht ladbar	nicht verwendet/bekannt
FW-Typ ladbar	nicht verwendet/bekannt
Revision ladbar	nicht verwendet/bekannt
Engeräte-CRC	nicht verwendet/bekannt
aktuelle Phase	Phase auf welcher das AMIS Telegramm vom Slave empfangen wurde
Prior Mode 0	nicht verwendet/bekannt
Hopanzahl Mode 0	nicht verwendet/bekannt
...	...
Prior Mode 15	nicht verwendet/bekannt
Hopanzahl Mode 15	nicht verwendet/bekannt

Tabelle 2: Beschreibung der Attribute einer LOG-DLC.CSV Datei

gesondert behandelt. Aus den zur Verfügung stehenden xlsx-Dateien entsteht in weiterer Folge der Datensatz „Netzstruktur“ (siehe Abbildung 10 auf Seite 18).

Eine weitere Beschreibung der Testnetze liegt in Form von Dateien im DIgSILENT-GIS-SCADA (dgs)-Format (siehe Kapitel 2.1.1 auf Seite 5) vor. Diese dgs-Dateien wurden von der Netz Oberösterreich GmbH zur Verfügung gestellt. Um die darin enthaltenen Daten in dieser Diplomarbeit nutzen zu können müssen die dgs-Dateien zuerst durch mehrere Konvertierungsschritte ins csv-Format konvertiert werden. In weiterer Folge entsteht aus diesen Daten der Datensatz „SGTS“ (siehe Abbildung 10 auf Seite 18). Eine genauere Beschreibung des Vorganges erfolgt in Kapitel 3.2.4 (Seite 26).

Weiters stehen noch Netzinformationen in Form von grafischen Ortsnetzplänen analog zu jenem in Abbildung 1 (Seite 5) zur Verfügung. Diese wurden ebenfalls von der Netz Oberösterreich GmbH zur Verfügung gestellt. Benötigt werden diese Ortsnetzpläne für die Erstellung des Datensatzes „Abzweigungen/Sammelschienen“ (siehe Abbildung 10 auf Seite 18). Eine genauere Beschreibung über die Erstellung dieses Datensatzes erfolgt in Kapitel 3.2.5 (Seite 27).

3.2.1 Datensatz „Netzstruktur“ (Variante 1)

Die erste Variante des Datensatzes Netzstruktur besteht aus einer xlsx-Datei mit zwei Arbeitsmappen (Worksheets). Es werden nun beispielhafte Ausschnitte aus zwei zusammengehörenden Arbeitsmappen, welche ein Netzsegment beschreiben und deshalb in der gleichen xlsx-Datei gespeichert sind, detailliert beschrieben.

Arbeitsmappe 1 - klassische Information über Netzanschlüsse

In Abbildung 13 sind der Header und einige Zeilen an Beispieldaten abgebildet, Tabelle 3 beinhaltet eine Beschreibung der Spalten der abgebildeten Arbeitsmappe.

	A	B	C	D	E
1	ON 06431 ATTNANG ROEMERSTR				
2	Stationsnr+Strang	Anschlussnummer	Ost (X)	Nord (Y)	Anmerkung
3	06431-01				
4	06	60025375			
5			13,7213		
6				48,0105	
7	06		13,7	48,0	
8	06		13,7	48,0	
9	06		13,7	48,0	
10	06		13,7	48,0	
11	06		13,7	48,0	
12	06		13,7	48,0	
13	06		13,7	48,0	

Abbildung 13: Zuordnung zu Ortsnetzplan - Variante 1, Arbeitsmappe 1;
Quelle: Netz Oberösterreich GmbH

Spaltenname	Beschreibung
Stationsnr+Strang	Nummer der Trafostation und des Stranges im Niederspannungsnetz
Anschlussnummer	Nummer des Hausanschlusses, von Netz Oberösterreich GmbH vergeben
Ost (X)	nicht verwendet/bekannt
Nord (Y)	nicht verwendet/bekannt
Anmerkung	Zusatzinformation, z.B. „keine Zähler vorhanden!“

Tabelle 3: Beschreibung der Attribute der Arbeitsmappe mit klassischen Informationen von Netzanschlüsse

Arbeitsmappe 2 - Smart Meter Information über Netzanschlüsse

Die Informationen, welche in dieser Arbeitsmappe enthalten sind, enthalten Angaben über die in den jeweiligen Hausanschlüssen verwendeten AMIS Endgeräte. In Abbildung 14 sind der Header und einige Zeilen an Beispieldaten abgebildet. In Tabelle 4 erfolgt eine Beschreibung der Spalten der in Abbildung 14 abgebildeten Arbeitsmappe.

3.2.2 Datensatz „Netzstruktur“ (Variante 2)

Im Gegensatz zu Variante 1 ist bei dieser Variante eine xlsx-Datei mit lediglich einer Arbeitsmappe vorhanden. Die darin enthaltenen Informationen unterscheiden sich auch inhaltlich zur Variante 1. Die einzelne Arbeitsmappe, welche in Variante 2 vorhanden ist, stellt somit nicht lediglich eine Aggregation der Inhalte der beiden Arbeitsmappen der xlsx-Datei von Variante 1 dar. So ist z.B. in Variante 2 die Information serialNr (siehe Tabelle 9 auf Seite 31) in der xlsx-Datei nicht enthalten, was in weiterer Folge beim Zusammenführen der Datensätze zur Datenbasis für die Analyse ein anderes Vorgehen im Vergleich zu Variante 1 erforderlich macht. In Abbildung 15 sind der Header und einige

	A	B	C	D	E	F	G	H	I
1	ON 06431 ATTNANG ROEMERSTR								
2	Stationsnr neu	Stationsname	Stangnr neu	Anschlussobjektnr	Funktionsklasse	Serialnr Hersteller	Serialnr EnergieAG	Anzahl der Anschlussobjekte	Anzahl der Zählpunkte
3	06431	ATTNANG ROEMERSTR	06431-01		AMIS			1	1
4						G23-512-C.21/049-000252			
5							3020230		
6					AMIS	G23-	3 49	1	1
7					AMIS	G23-	1 30	1	1
8					AMIS	G23-	7 49	1	1
9					AMIS	G23-	1 42	1	1
10					AMIS	G23-	8 42	1	1
11					AMIS	G23-	1 49	1	1
12					AMIS	G23-	9 42	1	1
13					AMIS	G23-	5 44	1	1
14					AMIS	G23-	2 42	1	1
15					AMIS	G23-	5 49	1	1
16					AMIS	G23-	3 42	1	1
17					AMIS	G23-	8 30	1	1
18				60025375					
19					AMIS	G23-	1 30	1	1
20					AMIS	G23-	7 30	1	1

Abbildung 14: Zuordnung zu Ortsnetzplan - Variante 1, Arbeitsmappe 2;
Quelle: Netz Oberösterreich GmbH

Spaltenname	Beschreibung
Stationsnr neu	Nummer der Trafostation
Stationsname	Name der Trafostation
Stangnr neu (sic!)	Strangnummer (Tippfehler in Datei!)
Anschlussobjektnr	Nummer des Hausanschlusses, von Netz Oberösterreich GmbH vergeben
Funktionsklasse	Smart Meter Einordnung
Serialnr Hersteller	Seriennummer welche vom Gerätehersteller vergeben wird
Serialnr EnergieAG	Seriennummer welche von der Netz Oberösterreich GmbH vergeben wird
Anzahl der Anschlussobjekte	nicht verwendet/bekannt
Anzahl der Zählpunkte	nicht verwendet/bekannt

Tabelle 4: Beschreibung der Attribute der Arbeitsmappe mit Smart Meter Informationen von Netzanschlüssen

Zeilen an Beispieldaten einer xlsx-Datei der Variante 2 abgebildet. Tabelle 5 beinhaltet eine Beschreibung der Spalten der in Abbildung 15 abgebildeten Arbeitsmappe.

	A	B	C	D	E	F	G	H
1	ON 01471 EBERSTALZELL							
2	Stationsnr neu	Stationsname	Strangnr alt	Strangnr neu	Anschlussobjektnr	Funktionsklasse	Serialnr Hersteller	
3	01471	EBERSTALZELL	440409201	01471-01				
4						AMIS	G23-511-C.01/948-001889	
5						AMIS	G23-	1
6					60240881	AMIS	G23-	0
7					6 2	AMIS	G23-	5
8					6 3	AMIS	G23-	6
9					6 4	AMIS	G23-	5
10					Ergebnis			
11					6 5	AMIS	G23-	5

Abbildung 15: Ausschnitt von Zuordnung zu Ortsnetzplan; Quelle: Netz Oberösterreich GmbH

Spaltenname	Beschreibung
Stationsnr neu	Nummer der Trafostation
Stationsname	Name der Trafostation
Strangnr alt	Strangnummer - altes Schema
Stangnr neu (sic!)	Strangnummer - neues Schema(Typo in Datei!)
Anschlussobjektnr	Nummer des Hausanschlusses, von Netz Oberösterreich GmbH vergeben
Funktionsklasse	Smart Meter Einordnung
Serialnr Hersteller	Seriennummer welche vom Gerätehersteller vergeben wird

Tabelle 5: Beschreibung der Attribute der Arbeitsmappe mit Daten von Netzanschlüssen

3.2.3 Datensatz „Netzstruktur“ (Variante 3)

Bei dieser Variante des Datensatzes Netzstruktur liegt eine csv-Datei vor. Die darin enthaltenen Informationen sind nominell zwar geringer als bei den beiden anderen Varianten, jedoch sind die für das Zusammenführen der Datensätze zur Datenbasis für die Analyse notwendigen Daten alle enthalten. In Abbildung 16 sind der Header und einige Zeilen an Beispieldaten einer csv-Datei der Variante 3 abgebildet. Tabelle 6 beinhaltet eine Beschreibung der Spalten der in Abbildung 16 abgebildeten csv-Datei.

	A	B	C
1	Seriennummer EAG	Objektanschlussnummer	Seriennummer AMIS
2	3035190		
3		60241131	
4			G23-511-C.01/145-004632
5	30 8	60 6	G23-5 2
6	30 2	60 7	G23-5 6
7	30 0	60 8	G23-5 4
8	30 4	60 9	G23-5 8
9	44 9	60 4	G23-5 1
10	30 6	60 4	G23-5 0
11	30 3	60 5	G23-5 7

Abbildung 16: Ausschnitt aus Zuordnung zu Ortsnetzplan - Variante 3;
Quelle: Netz Oberösterreich GmbH

Spaltenname	Beschreibung
Seriennummer EAG	Seriennummer welche von der Netz Oberösterreich GmbH vergeben wird
Objektanschlussnummer	Nummer des Hausanschlusses, von Netz Oberösterreich GmbH vergeben
Seriennummer AMIS	Seriennummer welche vom Gerätehersteller vergeben wird

Tabelle 6: Beschreibung der Attribute der csv-Datei mit Informationen von Netzanschlüssen

3.2.4 Datensatz „Smart Grids Topologie Simulator“

Dieser Datensatz liegt nicht wie die bisher beschriebenen direkt, d.h. in Form von Text in xlsx bzw. csv-Dateien vor, sondern der verarbeitbare Plaintext muss erst aus einer dgs-Datei erzeugt werden. In Abbildung 17 ist der Ablauf wie aus einer dgs-Datei eine csv-Datei wird grafisch dargestellt.

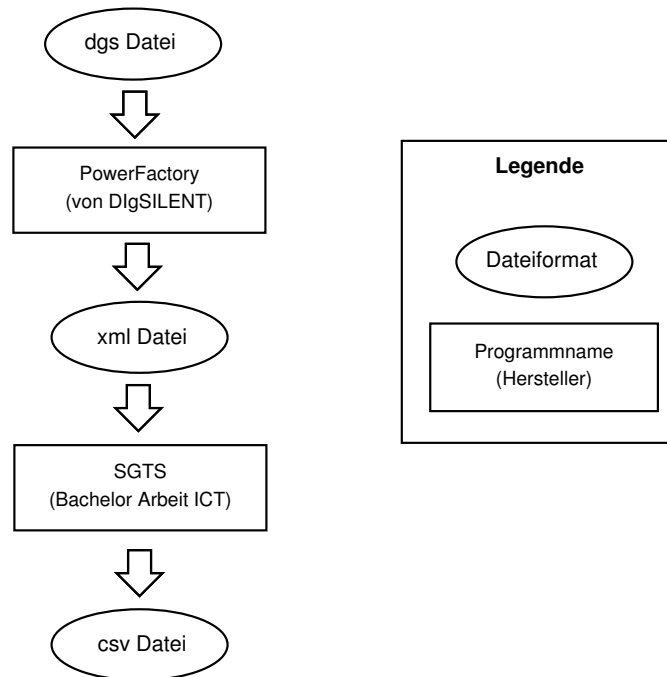


Abbildung 17: Konvertierungsschritte vom dgs-Format ins csv-Format

Da mit dem Programm PowerFactory die notwendigen Informationen nicht direkt in eine csv-Datei exportiert werden können ist es notwendig als Zwischenschritt zuerst eine xml-Datei zu erstellen. Diese xml-Datei ist nach dem Muster einer xml-Datei, wie sie in Abbildung 3 (Seite 6) dargestellt ist, aufgebaut. Von dieser exportierten xml-Datei wird mit dem Programm SGTS (siehe Kapitel 2.2.3 auf Seite 15) anschließend die gewünschte csv-Datei erzeugt. Da die vorliegende Version des SGTS etwaige geöffnete Trennschalter nicht berücksichtigt, muss die xml-Datei eventuell manuell editiert werden, da sonst Leitungslängen eventuell falsch berechnet werden. So sind z.B. bei Ringleitungen Einträge für offene Trennschalter aus der xml-Datei herauszulöschen. Ohne diese manuelle Bearbeitung der xml-Datei entspricht die berechnete Leitungslänge zwischen zwei Anschlüssen bei mehreren vorhanden Verbindungswegen eventuell nicht der Realität, da das Ergebnis davon abhängt welcher Weg zuerst berechnet wird.

In Abbildung 18 (Seite 27) sind der Header und einige Zeilen an Beispieldaten einer SGTS Output Datei (csv-Format) abgebildet. Tabelle 7 (Seite 27) beinhaltet eine Beschreibung der Spalten der in Abbildung 18 abgebildeten csv-Datei.

	A	B	C	D	E
1	*****				
2	* *				
3	* Knotenliste *				
4	* *				
5	*****				
6	Knotenname	DC	Hop-Ebene	Entfernung	Daempfung
7	DA_61665_60240751	false	-1 0.0		0.0
8	MA_616	false	-1 0.0		0.0
9	DA_616 52	false	-1 0.0		0.0
10	DA_616 69	false	-1 0.0		0.0
21	DA_434 28	false	0 0.6600000057369471		-13.200000114738941
22	DA_434 39	false	0 0.45500000566244125		-9.100000113248825
23	KÜMA305171b	false	0 0.4150000065565109		-8.300000131130219
24	DA_434 0	false	0 0.4840000048279762		-9.680000096559525
25	DA_434 1	false	0 0.5090000052005053		-10.180000104010105
26	DA_434 2	false	0 0.5380000043660402		-10.760000087320805
27	DA_434 3	false	0 0.5660000052303076		-11.320000104606152
28	DA_434 4	false	0 0.5890000052750111		-11.780000105500221
29	RICHTUNG_KÜMA305171rechts	false	0 0.39600000716745853		-7.920000143349171
30	RICHTUNG_DA61665	false	-1 0.0		0.0
31		60240894	false	0 0.17900000512599945	-3.580000102519989
32	KÜDA61375_60240895	false	0 0.15400000475347042		-3.0800000950694084
33		false	0 0.17500000074505806		-3.500000014901161
34	KK49829	false	0 0.11099999770522118		-2.2199999541044235
35		false	0 0.5000000093132257		-10.000000186264515

Abbildung 18: Header und Beispieldaten einer SGTS Output Datei

Spaltenname	Beschreibung
Knotenname	Bezeichnung des Knotens, siebenstellige Nummer → Nummer des Hausanschlusses
DC	Ist Knoten ein Data Concentrator (DC)?
Hop-Ebene	Berechnung direkt über Dämpfung (welche direkt der Entfernung entspricht)! (siehe Kapitel 2.2.3 auf Seite 15)
Entfernung	Entfernung von Knoten zu gewähltem Root Knoten (→ Data Concentrator (DC)) (Root Knoten muss beim Programmaufruf ausgewählt werden)
Daempfung	nur Multiplikation der Entfernung mit konstantem Faktor (z.B. 20dB pro km)! (siehe Kapitel 2.2.3 auf Seite 15)

Tabelle 7: Beschreibung der Attribute einer SGTS Output Datei

3.2.5 Datensatz „Abzweigungen/Sammelschienen“

Für die Erstellung dieses Datensatzes kommen zwei Möglichkeiten in Betracht. Die erste Möglichkeit ist die Abzweigungen und Sammelschienen auf den jeweiligen Leitungen zwischen dem DC und den Hausanschlüssen (siehe Abbildung 19 auf Seite 28) von Hand abzuzählen und manuell in eine csv-Datei einzutragen.

Die zweite Möglichkeit besteht darin, die Anzahl der Abzweigungen und Sammelschienen mit den bereits für die Entfernung als Informationsquelle verwendeten xml-Dateien (Kapitel 3.2.4 auf Seite 26) zu berechnen. Da die vorliegende Version des Smart Grids Topologie Simulator (SGTS) dies jedoch noch nicht unterstützt wurde für zwei Testnetze

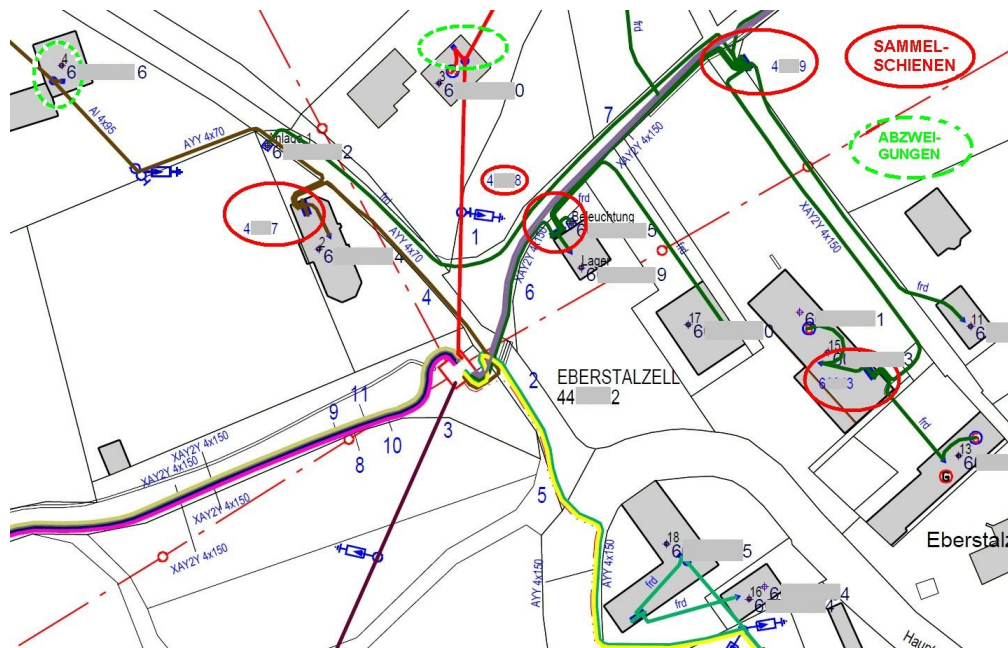


Abbildung 19: Informationsquelle für Daten Datei Abzweigungen/Sammelschienen; Quelle: Netz Oberösterreich GmbH

(EBERSTALZELL und LITTRING) die erste (manuelle) Möglichkeit gewählt und die csv-Datei manuell erstellt.

In Abbildung 20 (Seite 28) sind der Header und einige Zeilen an Beispieldaten einer selbst erstellten Datendatei Abzweigungen/Sammelschienen (csv Format) abgebildet. Tabelle 8 (Seite 29) beinhaltet eine Beschreibung der Spalten der in Abbildung 20 abgebildeten csv-Datei.

	A	B	C
1	EBERSTALZELL		
2			
9			
10	connectionObjectNr	nrOfBusBars	nrOfConnections
11		0	
12			12
13	60240749		
14	602		
15	602		
16	602		
17	602		

Abbildung 20: Header und Beispieldaten einer Datei Abzweigungen/Sammelschienen

Spaltenname	Beschreibung
connectionObjectNr	Nummer des Hausanschlusses, von Netz Oberösterreich GmbH vergeben
nrOfBusBars	Anzahl der Sammelschienen zwischen Trafostation und Hausanschluss
nrOfConnections	Anzahl der Abzweigungen zwischen Trafostation und Hausanschluss

Tabelle 8: Beschreibung der Attribute einer Datendatei Abzweigungen/Sammelschienen

3.3 Zusammenführen der Datensätze

Um die einzelnen Datensätze durch das in Kapitel 3.4.3 (Seite 42) beschriebene MATLAB Programm zur Datenbasis für die Analyse zusammenführen zu können müssen die einzelnen Quelldateien erstmal durch das MATLAB Programm eingelesen werden. Um das Einlesen zu vereinfachen werden lediglich csv-Dateien eingelesen. Alle im xlsx, xml und dgs-Format vorliegenden Dateien müssen deshalb zuerst in csv-Dateien umgewandelt werden. Bei jenen xlsx-Dateien welche mehrere Arbeitsmappen beinhalten (siehe Kapitel 3.3.1 auf Seite 30) wird für jede Arbeitsmappe eine äquivalente csv-Dateien erstellt. Jede der erstellten csv-Dateien erhält einen Anhang an den Dateinamen (z.B. `_sheet_1` vor der Dateierweiterung `.csv`) durch welchen jede csv-Dateie eindeutig der ursprünglichen Arbeitsmappe zugeordnet werden kann. Die entsprechenden csv-Dateien enthalten zusammen die gleiche Information wie ihre xlsx-Quelldatei, die Umwandlung dient ausschließlich dem leichteren Einlesen der Daten durch das im Kapitel 3.4.3 (Seite 42) beschriebene MATLAB Programm.

In den folgenden Unterkapiteln wird das Zusammenführen aller vorhandenen Datensätze zur Datenbasis für die Analyse (siehe Abbildung 10 auf Seite 18), detailliert beschrieben. Da die Datenquellen teilweise in unterschiedlichen Formaten vorliegen (csv-Dateien, xlsx, xml, dgs) und auch von verschiedenen Partnern zur Verfügung gestellt werden, ist das Zusammenführen der Datensätze ein aufwändiger Prozess. Dies kommt zu einem großen Teil daher, weil kein einheitliches Datenmodell, und oft auch keine einheitlichen Bezeichnungen, verwendet werden. Da die Bezeichnungen in den verschiedenen Datenquellen teilweise relativ ähnlich sind besteht auch noch die Gefahr einer Verwechslung beim Konfigurieren der Matching Parameter. Mit Matching Parameter ist hier die Bezeichnung der Parameter in den verschiedenen csv-Dateien gemeint, welche die selbe Bedeutung haben, jedoch unterschiedliche Namen. Für eine genaue Übersicht der synonymen Bezeichnungen in den verschiedenen Dateien siehe Tabelle 9 (Seite 31). Die genannte Tabelle zeigt, wenn man eine Reihe davon liest, die Synonymität der verschiedenen Bezeichnungen, welche in den einzelnen Datenquellen (\rightarrow csv-Dateien) und dem Datensatz „Gesamtdaten für Analyse“ (\rightarrow MATLAB Programm) verwendet werden. Außerdem kann man in den Spalten ablesen welche Informationen in welchen Datenquellen (\rightarrow csv-Dateien) vorhanden sind. Um die Tabelle 9 einfacher lesen zu können ist es vorteilhaft sie zusammen mit Abbildung 21 (Seite 32) zu lesen.

Um einer Verwechslung entgegenzuwirken werden für die „Datenbasis für Analyse“ eigene Bezeichnungen eingeführt, welche nach dem Einlesen der csv-Dateien in das erstellte

MATLAB Programm im Source Code des MATLAB Programms ausschließlich verwendet werden. Ein weiterer Vorteil der eigenen Bezeichnungen ist, dass bei diesen durchgehend als Sprache Englisch und als Schreibweise „lowerCamelCase“ verwendet wird. Dadurch wird eine weitere Verwendung des in dieser Diplomarbeit erstellten MATLAB Programms erleichtert. In den folgenden beiden Unterkapiteln folgt nun eine schematische Beschreibung des Zusammenführens der einzelnen Datensätze zur „Datenbasis für Analyse“. Eine detaillierte Beschreibung des Vorganges, auch eine teilweise Beschreibung der Implementierung des dazu erstellten MATLAB Programms, erfolgt daran anschließend im Kapitel 3.4.2 (Seite 40).

3.3.1 Zusammenführen mit Datensatz „Netzstruktur“ (Variante 1)

Es folgt nun eine schematische Beschreibung des Zusammenführens der Datensätze für den Fall, dass der Datensatz Netzstruktur in Variante 1 (genaue Beschreibung der Variante 1 siehe Kapitel 3.2.1 auf Seite 22) vorliegt. In Abbildung 21 ist der (geplante) Ablauf dargestellt nach welchem die einzelnen Datensätze im MATLAB Programm zusammengefügt werden und sich nach und nach zur Datenbasis für die Analyse entwickelt. Die für das Matching zwischen den einzelnen Datensätzen bzw. den dazugehörigen csv-Dateien zur Verfügung stehenden (→ in beiden Datensätzen bzw. csv-Dateien vorhandenen) Variablen lassen sich am besten aus Tabelle 9 (Seite 31) herauslesen. In dieser Tabelle ist auch die in den jeweiligen csv-Dateien verwendete Bezeichnung festgehalten. Der Grund, dass in Abbildung 21 (Seite 32) der unterste Teil als geplant eingezeichnet ist liegt darin begründet, dass dieser weitere Schritt um letzten Endes zur Entfernung zwischen DC und AMIS-Endgerät zu kommen nicht durchgeführt/implementiert wurde. Dies ist deshalb der Fall, da nach einem ersten Datencheck der AMIS-Messdaten von jenen Testnetzen, für die zum damaligen Zeitpunkt AMIS-Messdaten und Daten über die Netzstruktur zur Verfügung standen, festgestellt wurde, dass diese nicht verwendet werden können.

Das Problem mit den damals zur Verfügung stehenden AMIS-Messdaten war, dass darin keine Messwerte von auch im Datensatz Netzstruktur vorhanden AMIS-Endgeräten enthalten sind. Somit könnte mit diesen AMIS-Messdaten lediglich eine statistische Auswertung ohne Einbeziehung der Netzstruktur erfolgen, was jedoch als nicht zufriedenstellend (so etwas wurde am ICT bereits durchgeführt, siehe Kapitel 2.2.2, Seite 12) für diese Diplomarbeit eingestuft wurde. Aber es war zum damaligen Zeitpunkt schon absehbar, dass in naher Zukunft neue AMIS-Messdaten von anderen Testnetzen zur Verfügung stehen würden, mit welchen das ursprüngliche Ziel weiter verfolgt werden konnte. Da die Daten zur Netzstruktur für diese neuen Testnetze jedoch ausschließlich in einer anderen Variante (→ Datensatz Netzstruktur Variante 2) zur Verfügung standen war auch eine andere Vorgehensweise beim Erstellen der Datenbasis für die Analyse erforderlich. Diese wird nun im nächsten Unterkapitel beschrieben.

Datenbasis für die Analyse (MATLAB Programm intern)	AMIS-Messdaten		Netzstruktur			SGTS	Abzweigungen/ Sammelstienen
	Verzeichnis \DIAG\DL\	LOG-DL\C.VSY	Variante 1	Variante 2	Variante 3		
time	Uhrzeit	Uhrzeit	sheet_1				
stationNr	Stat-Nr	Stationsnummer					
priorityNr	Prior-Nr						
mode	Mode						
hopCount	Anzahl Hop						
delay	Laufzeit n°10ms						
ack	Quittung						
phase (1 2 3)	aktuelle Phase (L1 L2 L3)						
sur	SNR						
serialNr	Serial-Nr (scl) (e.g. 4098231)	Serialur EnergieAG			Seriennummer EAG		
serialNrProducer	Serien-Nr (scl) (e.g. G23-512-C-21/049-000273)	Serialur Hersteller			Seriennummer AMIS		
connectionObjectNr		Abschlussummer			Objektauswahlnummer	Knotenname (alle Knoten) Entfernung	connectionObjectNr
distance							
nOfBusBars							nOfBusBars
nOfConnections							nOfConnections
weekDay							
hourOfDay							
functionClass			Funktionsklasse		Funktionsklasse		
NICHT VERWENDETE DATEN!							
info	Zusatzinfo						
sl	SL						
ber	BER						
cordFullNr		Stationsnr+Strang			Strangnr neu in manchen Dateien „Strangnr neu“ (sic)		
connectionObjectNrCoordX		Ost (X)					
stationNrCoordY		Nord (Y)					
stationNameGeographic					Stationsname		
cordFullNrOld					Strangnr alt		
						Hop-Ebene (Berechnung direkt über Entfernung) Dämpfung (Berechnung direkt über Entfernung)	
status	Status						
	Ablaufzeit						
	Zieladresse						
	FW-Typ nicht ladbar						
	Revision nicht ladbar						
	FW-Typ ladbar						
	Revision ladbar						
	Emergenc-CHC						
	Prior Mode 0...15						
	ATYP-ABLESE 0...15	Anmerkung					
					Strangnr neu		
					Stationsname		
					Anzahl der Anschlussobjekte		
					Anzahl der Zählpunkte		
							DC

Tabelle 9: Inhalt der verschiedenen Datenquellen und Synonymität der Bezeichnungen

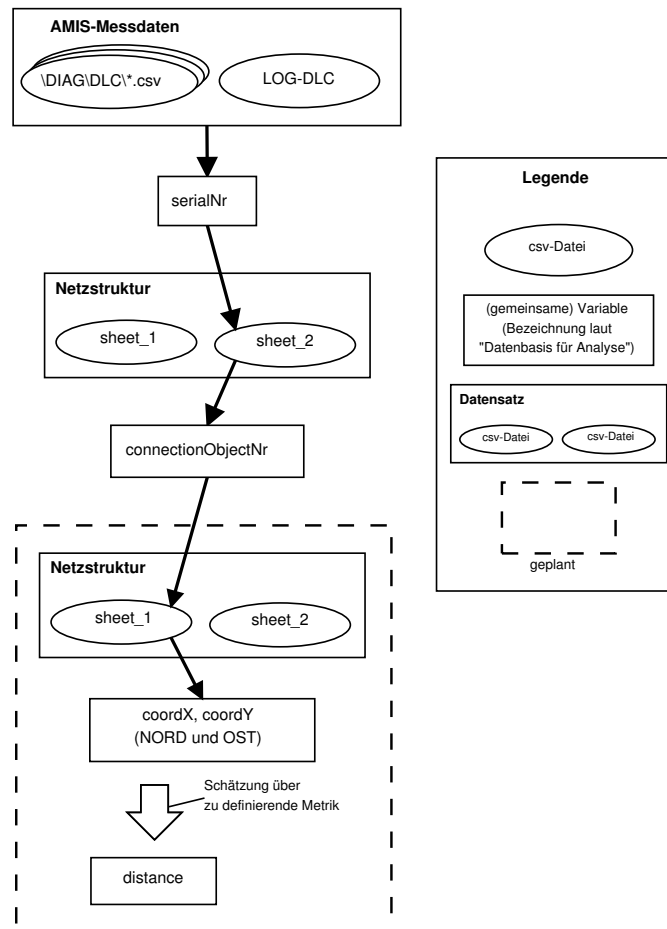


Abbildung 21: Zusammenführen der Datensätze (Datensatz Netzstruktur Variante 1)

3.3.2 Zusammenführen mit Datensatz „Netzstruktur“ (Variante 2)

Nun erfolgt die Beschreibung des Zusammenführens der Datensätze für den Fall, dass der Datensatz Netzstruktur in Variante 2 (genaue Beschreibung der Variante 2 siehe Kapitel 3.2.2 auf Seite 23) vorliegt. So wie im vorigen Kapitel 3.3.1 detailliert beschrieben beinhaltet Tabelle 9 auch die Bezeichnungen der für das Matching zur Verfügung stehenden Variablen für den Datensatz Netzstruktur (Variante 2). Um den Vorgang des Zusammenführens der einzelnen Datensätze bei Vorliegen des Datensatzes Netzstruktur (Variante 2) einfach verstehen zu können ist es vorteilhaft, die Tabelle 9 (Seite 31) und Abbildung 22 (Seite 33) zu verwenden.

Da im Datensatz Netzstruktur (Variante 2) im Gegensatz zur Variante 1 keine Daten über die serialNr (→ Seriennummer welche von der Netz Oberösterreich GmbH vergeben wird) vorhanden sind muss das Mapping über die serialNrProducer (→ Seriennummer welche vom Gerätehersteller vergeben wird) durchgeführt werden. Für die Implementie-

zung des MATLAB Programmes hat das den Nachteil, dass deswegen ein Vergleich von Strings (im Gegensatz zu einem Vergleich von Zahlen durchgeführt werden muss.

Außerdem ist auch keine Information über die geografische Lage des Anschlusses (und somit des AMIS-Endgerätes) vorhanden, welche für eine Schätzung der distance (Leitungslänge zwischen DC und AMIS-Endgerät) erforderlich wäre (siehe Abbildung 21). Um dieses Problem zu lösen wird das Programm Smart Grids Topologie Simulator (SGTS) verwendet, welches wie bereits in Kapitel 3.2.4 (26) beschrieben, aus einer vorliegenden dgs-Datei unter anderem die Leitungslänge zwischen einem DC und Hausanschlüssen berechnen und als Ausgabe eine csv-Datei erstellen kann. Gegenüber der für Variante 1 geplanten Version der Ermittlung der Leitungslänge zwischen DC und AMIS-Endgeräten hat dies auch den Vorteil, dass die Leitungslänge wesentlich genauer ist (Berechnung aus Netzdaten in Variante 2 im Gegensatz zu einer Schätzung anhand der geografischen Lage der Endpunkte wie sie in Variante 1 geplant war).

Das Hinzufügen der Daten, welche im Datensatz Abzweigungen/Sammelschienen (siehe Kapitel 3.2.5, Seite 27) vorhanden sind, erfolgt analog zu den bisher beschriebenen Datensätzen. Da dieser Datensatz selbst erstellt wurde, konnte er auf die eigenen Bedürfnisse bezüglich des Zusammenführens mit den anderen Datensätzen ausgelegt werden.

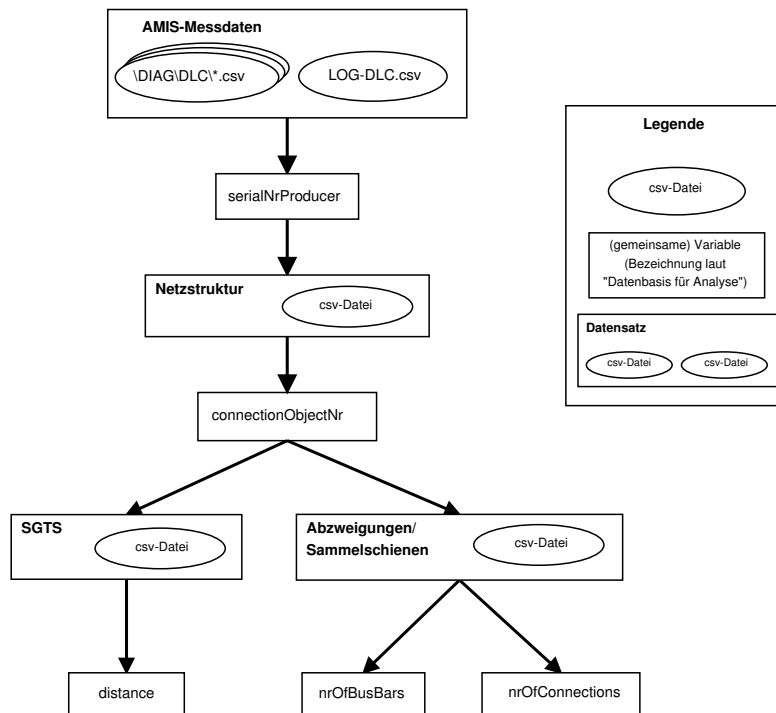


Abbildung 22: Zusammenführen der Datensätze (Datensatz Netzstruktur Variante 2/3)

3.3.3 Zusammenführen mit Datensatz „Netzstruktur“ (Variante 3)

Das Zusammenführen der Datensätze für den Fall, dass der Datensatz Netzstruktur in Variante 3 vorliegt, erfolgt gleich wie für Variante 2. Wie in Abbildung 22 dargestellt erfolgt das Matching der Datensätze AMIS-Messdaten und Netzstruktur über die gemeinsamen Daten von `serialNrProducer`. Wenn man jedoch die Tabelle 9 (Seite 31) genau betrachtet fällt auf, dass auch ein Matching über die gemeinsamen Daten von `serialNr` möglich wäre. Dieser Ablauf würde jedoch einen Mehraufwand (neue Methode bzw. Erweiterung der bestehenden Methode) in der MATLAB Programmimplementierung bedeuten. Die implementierte MATLAB Methode, welche zum Zusammenführen der Datensätze AMIS-Messdaten und Netzstruktur Variante 2 verwendet wird, kann jedoch direkt auch für das Zusammenführen mit dem Datensatz Netzstruktur Variante 3 verwendet werden. Dieser Vorteil überwiegt die Tatsache, dass das Matching über den Parameter `serialNr` etwas schneller wäre (Vergleich von Zahlen bei `serialNr` gegenüber Vergleich von Strings bei `serialNrProducer`).

3.4 Implementierung des MATLAB Programms (Datenaufbereitungsteil)

Wie bereits mehrfach erwähnt, wird die Datenaufbereitung und die daran anschließende Analyse mit einem eigens dafür erstellten MATLAB Programm durchgeführt. Um ein besseres und leichteres Verständnis für die im Kapitel 4 (Seite 50) beschriebene Analyse der erstellten Datenbasis zu haben erfolgt in diesem Kapitel lediglich die Beschreibung jenes MATLAB Codes, welcher die Implementierung der Datenaufbereitung enthält. Anders gesagt wird nun jenes MATLAB Programm beschrieben, welches die Implementierung des bis jetzt in diesem Kapitel (3) beschriebenen Vorganges, welcher als zusammengefasstes Schema in Abbildung 10 (Seite 18) darstellt ist.

Quasi-UML Diagramme

Da für das erstellte MATLAB Programm als Designkonzept die objektorientierten Programmierweise (für diese wurde Unified Modeling Language (UML) entwickelt) gewählt wurde, macht eine an UML angelehnte Darstellung durchaus Sinn, auch wenn aufgrund einiger MATLAB spezifischer Besonderheiten für die Diagramme keine reine UML Spezifikation verwendet wird. In Abbildung 23 ist ein beispielhaftes Quasi-UML Diagramm abgebildet, welches einige Beschreibungen für das Verständnis der in den folgenden Kapiteln verwendeten Quasi-UML Diagramme der MATLAB Implementierung enthält.

`ExampleClass_1` in Abbildung 23 (Seite 35) zeigt eine UML-konforme Klasse mit einer kurzen, in und neben der Klasse ersichtlichen, Beschreibung.

`ExampleClass_2` derselben Abbildung enthält mit `method_1` eine beispielhafte Darstellung einer Methode, welche keine Aufrufparameter und auch keinen Rückgabewert enthält. Solch eine Methode kann z.B. verwendet werden um eine Klassenvariable (hier `variable_1`) zu verändern, z.B. diese bei jedem Aufruf zu inkrementieren.

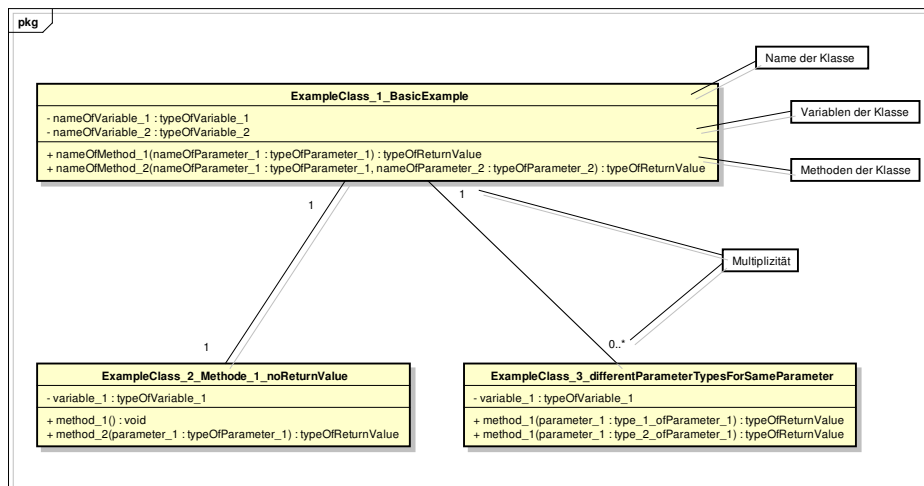


Abbildung 23: Beispiel eines Quasi-UML Diagrammes für MATLAB Klassen

Eine andere, bei der Implementierung des MATLAB Programmes für diese Diplomarbeit verwendete, Funktionalität zeigt die Klasse `ExampleClass_3` derselben Abbildung. Da von MATLAB bei Methoden der Typ der übergebenen Parameter nicht überprüft wird kann, falls dies gewünscht ist, für verschiedene Datentypen eines an die Methode übergebenen Parameters (hier `parameter_1` von `method_1`) ein anderer Algorithmus ausgeführt werden (\rightarrow Method/Function Overloading). Um diese Funktionalität in dem Quasi-UML Diagramm abzubilden ist die Methode im Diagramm mehrfach vorhanden. Die einzelnen Varianten unterscheiden sich im Quasi-UML Diagramm durch die unterschiedlichen Typen desselben Eingangsparameters (`type_1_ofParameter_1` und `type_2_ofParameter_1`). Für eine beispielhafte Implementierung sei auf die Methode `removeRowWhereValueEqual` verwiesen, an welche der Parameter `value` als Float oder String übergeben werden kann (siehe Abbildung 34 auf Seite 44). Im Sourcecode wird diese Funktionalität dadurch erreicht, dass in der Methode `removeRowWhereValueEqual` eine Typabfrage für den Parameter `value` durchgeführt wird und je nach Ergebnis ein anderer Algorithmus ausgeführt wird. Die Methode `removeRowWhereValueEqual` gibt es jedoch im MATLAB Quellcode nur ein einziges Mal.

Zusätzlich zu den implementierten MATLAB Methoden wurden eigene Datenstrukturen definiert. Abbildung 24 (Seite 36) beinhaltet eine beispielhafte Beschreibung eines dazu verwendeten, ebenfalls an UML angelehnten, Diagrammes.

In den folgenden Kapiteln erfolgt zuerst eine Beschreibung der selbst definierten Datenstrukturen und anschließend eine Beschreibung der implementierten Methoden. Diese Abfolge ist deshalb sinnvoll, da für ein leichteres Verständnis der implementierten Methoden Kenntnisse über die selbst erstellten Datenstrukturen von Vorteil sind.

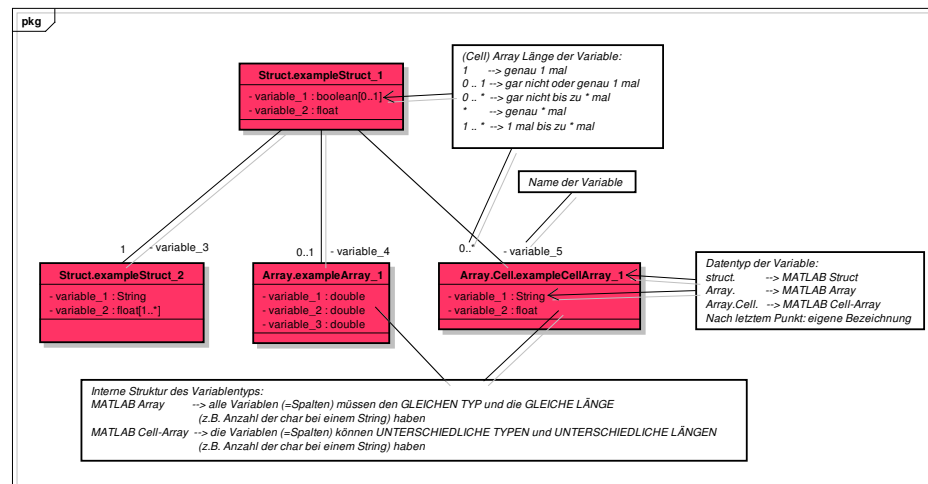


Abbildung 24: Beispiel eines Quasi-UML Diagrammes für MATLAB Datenstrukturen

3.4.1 Verwendete Datenstrukturen

Um die Erstellung der Datenbasis für die Analyse und die anschließend damit durchgeführte Analyse effizient durchführen zu können wurden eigene MATLAB Datenstrukturen definiert. Zur MATLAB internen Speicherung der Daten, welche in den einzelnen Datensätzen (siehe Abbildung 10, Seite 18) vorhanden sind, wird die in Abbildung 25 (Seite 37) schematisch beschriebene Datenstruktur definiert. Hierbei bezeichnet „myDataset“ nicht eine Variable bzw. Instanz einer von MATLAB selbst zur Verfügung gestellte Datenstruktur vom Typ dataset, sondern stellt die Generalisierung der selbst definierten Datenstrukturen dar. Die MATLAB interne Datenstruktur dataset würde zwar auch die benötigte Funktionalität bereit stellen, jedoch ist die Performance (im Sinne von Geschwindigkeit beim Zugriff auf die Datenstruktur) im Gegensatz zu der selbst definierten Datenstruktur myDataset, welche im Wesentlichen ein MATLAB Struct ist der MATLAB Arrays und MATLAB Cell Arrays beinhaltet, erheblich schlechter. Es folgt nun eine detaillierte Beschreibung der einzelnen, in Abbildung 25 schematisch abgebildeten, Datenstrukturen.

3.4.1.1 Repräsentation des Datensatzes „AMIS-Messdaten“

Diese Datenstruktur ist darauf ausgelegt die Daten des Datensatzes AMIS-Messdaten zu speichern. Da der eben genannte Datensatz aus mehreren Datenquellen (→ csv-Dateien) besteht (siehe Kapitel 3.1.2 auf Seite 19), sind zu deren Speicherung auch zwei verschiedene MATLAB Datenstrukturen vorhanden. Das Zusammenführen dieser beiden Datenstrukturen wird in Kapitel 3.4.3.1 (Seite 45) beschrieben.

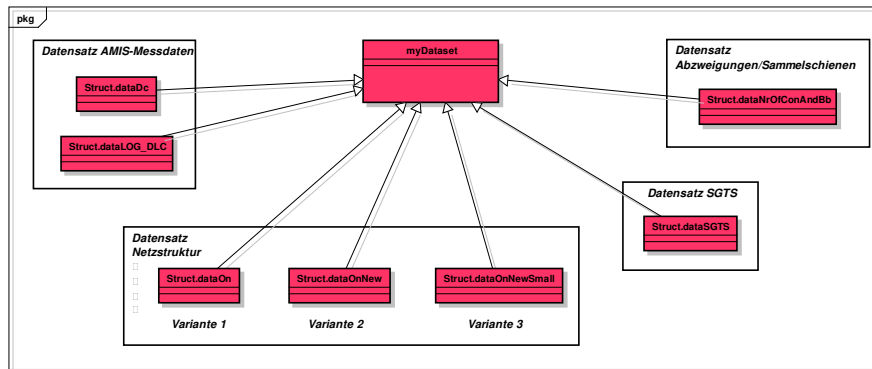


Abbildung 25: Schematisches Quasi-UML der selbst definierten Datenstrukturen

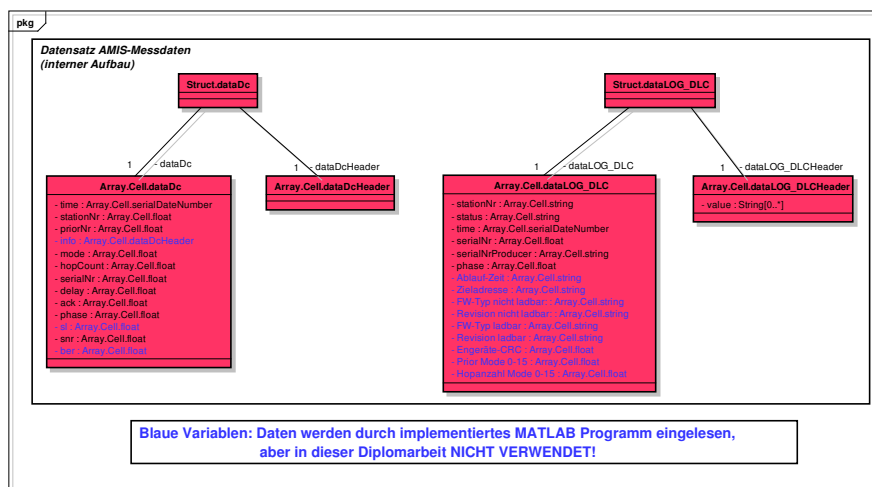


Abbildung 26: Quasi-UML Diagramm für den Datensatz AMIS-Messdaten

3.4.1.2 Repräsentation des Datensatzes Netzstruktur

Wie schon in Kapitel 3.2 (Seite 21) beschrieben liegt der Datensatz Netzstruktur in drei unterschiedlichen Varianten vor. Deshalb ist es auch notwendig für jede Variante eine eigene Datenstruktur zu definieren. In Abbildung 27 (Seite 38) ist der detaillierte Aufbau der Datenstruktur für jede der drei Varianten dargestellt.

3.4.1.3 Repräsentation des Datensatzes „Smart Grids Topologie Simulator“

Diese Datenstruktur wird dazu verwendet die Daten, welcher der Datensatz Smart Grids Topologie Simulator (SGTS) beinhaltet, in MATLAB zu speichern. Abbildung 28 (Seite 38) zeigt den detaillierten Aufbau der dazu definierten Datenstruktur.

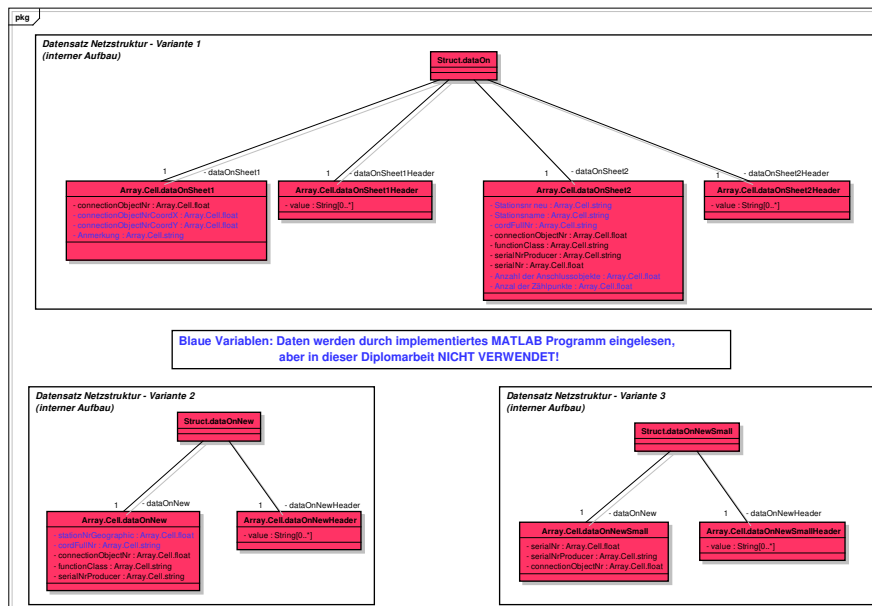


Abbildung 27: Quasi-UML Diagramm für den Datensatz Netzstruktur

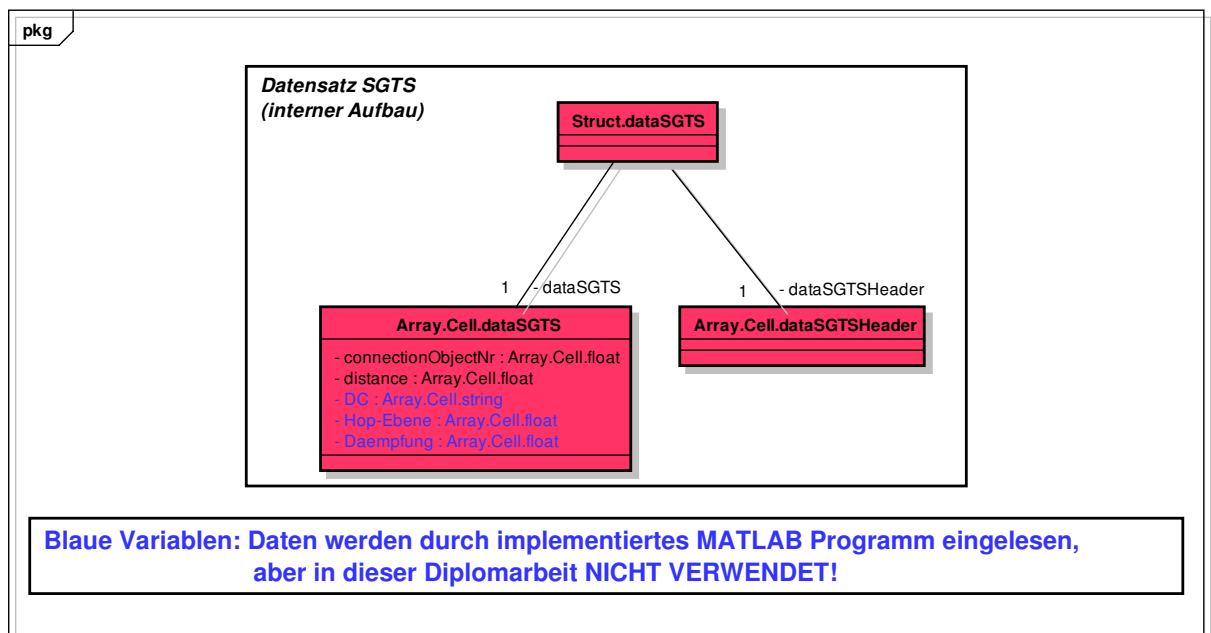


Abbildung 28: Quasi-UML Diagramm für den Datensatz SGTS

3.4.1.4 Repräsentation des Datensatzes Abzweigungen/Sammelschienen

Die Datenstruktur, welche für den Datensatz Abzweigungen/Sammelschienen definiert wurde, ist in Abbildung 29 ersichtlich. Diese zeigt in bereits gewohnter Weise den inneren Aufbau der dazu definierten Datenstruktur.

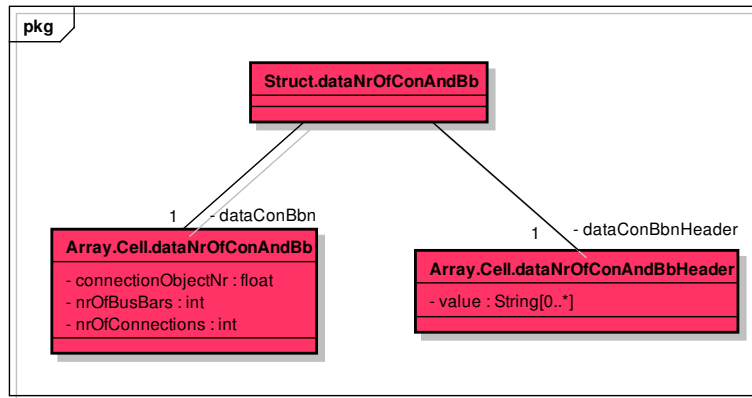


Abbildung 29: Quasi-UML Diagramm für den Datensatz Abzweigungen/Sammelschienen

3.4.1.5 Struktur des aggregierten Datensatzes Datenbasis für die Analyse

Wie bereits zu Beginn dieses Hauptkapitels beschrieben und in Abbildung 10 (Seite 18) schematisch dargestellt, müssen die einzelnen Datensätze zu einer gemeinsamen Datenbasis aggregiert werden, um für die Analyse verwendet werden zu können. Die zur Speicherung der Datenbasis für die Analyse definierte MATLAB Datenstruktur ist detailliert in Abbildung 30 (Seite 40) dargestellt. Wie in der eben genannten Abbildung ersichtlich ist der Aufbau analog zu den bisher beschriebenen Repräsentationen der ursprünglichen Datensätze (z.B. struct.dataDc, struct.dataOn, ...), jedoch sind alle für die Analyse benötigten Daten enthalten.

Im Prinzip könnte die Analyse mit der in Abbildung 30 (Seite 40) dargestellten Datenstruktur erfolgen, jedoch würde dieser einfache Aufbau bei jeder in MATLAB zu implementierenden Methode, welche damit arbeitet, einen erheblichen Programmieraufwand erfordern. Um eine umfangreiche und trotzdem effiziente Analyse der Datenbasis für die Analyse durchführen zu können ist es daher notwendig, die zur Speicherung der aggregierten Daten verwendete Datenstruktur intelligenter und komplexer aufzubauen. Damit ist gemeint, dass sie einerseits eine einfache Manipulation und Abfrage der gewünschten Daten unterstützen muss, andererseits damit aber auch eine durchaus komplexe Speicherung von gefilterten bzw. gruppierten Daten möglich sein soll. Die zu diesem Zweck definierte MATLAB Datenstruktur ist detailliert in Abbildung 31 (Seite 41) dargestellt.

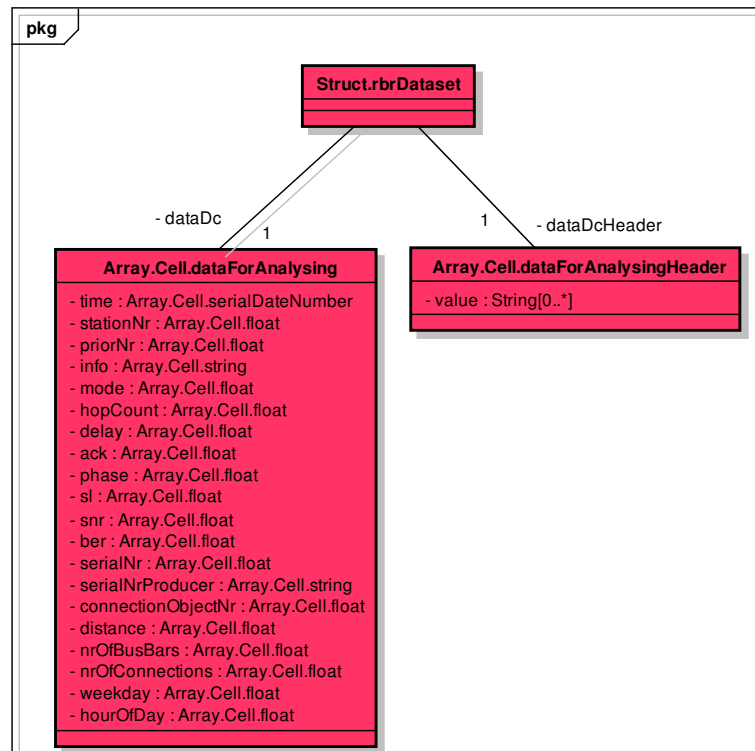


Abbildung 30: Quasi-UML Diagramm für die Datenbasis für die Analyse

Als Grundstruktur zur Speicherung der Daten ist im Teil links unten weiterhin die zuvor beschriebene Datenstruktur (struct.rbrDataset) vorhanden, jedoch wird diese erweitert. Um den zuvor beschriebenen Anforderungen gerecht zu werden wird als Grundaufbau der neuen, komplexeren Datenstruktur eine Baumstruktur gewählt, welche die Speicherung und das strukturierte Auslesen von gruppierten bzw. gefilterten Daten erlaubt.

Das Schema des dazu verwendeten Speicheraufbaus ist in Abbildung 32 (Seite 42) ersichtlich. Dabei sieht man, dass die eigentlichen Daten nur in den Blättern des Baumes gespeichert werden, zusammen mit eventuell vorhandenen Metadaten, welche eine optional erfolgte Filterung/Gruppierung der Ursprungsdaten (vom Typ struct.rbrDataset) speichern. Der Zugriff auf jedes einzelne Blatt des Baumes kann direkt durch Angabe des (mehrdimensionalen) Indexes erfolgen.

3.4.2 Erstellung der Datenbasis für die Analyse

In diesem Kapitel erfolgt eine genaue Beschreibung, wie aus den einzelnen Datensätzen Schritt für Schritt die Datenbasis für die Analyse heranwächst. Die prinzipielle Vorgehensweise wurde ja schon in Kapitel 3.3 (Seite 29) und die notwendigen, selbst definierten MATLAB Datenstrukturen im vorigen Kapitel 3.4.1 beschrieben. Abbildung 33 (Seite

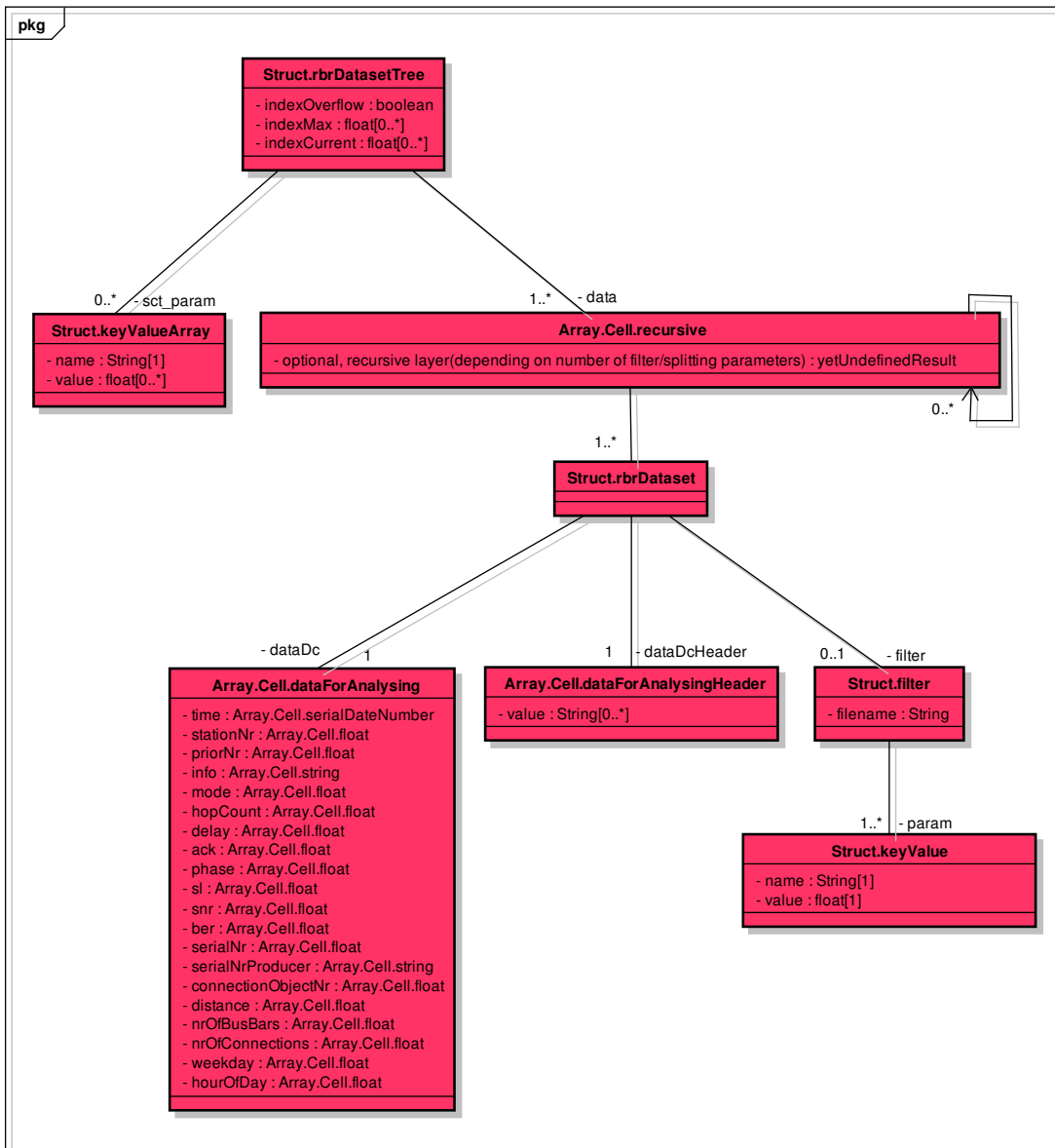


Abbildung 31: Quasi-UML Diagramm der selbst erstellten MATLAB Datenstruktur rbr-DatasetTree

43) zeigt den Ablauf, wie die Erstellung der Datenbasis für die Analyse aus den einzelnen Datensätzen erfolgt. Die eben genannten Abbildung ist praktisch eine detailliertere Version der Abbildung 22 (Seite 33), da die zuletzt genannte Abbildung lediglich die allgemeine Bezeichnung der Datensätze, jedoch nicht den Typ der MATLAB Datenstruktur enthält, in denen die Daten gespeichert sind.

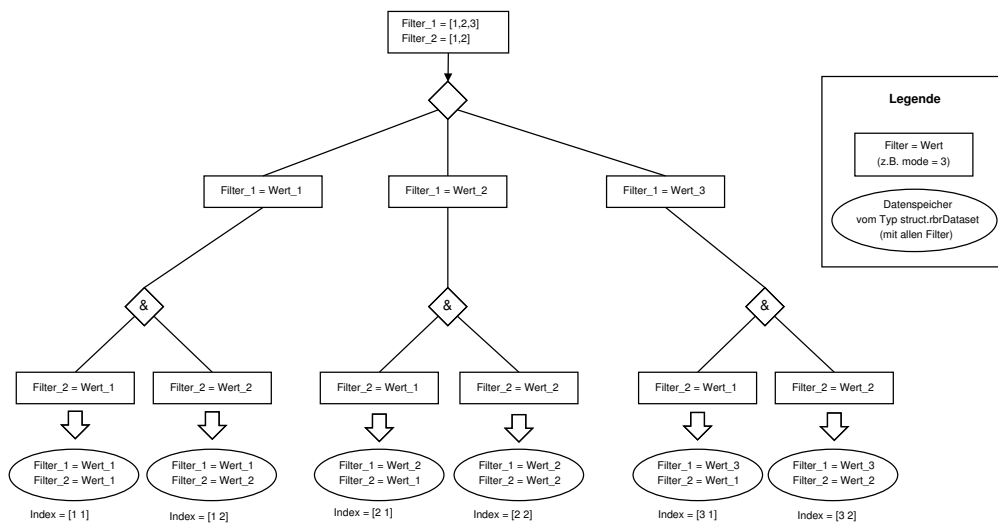


Abbildung 32: Beispiel eines Baumes der MATLAB Datenstruktur rbrDatasetTree

3.4.3 Struktur des MATLAB Programms

Der Aufbau des implementierten MATLAB Programmes ist in Abbildung 34 (Seite 44) dargestellt. Die Grundstruktur gliedert sich in drei Klassen:

- * MainCtrl
- * StatisticCtrl
- * PlotCtrl

Es folgt nun ein Überblick über die Aufgaben dieser drei Klassen und der darin enthaltenen Methoden und Klassenvariablen.

Klasse MainCtrl

In dieser Klasse befinden sich primär Methoden, welche zur Endbenutzung gedacht sind und welche Endergebnisse liefern. Als Beispiel sei die Methode `plotParamXVsParamY` angeführt, welche ein X-Y Diagramm (genauer eine MATLAB `figure`) von den übergebenen Parametern (`paramX` und `paramY`) aus den ebenfalls übergebenen Daten (`dataIn`) erzeugt (und je nach Konfiguration auch abspeichert).

Klasse StatisticCtrl

In dieser Klasse befinden sich primär Methoden, welche Berechnungen, Formatierungen oder andere Anpassungen an Daten vornehmen. Diese Methoden liefern jedoch keine direkten Endergebnisse, wie z.B. Grafiken, sondern werden von den in der Klasse `MainCtrl` implementierten Methoden aufgerufen, um die von diesen benötigten Datenmanipulationen durchzuführen. Als Beispiel sei hier die Methode `prepareHistogramDataset` angeführt, welche aus den gegebenen Eingangsdaten (Parameter `dataIn`) eine Datenstruk-

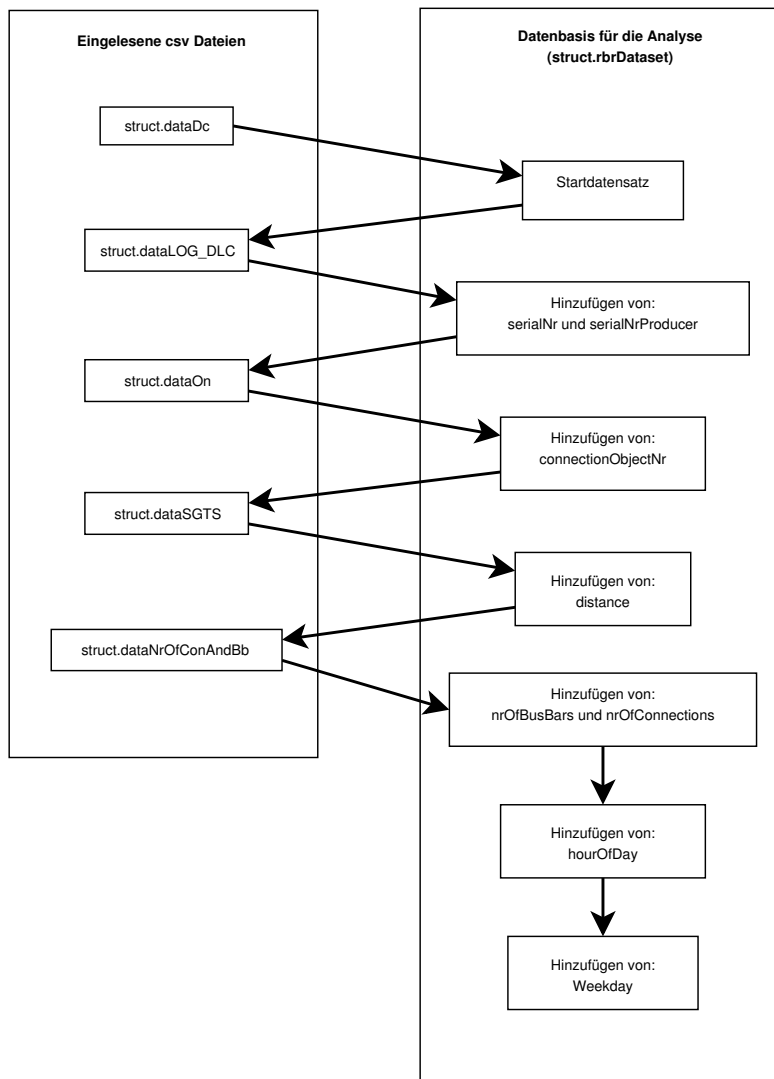


Abbildung 33: Schematischer Ablauf der Erstellung der Datenbasis für die Analyse

tur für den gewünschten Bereich (mit Parameter `dataRange` definiert) erzeugt, welche mit MATLAB direkt als Histogramm dargestellt werden kann. Das Erzeugen dieser Darstellung (des MATLAB Histogrammes) erfolgt jedoch in der zugehörigen Methode `plotHistogramOfParam` der Klasse `MainCtrl`.

Klasse `PlotCtrl`

In dieser Klasse befinden sich primär solche Methoden, welche für die Erzeugung, für das Layout und für die Formatierung der verschiedenen MATLAB Diagramme zuständig sind. Als Beispiel sei hier die Methode `plotHistogram` erwähnt, welche ein MATLAB

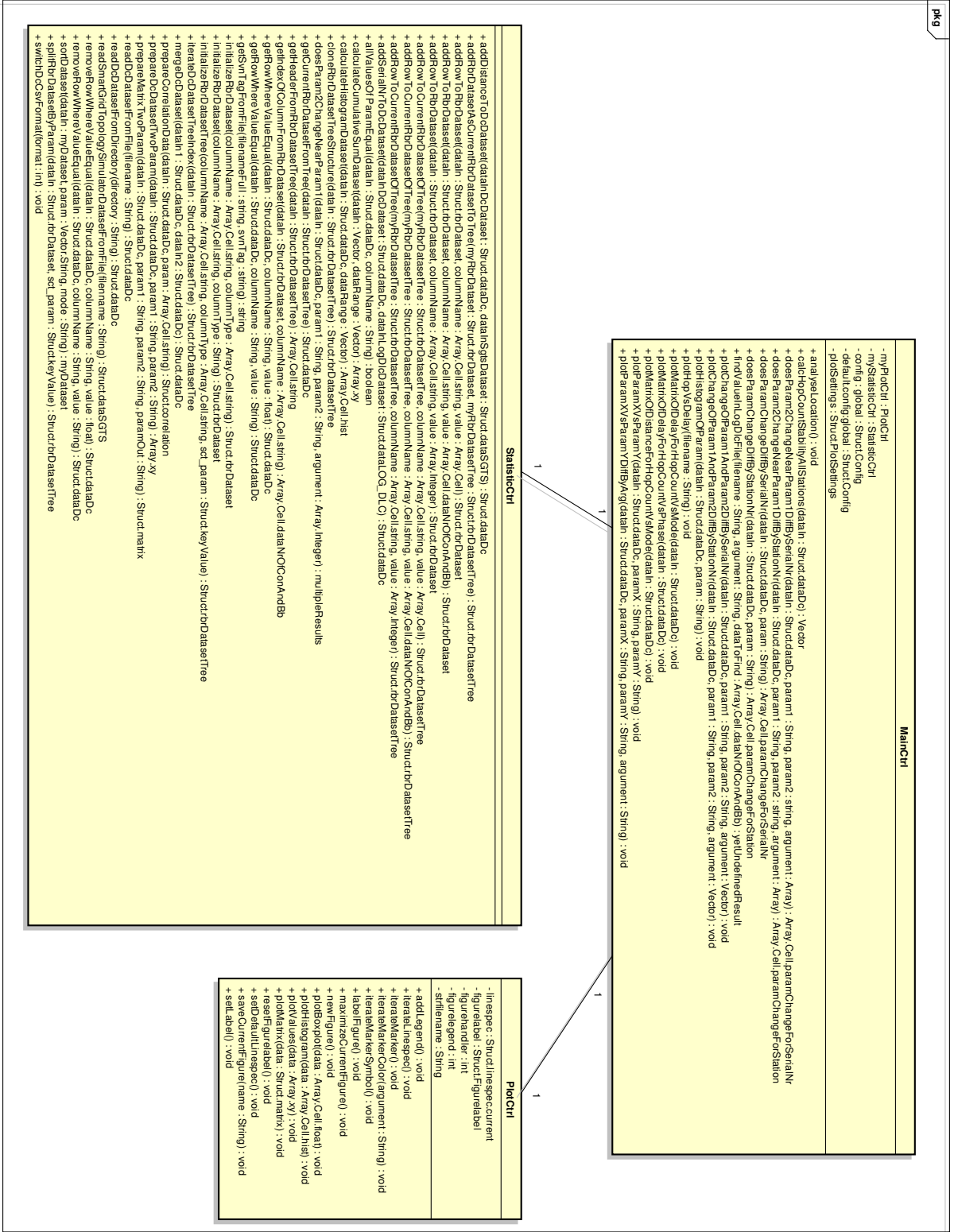


Abbildung 34: Quasi-UML Klassendiagramm des entwickelten MATLAB Programmes

Histogramm für die übergebenen Daten (Parameter `data`) erzeugt. Die übergebene Datenstruktur wurde vorher durch die Methode `prepareHistogramDataset` der Klasse `StatisticCtrl` erzeugt (als Beispiel im vorigen Absatz beschrieben). Weiters enthält diese Klasse auch Methoden, mit denen die Klassenvariablen, welche verschiedene Konfigurationsparameter (z.B. für das Layout oder die Beschriftung der Diagramme) enthalten, verändert werden können. Als Beispiel dafür sei die Methode `iterateLinespec` erwähnt, welche den Linientyp über alle in MATLAB möglichen Kombinationen der einzelnen Varianten (Farben, durchgezogen/strichliert, ...) iteriert, um mehrere Parameter in dem gleichen Diagramm darstellen zu können.

3.4.3.1 Erstellung des Datensatzes AMIS-Messdaten

Wie in Kapitel 3.1.2 (Seite 19) beschrieben besteht die Datenquelle des Datensatzes AMIS-Messdaten aus mehreren csv-Dateien. Um den Datensatz AMIS-Messdaten zu erhalten müssen diese csv-Dateien in MATLAB eingelesen und die Daten anschließend richtig zusammengeführt werden. Den grundsätzlichen Ablauf dazu beschreibt das Sequenzdiagramm in Abbildung 35 (Seite 46). Dass die `stationNr` der AMIS Endgeräte nicht statisch ist, sondern vom DC dynamisch vergeben wird (siehe Kapitel 3.1.2, Seite 19), muss für die Zuweisung der `serialNr` und der `serialNrProducer` berücksichtigt werden. Der dazu verwendete Algorithmus ist, beispielhaft anhand der implementierten MATLAB Methode `addSerialNrToDcDataset`, als Sequenzdiagramm in der Abbildung 36 (Seite 47) dargestellt. Der Algorithmus für das Hinzufügen der `serialNrProducer` ist vom Prinzip der gleiche, welcher auch für das Hinzufügen der `serialNr` verwendet wird. Die Implementierung unterscheidet sich jedoch leicht aufgrund des anderen Datentyps (String bei `serialNrProducer` im Gegensatz zu Numeric bei `serialNr`).

3.4.3.2 Hinzufügen des Datensatzes Netzstruktur

Die Erstellung des Datensatzes Netzstruktur ist wesentlich einfacher als die eben beschriebene Erstellung des Datensatzes AMIS-Messdaten, da keine dynamische Änderung etwaiger Matching Parameter vorkommt. Jedoch ist der Algorithmus zur Erstellung je nach vorliegender Variante der Datenquelle (\rightarrow csv-Datei), aufgrund des unterschiedlichen Aufbaus und Informationsgehalts derselben, nicht derselbe.

Erstellen des Datensatzes Netzstruktur - Variante 1

Bei dieser Variante müssen aufgrund des Vorliegens von zwei csv-Dateien die Daten nach dem Einlesen zusammengeführt werden. Die dazu implementierte MATLAB Methode hat den Namen `readOnDatasetFromFile`. Das Matching erfolgt über den in beiden csv-Dateien vorhandenen Parameter `connectionObjectNr`. Dieser muss aufgrund des Aufbaus der csv-Datei (siehe Abbildung 14 auf Seite 24) jedoch vor dem Matching auf alle Zeilen, für welche er gilt, erweitert werden. Weiters werden alle Zeilen, welche als `functionClass` nicht den String AMIS enthalten, gelöscht.

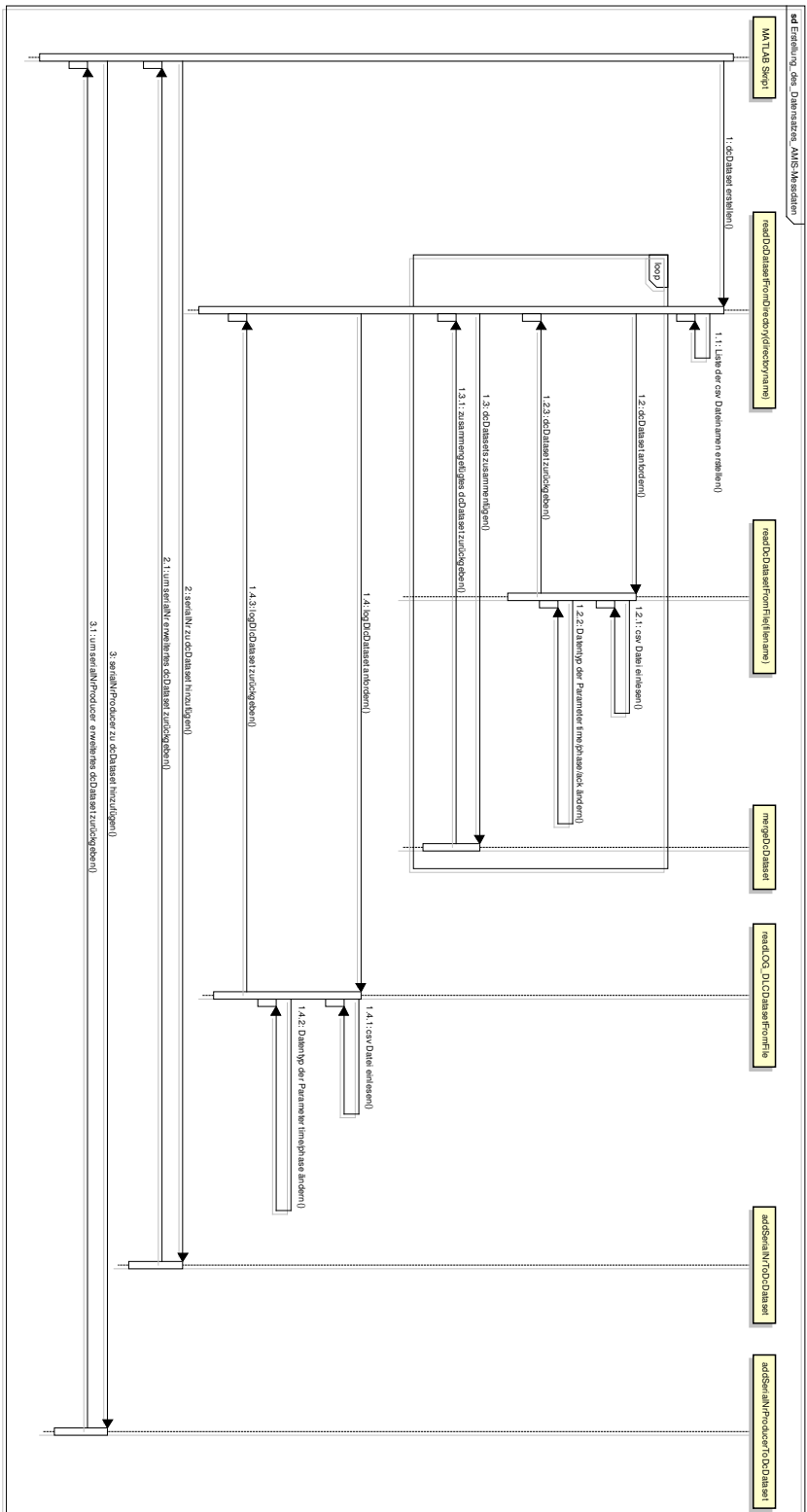


Abbildung 35: Vereinfachtes Sequenzdiagramm zur Erstellung des Datensatzes AMIS-Messdaten

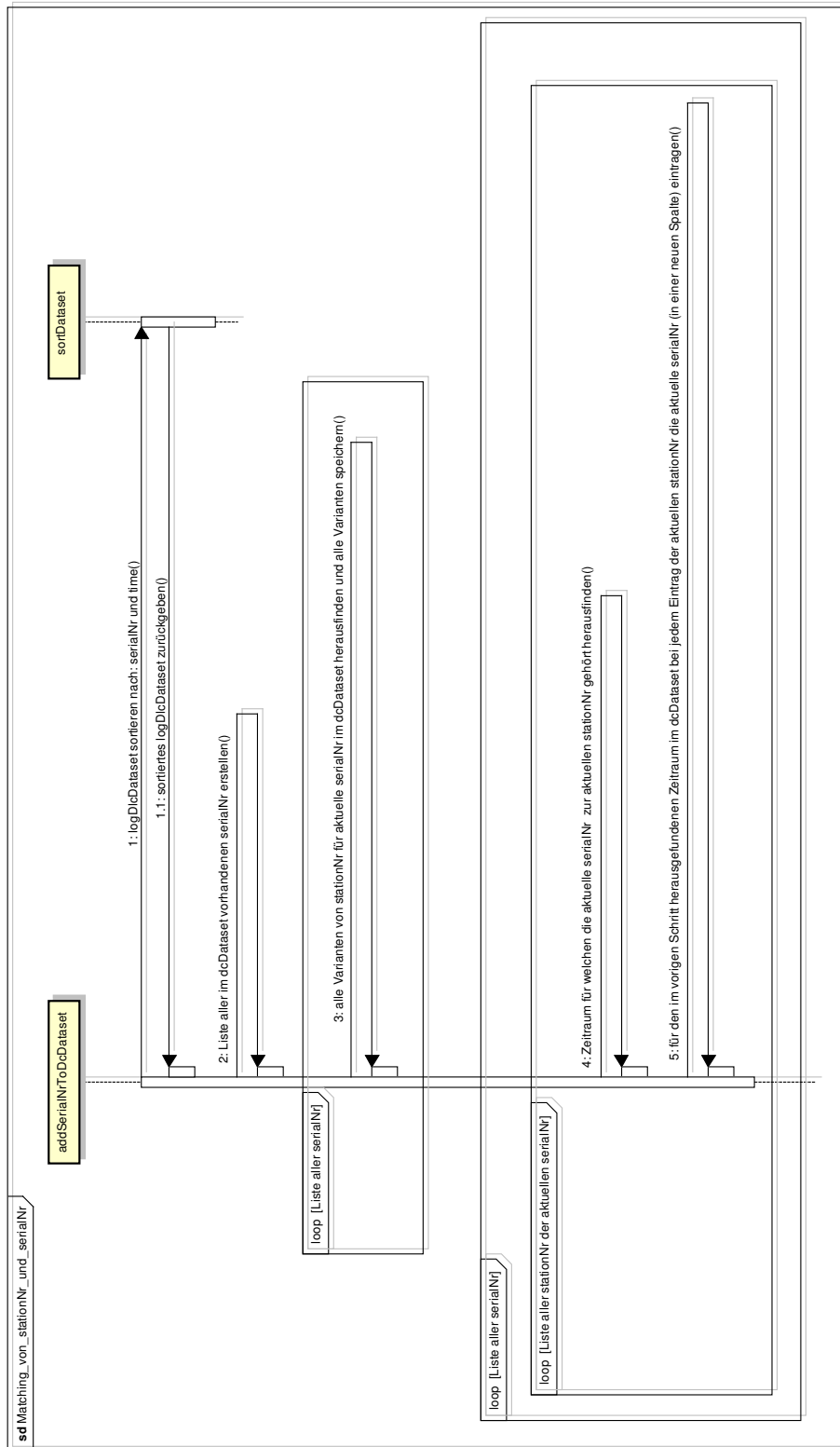


Abbildung 36: Vereinfachtes Sequenzdiagramm für das Matching von stationNr und serialNr

Erstellen des Datensatzes Netzstruktur - Variante 2

Bei dieser Variante liegt lediglich eine csv-Datei vor, weshalb sich das Erstellen einfacher gestaltet als bei Variante 1. Die dazu implementierte MATLAB Methode hat den Namen `readOnDatasetNewFromFile`. Nach dem Einlesen der csv-Datei sind jedoch aufgrund des Aufbaus dieser (siehe Abbildung 13 auf Seite 23), analog wie in Variante 1, die Einträge für den Parameter `connectionObjectNr` zu erweitern.

Erstellen des Datensatzes Netzstruktur - Variante 3

Bei dieser Variante ist lediglich das Einlesen der csv-Datei erforderlich. Dazu wird die gleiche Methode wie für Variante 2 (`readOnDatasetNewFromFile`) verwendet.

Nach dem Erstellen des Datensatzes Netzstruktur in MATLAB liegt je nach Variante eine andere MATLAB Datenstruktur vor (siehe Abbildung 27 auf Seite 38). Der zum Hinzufügen des Parameters `connectionObjectNr` implementierte MATLAB Methode `addConnectionObjectNrToDcDataset` kann jede der drei definierten MATLAB Datenstrukturen des Datensatzes Netzstruktur übergeben werden. Intern wird abhängig vom übergebenen Datentyp des Datensatzes Netzstruktur ein anderer Algorithmus ausgeführt.

3.4.3.3 Hinzufügen des Datensatzes “Smart Grids Topologie Simulator”

Die Erstellung des Datensatzes Smart Grids Topologie Simulator (SGTS) ist im Gegensatz zu den beiden bisher besprochenen relativ einfach. Wie man in Abbildung 18 (Seite 27) erkennen kann ist es jedoch nach dem Einlesen der Datenquelle (\rightarrow csv-Datei) notwendig, jene Einträge in der Spalte Knotenname zu identifizieren, welche eine gültige `connectionObjectNr` enthalten. Das ist dann der Fall, wenn der Eintrag eine acht-stellige Nummer ist, Zeilen mit anderen Einträgen werden aus dem Datensatz gelöscht. Anschließend werden alle Einträge für die `connectionObjectNr` vom Datentyp String zum Datentyp Numeric konvertiert und die Einträge für `distance` in Meter umgerechnet (durch Multiplikation mit dem Faktor 1000). Die implementierte MATLAB Methode, welche das Einlesen und die Anpassung der Daten durchführt, hat den Namen `readSmartGridTopologySimulatorDatasetFromFile`.

Das Hinzufügen der `distance` zur Datenbasis für die Analyse erfolgt durch die Methode `addDistanceToDcDataset`. Dabei wird das Mapping zu den bereits vorhandenen Daten über die `connectionObjectNr` hergestellt (siehe Abbildung 10 auf Seite 18). Da das zur Erstellung der als Datenquelle verwendeten csv-Datei (siehe Kapitel 3.2.4 auf Seite 26) durch ein Programm erfolgt, welches sich in der vorliegenden Version noch in der Entwicklungsphase befindet und daher teilweise noch fehlerhafte Ausgaben produziert, werden vor und während des Mappings verschiedene Fehlerabfragen und Plausibilitätschecks durchgeführt, um die Gültigkeit der Daten sicherzustellen. So wird z.B. erkannt wenn für eine bestimmte `connectionObjectNr` mehrere Einträge für die `distance` vorhanden sind

(wie es dazu kommen kann siehe Kapitel 3.2.4 auf Seite 26). Weiters wird überprüft, ob für die in der Datenbasis für die Analyse vorliegenden Einträge für die `connectionObjectNr` überhaupt zugehörige `distance` Einträge gefunden werden können. Es wird eine Warnung ausgegeben falls keine Übereinstimmungen vorliegen, was z.B. der Fall ist, wenn die vorliegenden Datenquellen (\rightarrow csv-Dateien) nicht aus dem gleichen Netz stammen.

3.4.3.4 Hinzufügen des Datensatzes Abzweigungen/Sammelschienen

Da wie in Kapitel 3.2.5 (Seite 27) schon beschrieben diese Datenquelle selbst erstellt wurde ist der Aufbau der csv-Datei natürlich derart, dass eine einfache Verwendung gewährleistet ist. Die Erstellung des Datensatzes Abzweigungen/Sammelschienen beschränkt sich daher darauf, die Datenquellen (\rightarrow csv-Datei) in MATLAB einzulesen. Weitere Anpassungen sind nicht notwendig. Erledigt wird das Einlesen und Abspeichern im `dataset` vom Typ `dataNrOfConAndBb` (siehe Abbildung 29 auf Seite 39) durch die Methode `readConnectionObjectNrBusbarNrDatasetFromFile`.

Das Hinzufügen des Parameters `nrOfBusBars` zur Datenbasis für die Analyse erfolgt durch die Methode `addNrOfBusBarsToDcDataset`, das des Parameters `nrOfConnections` durch die Methode `addNrOfBusBarsToDcDataset`. Das Matching beider Parameter mit den bereits vorhandenen Daten in der Datenbasis für die Analyse erfolgt über den Parameter `connectionObjectNr`.

3.4.3.5 Fertigstellen der Datenbasis für die Analyse

Schlussendlich werden noch die Parameter `weekday` und `hourOfDay` zur Datenbasis für die Analyse hinzugefügt (siehe Abbildung 10 auf Seite 18). Die Berechnung der Werte dieser beiden Parameter erfolgt direkt aus dem Parameter `time` in den Methoden `addWeekdayToDcDataset` und `addHourOfDayToDcDataset`. Die Funktionsweise dieser beiden Methoden ist sehr ähnlich. Zuerst wird bei beiden eine neue Spalte zur Datenbasis für die Analyse hinzugefügt. Anschließend wird mittel MATLAB Standardfunktionen (`datevec` für den Parameter `hourOfDay` und `weekday` für den Parameter `weekday`) aus dem Parameter `time` der entsprechende Parameter berechnet.

Mit dem Hinzufügen dieser beiden Parameter ist die Erstellung der Datenbasis für die Analyse abgeschlossen. Im nächsten Kapitel wird nun beschrieben, welche Analysen damit durchgeführt werden und wie die dazu verwendeten MATLAB Methoden aufgebaut und implementiert sind.

4 Analyse der erstellten Datenbasis

Mit dem Wissen über den Aufbau der Datenstrukturen kann nun eine Beschreibung der für die Analyse der erstellten Datenbasis verwendeten Methoden und der dazu implementierten MATLAB-Methoden erfolgen. Zuerst werden dazu im Kapitel 4.1 die für eine erste Analyse verwendeten, grafischen Methoden und Diagramme vorgestellt. Es wird dabei auch der Aufbau des dazu implementierten MATLAB-Programms schematisch beschrieben. Bevor mit der eigentlichen Analyse begonnen werden kann, muss die Datenbasis dafür noch einen Preprocessing Prozess durchlaufen, in welcher z.B. nicht benötigte Einträge gelöscht oder diverse Filterungen und Gruppierungen der Daten durchgeführt werden. Die genaue Beschreibung dazu erfolgt in Kapitel 4.1.2.2 (Seite 60).

Auf die in weiterer Folge durchgeführte, erweiterte Analyse der Datenbasis mittels eines heuristischen Ansatzes wird in Kapitel 4.2 (Seite 67) näher eingegangen. Es erfolgt zuerst eine kurze allgemeine Beschreibung, wie eine Heuristik in MATLAB zu verwenden ist, um danach die vorgenommene Implementierung und die Ergebnisse verstehen zu können. Abschließend erfolgt eine Erläuterung, wie der am ICT der TU-Wien entwickelte AMIS-Kommunikationssimulator durch Entwicklungen, welche im Zuge dieser Diplomarbeit gemacht wurden, verbessert werden konnte.

4.1 Grafische Analyse der Datenbasis

Um eine bessere Vorstellung vom Inhalt der Datenbasis zu bekommen wurden zuerst einige grafische Darstellungen der Datenbasis angefertigt. Die dazu verwendeten Diagramme sind relativ einfache, mit MATLAB Standardfunktionen berechenbare, Standarddiagramme, wie z.B. Histogramme, XY-Diagramme und Tabellen. Der eigentliche Aufwand bei den grafischen Darstellungen liegt darin, welche Daten angezeigt werden sollen und wie diese berechnet bzw. strukturiert werden. Um die verwendbaren Input Daten(kombinationen) der Diagramme möglichst flexibel zu halten wurde einerseits die in Kapitel 3.4.1.5 (Seite 39) bereits beschriebene Datenstruktur `rbrDataSetTree` definiert und andererseits wurden die in der Klasse `PlotCtrl` implementierten MATLAB-Methoden zur Erzeugung der Diagramme möglichst allgemein gehalten, sodass eine endgültige Festlegung der darzustellenden Parameter erst durch die aufrufende MATLAB-Methode vorgenommen wird. Für eine genauere Beschreibung der Struktur des implementierten MATLAB-Programms mit Beispielen sei auf die bereits erfolgte Beschreibung in Kapitel 3.4.3 (Seite 42) verwiesen. Es folgt nun im nächsten Kapitel eine Beschreibung der zur Analyse verwendeten Diagrammtypen.

4.1.1 Verwendete Diagramme

In diesem Kapitel werden die zur Analyse verwendeten Diagramme beschrieben. Im Prinzip handelt es sich um in MATLAB verfügbare Standard-Diagramme, welche jedoch um benötigte Funktionen erweitert wurden. Es wird auch die benutzte Beschriftung explizit erklärt und auf wichtige Details hingewiesen, welche übersehen werden könnten.

4.1.1.1 Histogramm

In Abbildung 37 ist als Beispiel ein Histogramm abgebildet, welches mit der Methode `MainCtrl.plotHistogramOfParam` erstellt wurde.

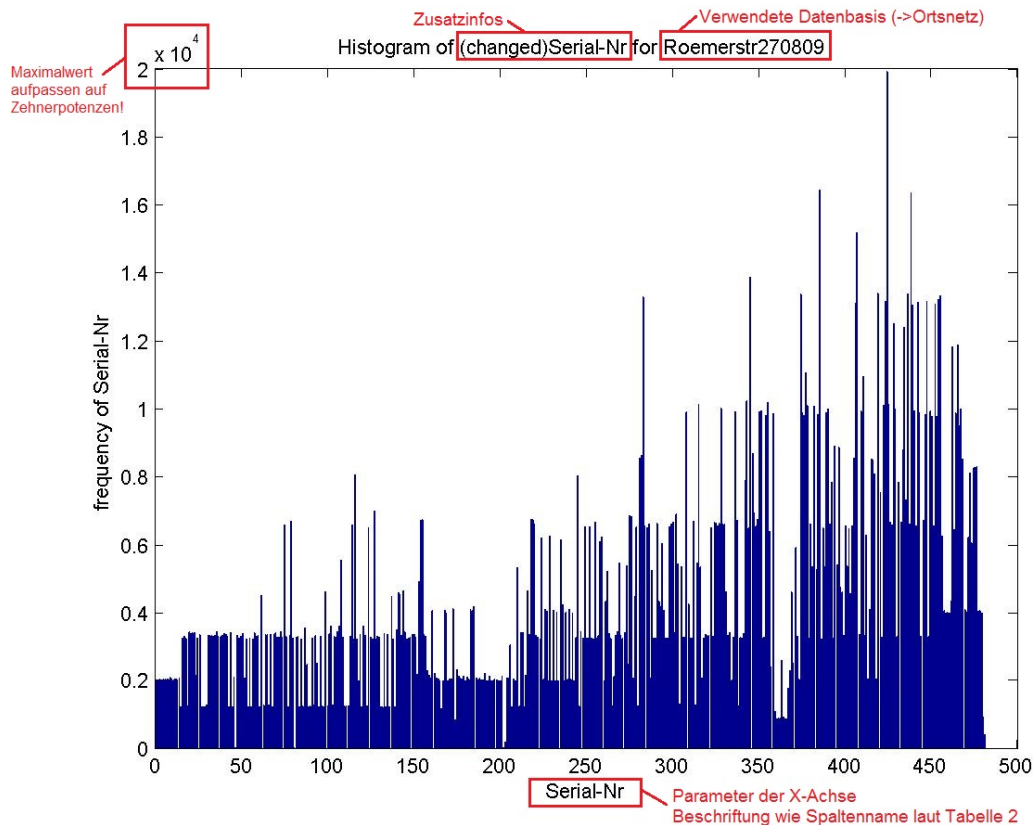


Abbildung 37: Beispiel eines mit MATLAB erstellten Histogramms

Die genannte Methode ruft die Methode `StatisticCtrl.prepareHistogramDataset` auf, welche aus den übergebenen Daten (`dataIn`) und dem gewünschten Bereich (`dataRange`) eine Datenstruktur vom Typ `Array.Cell.hist` erstellt, welche angepasste Daten enthält und deren Aufbau in Abbildung 38 (Seite 52) abgebildet ist. Diese Anpassung ist deshalb notwendig, damit das mit der Methode `PlotCtrl.plotHistogram` erstellte Histogramm eine sinnvolle Skalierung erhält. Abschließend erfolgt durch die Methode `MainCtrl.plotHistogramOfParam` noch die Beschriftung und eventuell eine Speicherung des erstellten Histogrammes.

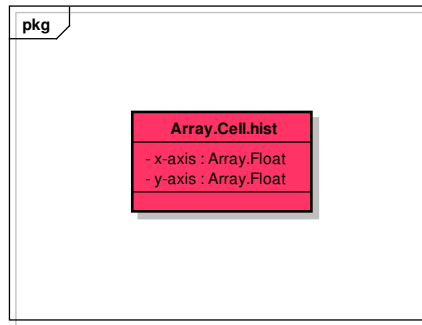


Abbildung 38: Quasi-UML Diagramm der selbst erstellten MATLAB-Datenstruktur für das Histogramm

4.1.1.2 XY-Diagramm

In Abbildung 39 ist als Beispiel ein relativ aufwändiges XY-Diagramm abgebildet, welches mit der Methode `MainCtrl.plotChangeOfParam1AndParam2DiffBySerialNr` erstellt wurde.

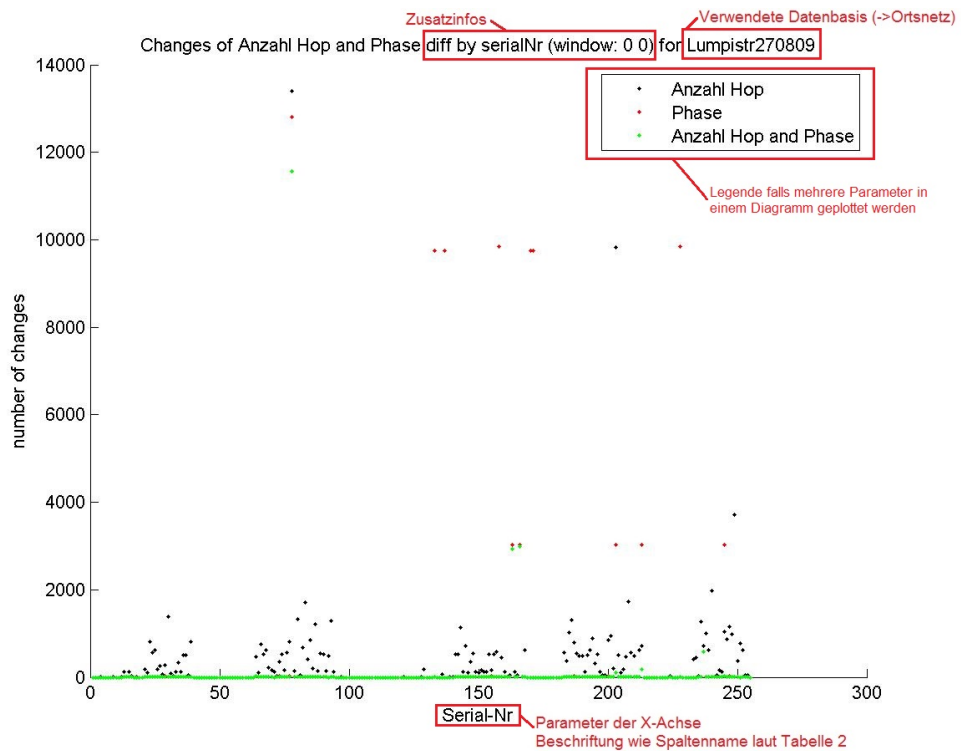


Abbildung 39: Beispiel eines mit MATLAB erstellten XY-Diagramms

Von dieser Methode wird zuerst eine Anpassung der Datenbasis derart vorgenommen, dass nach dem gesamten Anpassungsalgorithmus eine Datenstruktur vom Typ `Array.xy`, deren Aufbau in Abbildung 40 (Seite 53) dargestellt ist, mitsamt den gewünschten Daten herauskommt, welche anschließend zur Erstellung des eigentlichen Diagramms an die Methode `PlotCtrl.plotValues` übergeben wird. Zuletzt erfolgt in der Methode `MainCtrl.plotChangeOfParam1AndParam2DiffBySerialNr` unter Zuhilfenahme von weiteren Methoden der Klasse `PlotCtrl` noch die Beschriftung, Formatierung und eventuell eine Speicherung des erstellten XY-Diagramms.

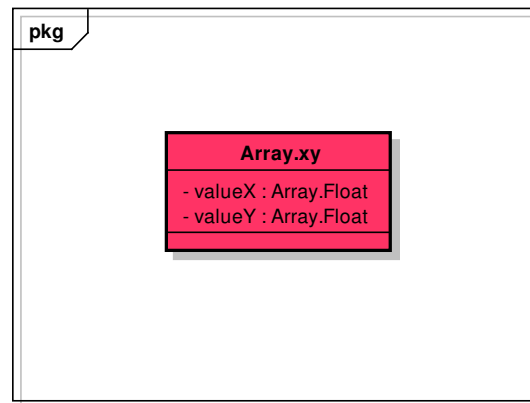


Abbildung 40: Quasi-UML Diagramm der selbst erstellten MATLAB-Datenstruktur für das XY-Diagramm

Dieser soeben beschriebene Ablauf zur Erstellung eines XY-Diagramms wird, bis auf einen etwaigen anderen Algorithmus zur Anpassung der Datenbasis, auch noch von folgenden Methoden verwendet:

- * `MainCtrl.plotChangeOfParam1AndParam2DiffByStationNr`
- * `MainCtrl.plotChangeOfParam1AndParam2DiffBySerialNr`
- * `MainCtrl.plotParamXVsParamY`
- * `MainCtrl.plotParamXVsParamYDiffByArg`

4.1.1.3 XY-Diagramm mit automatisch hinzugefügtem Jitter

Um zusätzlich zu den tatsächlichen Werten auch Informationen über die Häufigkeit des Auftretens jedes einzelnen Wertes im gleichen Diagramm darstellen zu können, kann das implementierte XY-Diagramm mit einem für die Anzeige zu jedem Wert automatisch und zufällig hinzugefügtem Jitter verwendet werden. In Abbildung 41 (Seite 54) ist ein Beispiel eines solchen Diagramms abgebildet, Abbildung 42 (Seite 54) zeigt ein mit den selben Werten erstelltes XY-Diagramm (Beschreibung siehe Kapitel 4.1.1.2 auf Seite 51).

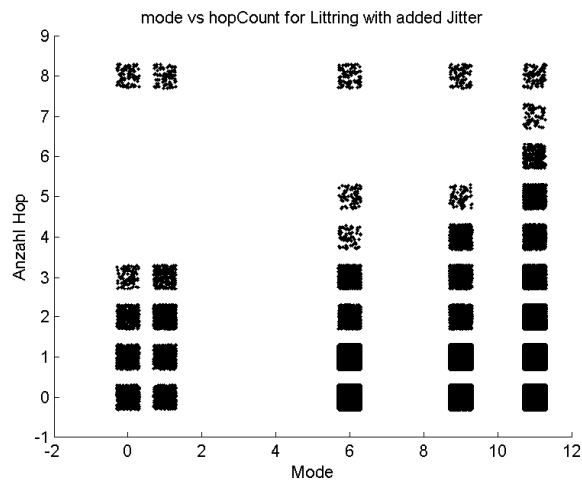


Abbildung 41: Beispiel eines XY-Diagramms MIT automatisch und zufällig hinzugefügtem Jitter (nur für die Anzeige)

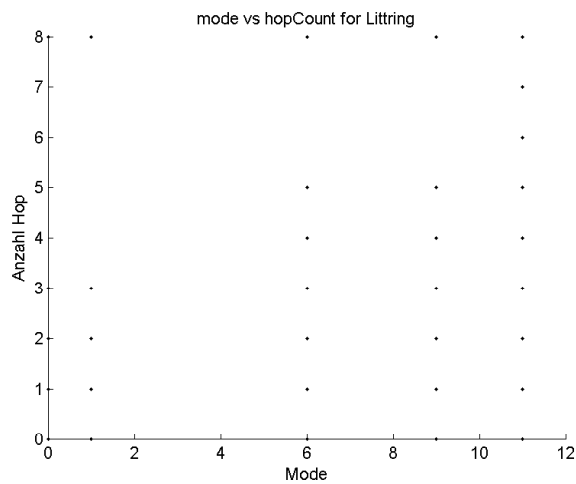


Abbildung 42: Beispiel eines XY-Diagramms OHNE automatisch hinzugefügtem Jitter

4.1.1.4 Korrelationsmatrix-Diagramm

In Abbildung 43 ist ein Beispiel eines Korrelationsmatrix-Diagrammes mit einer erklärenden Beschriftung abgebildet. Solche Diagramme können mit der MATLAB Stan-

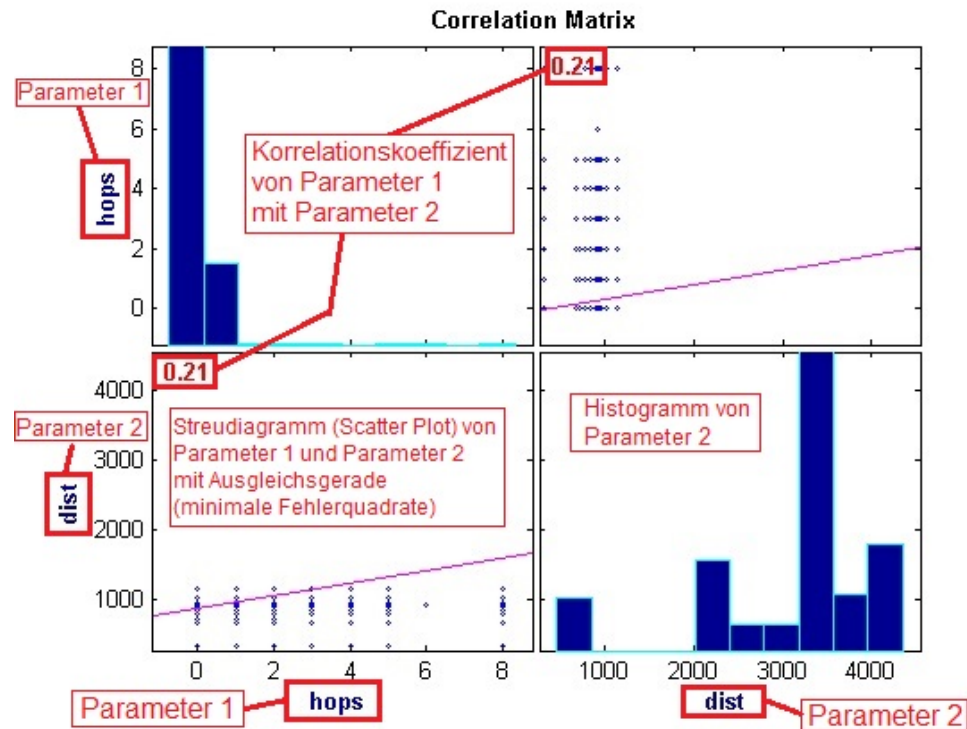


Abbildung 43: Beispiel eines mit MATLAB erstellten Korrelationsmatrix-Diagrammes

dardfunktion `corrplot`¹ erzeugt werden. Die selbst implementierte MATLAB-Methode `MainCtrl.plotCorrelationOfParam` erstellt ein Korrelationsmatrix-Diagramm für die übergebenen Daten (`dataIn` vom Typ `struct.dataDc`) mit den gewünschten Parametern (`param` vom Typ `Array.String`). Die Methode unterstützt auch das Erstellen von Korrelationsmatrix-Diagrammen mit mehr als zwei Parametern. Der prinzipielle Ablauf ist analog zu denen der Erstellung von Histogrammen bzw. XY-Diagrammen. Zuerst werden in der Methode `MainCtrl.plotCorrelationOfParam` die Daten durch Verwendung der Methode `StatisticCtrl.prepareCorrelationData` angepasst und in einer Datenstruktur vom Typ `Array.Float` gespeichert. Anschließend wird aus diesem Datensatz durch die Methode `PlotCtrl.plotCorrelation`, welche die bereits genannte MATLAB-Standardmethode `corrplot` verwendet, das gewünschte Korrelationsmatrix-Diagramm erzeugt. Zuletzt wird noch durch die Methode `MainCtrl.plotCorrelationOfParam` die Beschriftung, Formatierung und eventuell eine Speicherung des erstellten Diagrammes durchgeführt.

¹<http://www.mathworks.de/de/help/econ/corrplot.html>

4.1.1.5 AMIS-Parameter Matrix

In Abbildung 44 (Seite 56) ist ein Beispiel einer AMIS-Parameter Matrix mit erklärenden Beschriftungen abgebildet. In der eben genannten Abbildung ist als Beispiel ein AMIS-Parameter Matrix für den Mittelwert der Verzögerungszeit für alle Kombinationen von mode und hopCount abgebildet. Die Erstellung solch einer numerischen Matrix er-

	0	1	2	3	4	5	6	7	8
0	NaN	168	343.8214	457.6403	570.3482	NaN	NaN	NaN	1.0212e+03
1	NaN	126	244.9659	324.8832	404.9936	NaN	NaN	NaN	723.4907
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
6	44.5363	72.0638	159.8153	251.8295	319.1429	389.0915	453	NaN	580.6412
7	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9	35.9965	63.5175	88.1624	170.1464	225.2481	288.9398	336.2500	383	430.4125
10	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
11	27.0394	49.1564	72.6955	96.2838	142.9465	211.4388	255.5808	304.1343	315.8772
12	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
13	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
14	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
15	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Parameter 2 hopCount

Wert für: hopCount = 8 mode = 6

NaN (Not a Number) -> kein Wert vorhanden

Parameter 1 - mode

Abbildung 44: Beispiel einer mit MATLAB erstellten numerischen Matrix

folgt durch die Methode `MainCtrl.plotMatrixOfDelayForHopCountVsMode`. Die eigentliche Berechnung der Werte für die AMIS-Parameter Matrix erfolgt in der Methode `StatisticCtrl.prepareMatrixTwoParam`. Diese Methode berechnet für alle Kombinationen der zwei Parameter (`param1` und `param2`)² Werte für den gewünschten Ausgangsparameter (`paramOut`) und speichert diese in einer Datenstruktur vom Typ `Struct.matrix`, deren Aufbau in Abbildung 45 (Seite 57) abgebildet ist, ab. Die berechneten und gespeicherten Werte für den gewünschten Ausgabeparameter (für alle Kombinationen) sind:

- * Minimalwert (min)
- * Maximalwert (max)
- * Mittelwert (avg)
- * Median (med)
- * Verluste (ploss)
- * Anzahl (number)

4.1.2 Implementierung des MATLAB-Programms (Analyseteil)

In diesem Kapitel erfolgt eine grundlegende Beschreibung des im Zuge dieser Diplomarbeit implementierten MATLAB-Programms, welches für die erste Analyse verwendet wird. Es erfolgt hier keine detaillierte Beschreibung des Quellcodes, vielmehr soll ein

²derzeit nur hopCount vs. mode und hopCount vs. phase implementiert

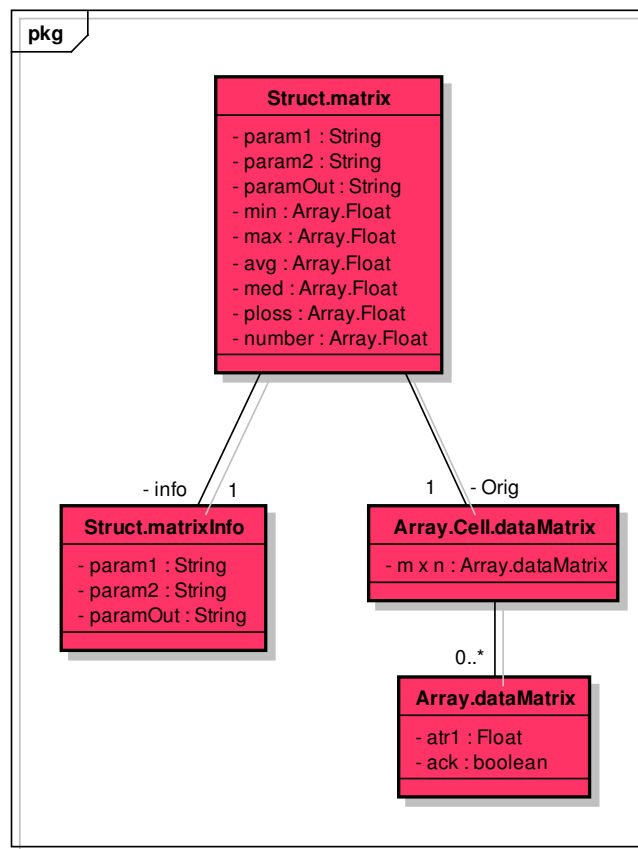


Abbildung 45: Quasi-UML Diagramm der MATLAB-Datenstruktur für die AMIS-Parameter Matrix

grundlegendes Verständnis des Programmablaufes vermittelt werden um die Methodik der Analyse und die Ergebnisse besser verstehen zu können.

4.1.2.1 Konfiguration der Analyse

Abbildung 46 (Seite 58) zeigt den Ablauf, nach welchem die Initialisierungen und Definitionen der für die Analyse benötigten Parameter ablaufen.

Dieser Ablauf lässt sich in zwei Schritte unterteilen:

1. Initialisierungen und Definitionen allgemeiner Programmparameter (durch das MATLAB Skript `configini.m`)
2. Initialisierungen und Definitionen der auszuführenden Analyse (direkt im MATLAB Skript `analyseLocations.m`)

Als erster Schritt zur Konfiguration wird vom MATLAB Skript `analyseLocations.m` das MATLAB Skript `configini.m` aufgerufen (siehe Abbildung 46 auf Seite 58). Das

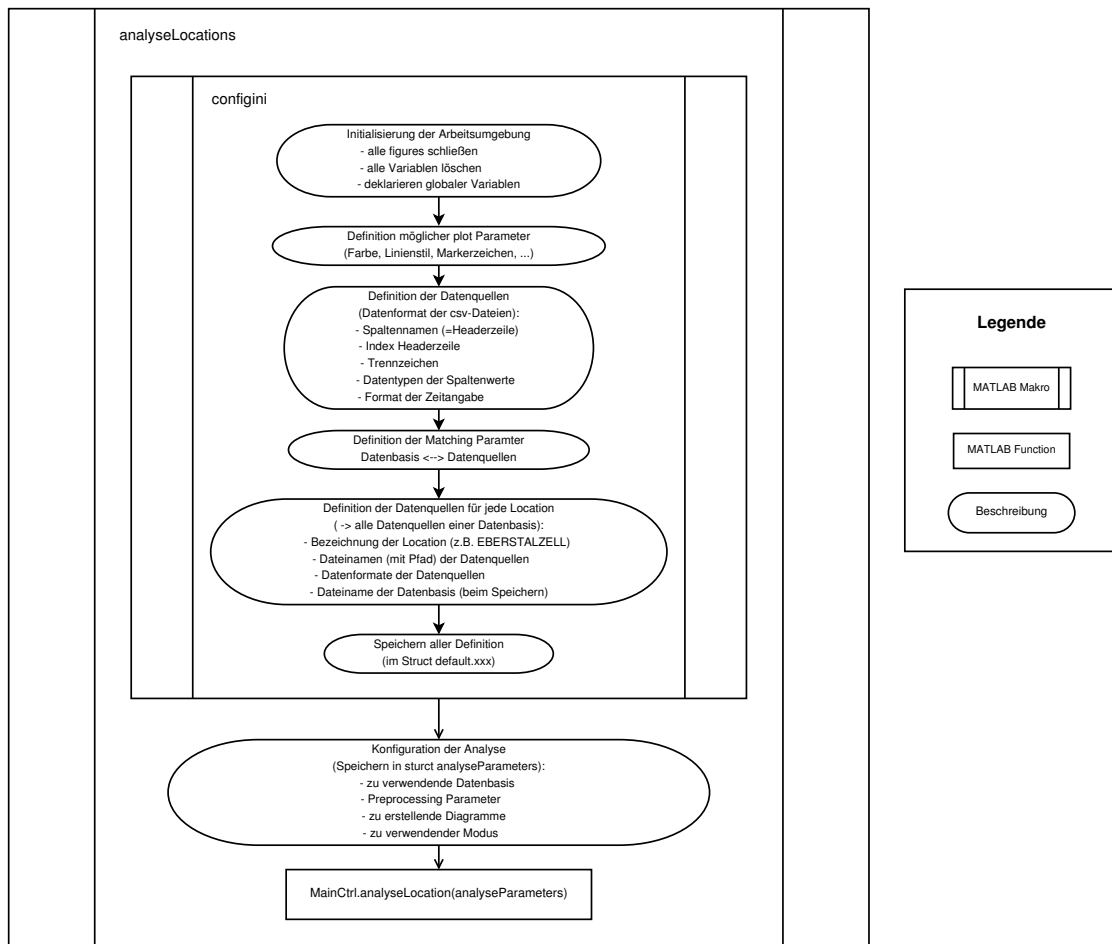


Abbildung 46: Schematischer Ablauf des MATLAB Skriptes zur Analyse

MATLAB Skript `configini.m` führt zuerst eine allgemeine Initialisierung der Arbeitsumgebung aus, welche in keinem MATLAB Skript fehlen sollte. Dadurch werden nämlich Fehler, welche durch vorhergehende Aufrufe des MATLAB Skriptes entstehen können (z.B. Variable oder Anfangswert einer Variable noch vom vorigen Skript Durchlauf vorhanden), vermieden.

Weiters erfolgt eine Definition der möglichen plot Parameter, welche von Methoden der Klasse `PlotCtrl` verwendet werden. Damit lassen sich automatisch Diagramme mit mehreren anzuzeigenden (und unterscheidbaren) Parametern erstellen.

Anschließend erfolgt die Definition jener Parameter der verschiedenen Datenquellen (→ csv-Dateien), welche von den Methoden, welche die jeweiligen csv-Dateien einlesen und daraus die zugehörigen MATLAB-Datenstrukturen erstellen, benötigt werden (siehe 3.4.3 ab Seite 42). Eine Beschreibung dieser Parameter mit Beispielen erfolgt in Tabelle 10.

Beschreibung	Bezeichnung (MATLAB)	Beispiel
Spaltennamen	Header	'Uhrzeit','Stat-Nr','Prior-Nr','Mode','Anzahl Hop',...
Index Headerzeile	HeaderLineIndex	10
Trennzeichen	delimiter	;
Datentyp der Spalten	formatString	'%s%f%f%s%d' ³
Format der Zeitangabe	formatTime	'dd.mm.yyyy HH:MM:SS:FFF' ⁴

Tabelle 10: Parameter zur Format Definition der Datenquellen (→ csv-Dateien)

Als nächster Schritt wird die Definition (im programmiertechnischen Sinne) der Matching Parameter der Datenbasis zu den einzelnen Datensätzen bzw. der einzelnen Datensätze untereinander durchgeführt. Dazu wird die Bezeichnung aller in einer Datenquelle (→ csv-Datei) vorkommenden Spalten (siehe dazu Tabelle 9 auf Seite 31) der Bezeichnung der Datenbasis (→ MATLAB interne Bezeichnung) zugewiesen. Die Speicherung der Matching Parameter erfolgt wiederum in einem Struct mit dem Namen `config.datasource.xxx.yyy`, wobei das `xxx` für die Bezeichnung der Datenquelle (→ csv-Datei) und das `yyy` für die Bezeichnung in der Datenbasis, steht. Zum besseren Verständnis dieser Speicherart ist in Tabelle 11 ein Beispiel angeführt.

Bezeichnung MATLAB	Datenquelle		Beispiel
	NAME	SPALTE	
serialNr	logDlc	Serial-Nr	logDlc.serialNr = 'Serial-Nr';
serialNr	onSheet2	Serialnr EnergieAG	onSheet2.serialNr = 'Serialnr EnergieAG';

Tabelle 11: Speicherung der Definition der Matching Parameter zw. der Datenbasis und den Datenquellen (→ csv-Dateien)

Anschließend wird die Definition aller zu einer Datenbasis für ein Testnetz gehörenden Datenquellen (→ csv-Dateien) durchgeführt. Dies beinhaltet sowohl das Format der Datenquellen, welches aus den zuvor definierten Formaten ausgewählt wird, als auch die Pfade, unter welchen die csv-Dateien abgespeichert sind. Die Speicherung erfolgt dabei wiederum in einem Struct, als Beispiel sei die Definition für den Pfad der LOG-DLC und SGTS csv-Datei angegeben:

```
analyseParameters.Littring.Data.filename.logDlc = './Eberstalzell/LOG-DLC.CSV';
analyseParameters.Littring.Data.filename.sgts = './Littring/08572_LITTRING.csv';
```

Um im Verlauf des Programmes die ursprünglichen Definitionen und Standardwerte leicht wiederherstellen zu können (wie z.B. das Setzen der Standard Linien- und Markertypen durch die Methode `PlotCtrl.setDefaultLinespec()`) wird als nächster Schritt die Speicherung aller definierten und initialisierten Variablen vorgenommen. Alle diese Variablen werden in einem Struct mit dem Namen `default` gespeichert, z.B.:
Struct `plotSettings` → `default.plotSettings`. Damit ist die allgemeine Konfiguration abgeschlossen und das MATLAB Skript `configini.m` wird beendet.

Als nächster Schritt wird nun die eigentliche Konfiguration der Analyseparameter im MATLAB Skript `analyseLocations.m` durchgeführt. Dazu zählt z.B. die für den nachfolgenden Analysevorgang (durch Aufruf der Methode `MainCtrl.analyseLocation` mit den nun definierten Analyseparametern \rightarrow Struct `analyseParameters`) zu verwendende Datenbasis. Weiters wird festgelegt ob diese zu verwendende Datenbasis neu erstellt werden soll oder eine abgespeicherte von früheren Durchläufen verwendet werden soll. Es können auch verschiedene Preprocessing Parameter angegeben werden, wie z.B. nach welchen Parametern der aus der Datenbasis erstellte `rbrDatasetTree` (siehe Kapitel 3.4.1.5) gefiltert/gruppirt werden soll (eine genauere Beschreibung des Preprocessings erfolgt im Kapitel 4.1.2.2). Weiters können noch die bei der Analyse zu erstellenden Diagramme konfiguriert werden, da ev. nicht bei jedem Analysedurchgang (z.B. Analyse mit verschiedenen Filter/Gruppierungsparametern) alle Diagramme benötigt werden. Außerdem können noch verschiedene Modi der Analyse eingestellt werden, wie z.B. ein Testmodus (\rightarrow einige Methoden führen erweiterte Tests der Daten und des eigenen Algorithmus durch) oder ein Entwicklungsmodus (\rightarrow kein Rücksprung aus der Methode `MainCtrl.analyseLocation` damit die lokalen Variablen nicht gelöscht werden).

Nachdem die gewünschten Analyseparameter definiert worden sind erfolgt nun die Analyse durch Aufruf der Methode `MainCtrl.analyseLocation(analyseParameters)` mit den gewünschten Parametern, welche an die besagte Methode zusammengefasst im Struct `analyseParameters` übergeben werden.

4.1.2.2 Vorbereitung der Datenbasis vor der Analyse

Nachdem im Programmablauf mittlerweile alle benötigten Analyseparameter definiert wurden können diese nun erstmals angewendet werden. Wie in Abbildung 47 (Seite 61) dargestellt, wird von der Methode `MainCtrl.analyseLocation` zuerst überprüft ob die Datenbasis neu erstellt oder eine bereits bestehende eingelesen werden soll. Falls die Datenbasis neu zu erstellen ist erfolgt dies wie bereits im Kapitel 3.4.2 (Seite 40) beschrieben. Anschließend wird die erstellte Datenbasis als *MAT-file*⁵ Datei abgespeichert. Dies erfolgt deshalb da das Erzeugen einer Datenbasis aus den Datenquellen (\rightarrow csv-Dateien) wesentlich länger dauert als das Laden derselben wenn sie bereits als *MAT-file* vorliegt.

Nachdem die Datenbasis für die Analyse nun vorliegt wird ein Preprocessing durchgeführt. Dabei werden einige Änderungen an der Datenbasis durchgeführt, wobei alle Schritte optional sind. Ob die optionalen Preprocessing-Schritte durchgeführt werden wird, wie bereits in Kapitel 4.1.2.1 (Seite 57) erwähnt, durch den Wert der entsprechenden Parameter im Struct `analyseParameters` festgelegt. Die möglichen Preprocessing-Schritte sind in Tabelle 12 (Seite 61) aufgelistet.

Nachdem nun die Datenbasis für die Analyse vorbereitet wurde, kann mit der Erstellung der Diagramme fortgefahren werden. Dies wird im nächsten Kapitel erläutert.

⁵MATLAB-Datendatei mit der Endung `.mat` in welcher Variablen gespeichert werden können

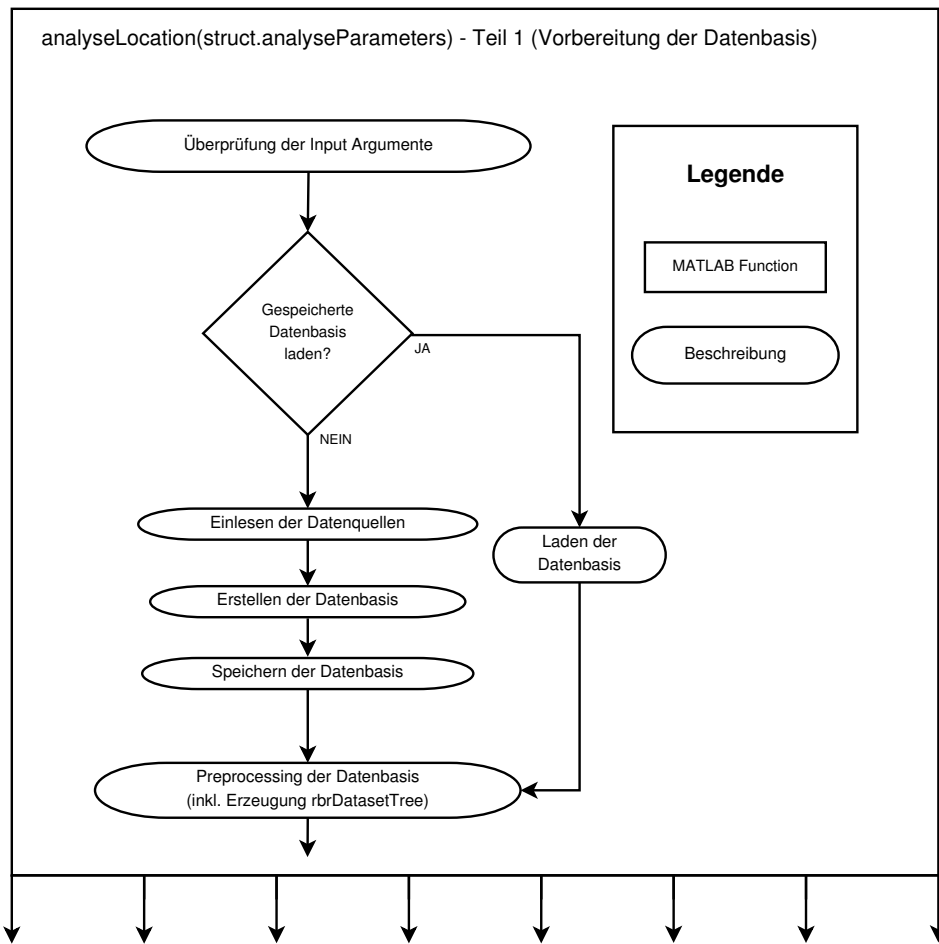


Abbildung 47: Schematischer Ablauf der MATLAB-Methode analyseLocation - Teil 1

Beschreibung	Grund (Beispiel)	verwendete Methode
alle Broadcasts löschen (stationNr 1023)	verfälscht Ergebnisse	removeRowWhereValueEqual
ungültige distance Einträge löschen (NaN → Not A Number)	sonst ev. Fehler bei manchen Methoden	removeRowWhereValueEqual
fortlaufende serialNr erstellen	bessere Darstellung in Histogrammen	direkt implementiert
fortlaufende connectionObjectNr erstellen	bessere Darstellung in Histogrammen	direkt implementiert
rbrDatasetTree erstellen	Filterung/Gruppierung der Datenbasis	getDcDatasetTreeSplitByParam

Tabelle 12: Preprocessing-Schritte der Datenbasis vor der Analyse

4.1.2.3 Erstellung der Diagramme

Wie aus Abbildung 48 (Seite 63) ersichtlich erfolgt die Erstellung der Diagramme in einer Schleife. Dies ist deshalb der Fall weil die Erstellung der Diagramme für jedes Blatt der in ein Datenstruktur vom Typ `rbrDatasetTree` umgewandelten Datenbasis durchgeführt wird. Bei jedem Schleifendurchlauf wird das nächste Blatt ausgewählt, genau genommen wird der Index durch die Methode `StatisticCtrl.iterateDcDatasetTreeIndex` auf den nächsten Wert (siehe auch Abbildung 32 auf Seite 42) gesetzt. Anschließend wird durch die Methode `StatisticCtrl.getCurrentDcDatasetFromTree` das Blatt, auf welches der aktuelle Index verweist, in einer lokalen Variable gespeichert und damit ein Schleifendurchlauf durchgeführt. Das bedeutet, dass nach einem optionalen Preprocessing-Schritt, welcher auf das aktuelle Blatt angewendet wird, alle gewünschten Diagramme erstellt werden. Dieser weitere Preprocessing-Schritt innerhalb der Schleife ist notwendig, da dabei nicht die Datenbasis (welche mittlerweile in Form eines `rbrDatasetTree` vorliegt) das Ziel ist, sondern nur das aktuelle Blatt derjenigen. Konkret können in diesem Preprocessing-Schritt alle Zeilen entfernt werden, welche einen Eintrag für `hopCount` enthalten, bei dem die Summe unter einer definierten Prozentmarke (z.B. 2%) bezogen auf die Gesamteinträge für dieses Blatt liegt. Anhand des in Tabelle 13 gezeigten Beispiels (Gesamtzahl der Zeilen = 100 000, Grenze = 2 %) ist diese Funktionalität praktisch ersichtlich.

hopCount	Anzahl der Zeilen		Konsequenz (bei Unterschreiten von 2%)
	absolut	relativ	
0	65 000	65 %	—
1	25 000	25 %	—
2	8 500	8,5 %	—
3	1 500	1,5 %	ALLE diese Zeilen werden gelöscht
ALLE	100 000	100 %	

Tabelle 13: Preprocessing des aktuellen Blattes vor der Diagrammerstellung

Nachdem der letzte Durchlauf der Schleife erfolgt ist, das bedeutet die gewünschten Diagramme wurden für alle Blätter des `rbrDatasetTree` (→ Filterung/Gruppierung) der Datenbasis erzeugt, werden alle erstellten Diagramme in ein automatisch erstelltes Verzeichnis verschoben. Der Verzeichnisname beinhaltet das Datum, die Zeit und weitere Informationen über die Erstellung. Zum Abschluss werden in diesem Verzeichnis noch die Analyseparameter im Struct `analyseParameters` gespeichert, um eine nachträgliche erweiterte Analyse bzw. Nachvollziehbarkeit der Ergebnisse mit der gewählten Filterung/Gruppierung zu ermöglichen.

4.1.3 Ergebnis der grafischen Analyse

Es erfolgt nun eine Beschreibung der Ergebnisse und Erkenntnisse, welche durch die grafische Analyse mit Hilfe der in Kapitel 4.1.1 (Seite 50) vorgestellten Diagramme, ge-

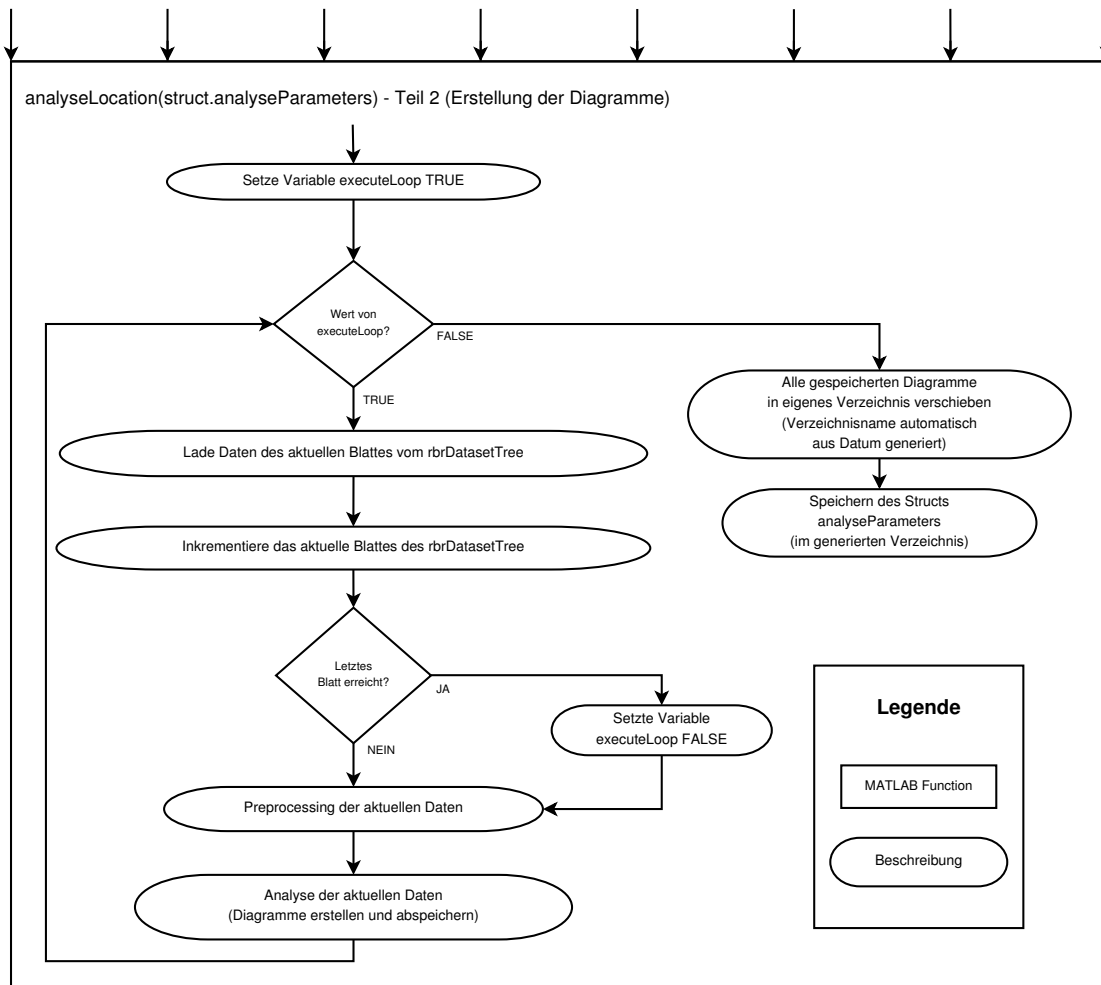


Abbildung 48: Schematischer Ablauf der MATLAB-Methode analyseLocation - Teil 2

wonnen werden konnten. Es wird darauf hingewiesen, dass sehr viele Diagramme erstellt wurden, und in dieser Diplomarbeit nur einige repräsentative Abbildungen zum besseren Verständnis der gewonnenen Erkenntnisse abgebildet sind.

Die Analyse konnte nur mit den Messdaten der Ortsnetze LITTRING und EBERSTALZELL durchgeführt werden, da für die anderen Ortsnetze keine für die Auswertung verwendbare Datenbasis erstellt werden konnte. Das Problem dabei war, dass die zur Verfügung stehenden AMIS-Messdaten für die Ortsnetze von ATTNANG-PUCHHEIM und VÖCKLABRUCK keine Messdaten enthielten, mit welchen ein Matching mit den Netzstrukturdaten durchgeführt werden konnte. Dass das Problem an den Messdaten, und nicht am Matching Algorithmus lag, wurde durch Einfügen einer kompletten Matching Kette in alle involvierten csv-Dateien (siehe Kapitel 3.3 ab Seite 29) verifiziert.

4.1.3.1 AMIS-Parameter Matrix

Abbildung 49 (Seite 64) zeigt eine AMIS-Parameter Matrix der Gesamtzahl an Paketen für die Parameter phase und hopCount für das Ortsnetz von EBERSTALZELL. Man kann erkennen, dass die Gesamtzahl der Pakete relativ gleich auf die Phasen verteilt ist, es gibt also keine ausgeprägte Tendenz eine Phase gegenüber den anderen zu bevorzugen.

	0	1	2	3	4	5	6	7	8
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	38.2513	51.2810	76.5052	176.5226	263.0413	256.0402	246.6000	307.3636	602.4164
2	32.9490	54.6622	76.5856	105.5681	227.8039	254.4559	260.9597	305.1923	587.5204
3	28.9629	61.7177	86.7613	114.8639	167.6861	252.7261	356.4737	342.4000	563.6258

Abbildung 49: AMIS-Parameter Matrix der Gesamtzahl an Paketen für die Parameter phase (1-3) und hopCount (0-8) für EBERSTALZELL

Abbildung 50 zeigt eine AMIS-Parameter Matrix der Gesamtzahl an Paketen für die Parameter mode und hopCount für das Ortsnetz von EBERSTALZELL. Hier zeigt sich ein deutlicher Unterschied bezüglich der Gesamtzahl der Pakete für die einzelnen Modi. Für einen Großteil der 16 Modi sind keine Pakete in den AMIS-Messdaten enthalten, auf die enthaltenen Modi verteilen sich die Pakete teilweise sehr ungleichmäßig.

	0	1	2	3	4	5	6	7	8
0	NaN	168	343.8214	457.6403	570.3482	NaN	NaN	NaN	1.0212e+03
1	NaN	126	244.9659	324.8832	404.9936	NaN	NaN	NaN	723.4907
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
6	44.5363	72.0638	159.8153	251.8295	319.1429	389.0915	453	NaN	580.6412
7	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9	35.9965	63.5175	88.1624	170.1464	225.2481	288.9398	336.2500	383	430.4125
10	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
11	27.0394	49.1564	72.6955	96.2838	142.9465	211.4388	255.5808	304.1343	315.8772
12	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
13	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
14	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
15	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Abbildung 50: AMIS-Parameter Matrix der Gesamtzahl an Paketen für die Parameter mode (0-15) und hopCount (0-8) für EBERSTALZELL

4.1.3.2 Korrelationsanalyse

Abbildung 51 zeigt ein Korrelationsmatrix-Diagramm (Beschreibung siehe Kapitel 4.1.1.4 auf Seite 55) für die Parameter distance, hopCount und nrOfBusBars. Es ist ersichtlich, dass zwischen den Parametern distance und nrOfBusBars ein näherungsweise linearer Zusammenhang besteht. Zwischen den Parametern hopCount und distance bzw.

nrOfBusBars besteht jedoch kein linearer Zusammenhang, durch die Auswertung von Korrelationsmatrix-Diagrammen für die Parameter hopCount in Kombination mit anderen Parametern (SNR, nrOfConnections, ...) konnte ebenfalls kein linearer Zusammenhang festgestellt werden.

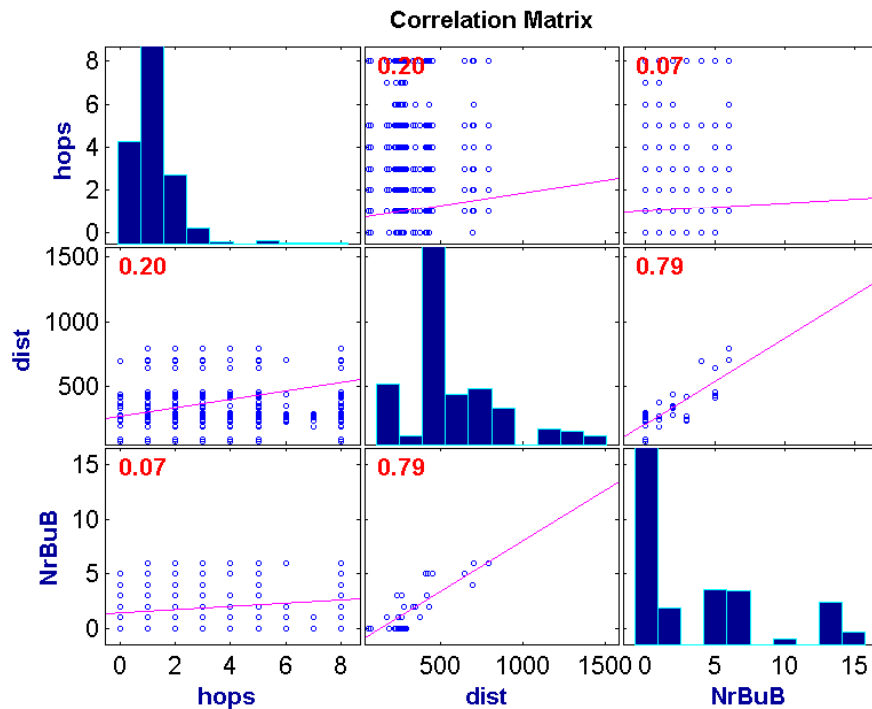


Abbildung 51: Korrelationsmatrix-Diagramm für die Parameter distance, hopCount und nrOfBusBars für EBERSTALZELL

4.1.3.3 Zeitanalyse

Abbildung 52 (Seite 66) zeigt ein XY-Diagramm mit Jitter (Beschreibung siehe Kapitel 4.1.1.3 auf Seite 53). Es lässt sich erkennen, dass der hopCount von der Tageszeit abhängt, da die Punktwolken ungleich verteilt sind. Um auszuschließen, dass die unterschiedlich dichten Punktwolken durch die Darstellungsform auf Grund einer ungleichen Verteilung der Gesamtanzahl an Daten für jeden Wert des Parameters hoursOfDay zustande kommt, ist in Abbildung 53 (Seite 66) das zugehörige Histogramm abgebildet. Es zeigt die gleichmäßige Verteilung der Daten über den gesamten Wertebereich des Parameters hoursOfDay (0 bis 23).

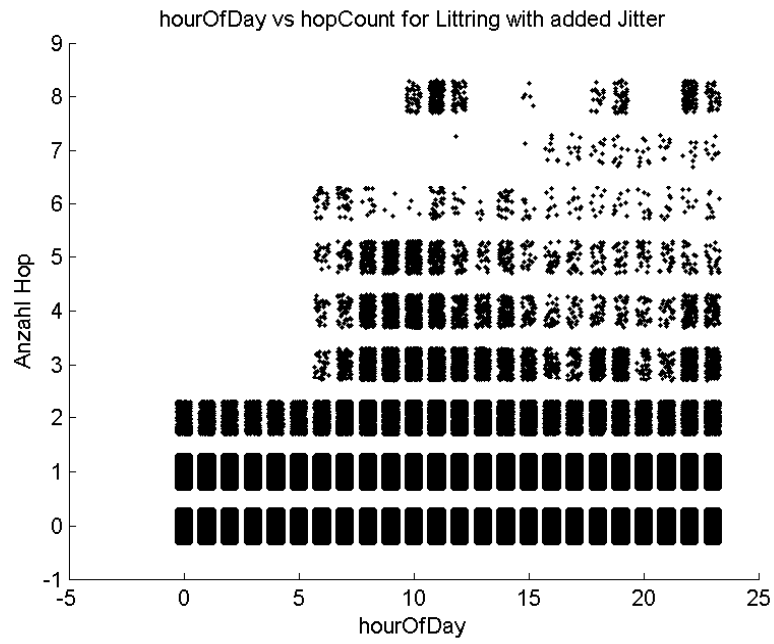


Abbildung 52: XY-Diagramm mit Jitter der Parameter hourOfDay vs. hopCount für LITTRING

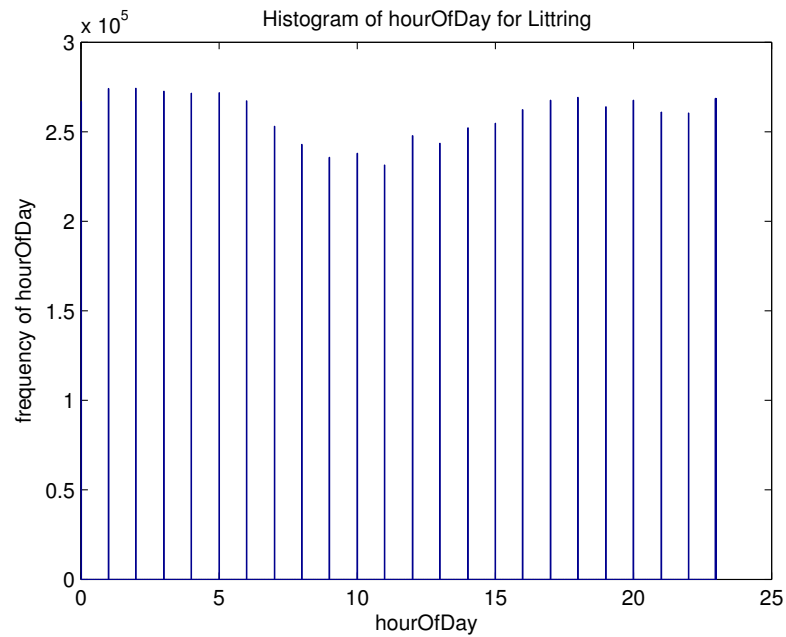


Abbildung 53: Histogramm des Parameters hourOfDay für LITTRING

4.1.3.4 Schlussfolgerungen

Zusammenfassend lässt sich über die nun erfolgte, grafische Analyse sagen, dass das Ergebnis des *hopCount* der AMIS-Kommunikation von einigen Parametern gleichzeitig beeinflusst wird. Es konnte kein einfacher und eindeutiger (z.B. linearer) Zusammenhang des Parameters *hopCount* mit den anderen, in der Datenbasis zur Verfügung stehenden, Parameter (*distance*, *SNR*, *nrOfBusBars*, *nrOfConnections*, ...) gefunden werden.

Somit kann kein lineares (und damit relativ einfaches) Modell zur Berechnung des resultierenden *hopCount* verwendet werden. Vielmehr ist ein nichtlinearer Zusammenhang zwischen den Parametern der Netzstruktur und dem daraus resultierenden *hopCount* zu erwarten.

4.2 Heuristische Analyse der Datenbasis

Da durch die grafische Analyse kein Modell für die Vorhersage der AMIS-Kommunikationsparameter auf Grund von Daten über die Netzstruktur entwickelt werden kann, wird nun ein andere Ansatz verfolgt. Dabei sollen Nichtlinearitäten und Wechselwirkungen zwischen den Parametern *distance*, *nrOfBusBars* und *nrOfConnections* berücksichtigt werden. Da auch die Parameter *hourOfDay* und *mode* einen Einfluss auf den resultierenden *hopCount* haben (siehe dazu Kapitel 4.1.3.3 auf Seite 65 und Kapitel 4.1.3.1 auf Seite 64) müssen diese beiden Parameter ebenfalls berücksichtigt werden. Da es sich bei diesen beiden Parametern um nominale Parameter handelt, erfolgt dies in Form einer Filterung/Gruppierung der Input Daten (→ Erzeugung eines *Struct.rbrDatasetTree*, siehe Kapitel 3.4.1.5 auf Seite 41) für die Heuristik.

Die für diese Analyse zur Verfügung stehenden Input Daten stammen von den Testnetzen LITTRING und EBERSTALZELL. Da diese beiden Testnetze eine sehr unterschiedliche Netzstruktur (*distance*, *nrOfBusBars*, *nrOfConnections*, ...) haben, lässt sich damit sehr gut feststellen, ob eine Vorhersage des *hopCount* für eine generische Netzstruktur anhand der gewählten Parameter prinzipiell möglich ist.

4.2.1 Allgemeine Beschreibung der Heuristik

Abbildung 54 (Seite 68) zeigt, sehr allgemein gehaltenen, den Aufbau der verwendeten Heuristik. Eine Beschreibung der genannten Abbildung erfolgt in diesem Kapitel mit Hilfe von MATLAB Code und textuellen Beschreibungen. Eine zusammengefasste Übersicht der Zuordnung der im MATLAB Code verwendeten Variablen zu den Bezeichnungen von Abbildung 54 (Seite 68) ist in Tabelle 14 (Seite 69) ersichtlich.

Für diese Analyse wird eine in MATLAB vorhandene Funktionalität⁶, nämlich der *Genetic Algorithm Solver (GA-Solver)* mit *Parameterized Fitness Function*, wie sie in [MATb] beschrieben ist, verwendet.

⁶vorhanden in der Global Optimization Toolbox

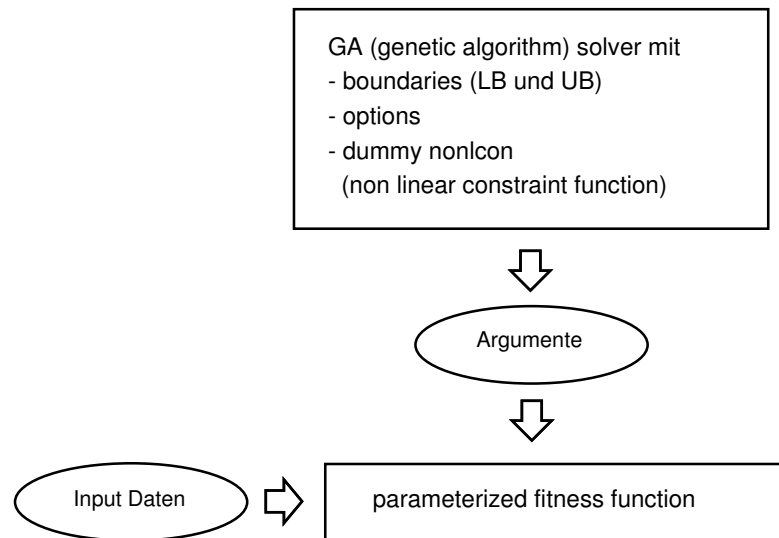


Abbildung 54: MATLAB Heuristik - Allgemeiner Aufbau

Parameterized Fitness Function

Bei diesem Ansatz wird versucht, in mehreren Runden durch Anpassung von Argumenten, deren Werte durch einen genetischen Algorithmus erzeugt werden, das Ergebnis einer sogenannten *Fitness Function* zu minimieren (Wichtig: nicht an 0 annähern, sondern wirklich minimieren!). Da die für diese Analyse verwendete *Fitness Function* für die vorhandenen Werte in der Datenbasis für die Analyse (konkret für die Parameter `hopCount`, `distance`, `nrOfBusBars`, `nrOfConnections` davon) optimiert werden soll ist es notwendig, auch Input Daten an die *Fitness Function* übergeben zu können. Daher muss eine *Parameterized Fitness Function* verwendet werden. Der folgende MATLAB Code beschreibt eine beispielhafte Definition solch einer Funktion:

```
function y = parameterized_fitness(x,a,b)
y = a * (x(1)^2 - x(2))^2 + (b - x(1))^2;
```

Boundaries (Grenzwerte)

Die Konfiguration von Boundaries (\rightarrow Grenzwerte), welche dem GA-Solver übergeben werden, sorgt dafür, dass die Werte der Argumente, welche durch den GA-Solver an die (*Parameterized*) *Fitness Function* übergeben werden, einen definierten Grenzwert nicht überschreitet (UB \rightarrow UpperBoundary) bzw. unterschreitet (LB \rightarrow Lower Bound). Durch diese Einschränkung des Argumentbereiches lassen sich Ergebniswerte schneller bzw. überhaupt erst finden. Diese Funktionalität kann in diesem Fall deshalb verwendet werden, da der resultierende `hopCount` zwischen 0 und 8 liegen muss. Die genaue Beschreibung, wie die Boundary Werte konkret berechnet werden, erfolgt in Kapitel 4.2.1.2

(Seite 70), zuerst jedoch noch ein allgemeines Beispiel:

Gewünschte Grenzwerte:

$$\begin{aligned} 0 &\leq x(1) \leq 1 \\ 0 &\leq x(2) \leq 13 \end{aligned}$$

MATLAB Code für die Festlegung von LB und UB:

```
LB = [0 0]; % Lower bound
UB = [1 13]; % Upper bound
```

Non Linear Constraints

Die Funktionalität der *Non Linear Constraint Function (nonlcon)* wird hier nicht verwendet. Vom MATLAB GA-Solver wird diese jedoch zwingend verlangt, weshalb eine dummy Funktion definiert werden muss. Diese liefert, wie in [MATa] beschrieben, dem GA-Solver das für seinen Programmablauf benötigte leere Array zurück, hat jedoch keinen Einfluss auf das Ergebnis.

Abbildung 54	MATLAB Variable	GA Solver Bezug
Argumente	x (MATLAB Vektor)	Erzeugung auf Basis eines genetischen Algorithmus
Input Daten	a und b	werden nicht verändert (Verwendung durch die Parameterized Fitness Function)
LB (Lower Bound)	LB (MATLAB Vektor)	Beschränkung der Argumentwerte (x)
UB (Upper Bound)	UB (MATLAB Vektor)	Beschränkung der Argumentwerte (x)
nonlcon	_____	nicht verwendet (Rückgabewert: leeres Array)

Tabelle 14: Verknüpfung von Bezeichnungen in Abbildung 54 (Seite 68) und Variablen des MATLAB Codes (Seite 68ff) für eine *Parameterized Fitness Function (y)*

4.2.1.1 Definition der Parameterized Fitness Function

Eine allgemeine, mathematische Beschreibung der für diese Analyse verwendeten *Parameterized Fitness Functions (PFF)* lautet wie folgt:

$$\begin{aligned} PFF = \sum &|f(\text{distance} \times \text{arg}_1) + f(\text{nrOfBusBars} \times \text{arg}_2) + f(\text{nrOfCon} \times \text{arg}_3) + \\ &f(\text{distance} \times \text{arg}_4^2) + f(\text{nrOfBusBars} \times \text{arg}_5^2) + f(\text{nrOfCon} \times \text{arg}_6^2) + \\ &f(\text{distance} \times \text{nrOfBusBars} \times \text{arg}_7) + f(\text{distance} \times \text{nrOfCon} \times \text{arg}_8) + \\ &f(\text{nrOfBusBars} \times \text{nrOfCon} \times \text{arg}_9) - \\ &\text{hopCount}| \end{aligned}$$

Hierbei steht die Multiplikation (\times Symbol) für eine Multiplikation welche MATLAB auf

einen Vektor anwendet⁷, es wird also der Wert jeder Zeile mit dem entsprechenden Wert multipliziert, das Ergebnis ist wiederum ein Vektor. Die Funktion $f()$ ist optional und kann z.B. eine Rundungsfunktion sein. Die Summe aller berechneten Datenwerte multipliziert mit den jeweiligen Argumenten ergibt den für diese Argumentwerte berechneten hopCount. Von dieser Summe wird der tatsächlich vorhandene hopCount abgezogen und vom Ergebnis der Betrag berechnet, es wird also die absolute Abweichung der Berechnung/Vorhersage vom tatsächlichen Wert ermittelt. Die Summe über alle absoluten Abweichungen ($\rightarrow \sum$ Symbol) wird nun vom GA-Solver durch heuristische Wahl der Argumente minimiert. Durch die Anwendung der Betragsfunktion ist das optimale Ergebnis 0. Der Wert der Summe über alle absoluten Abweichungen ermöglicht nun eine Aussage darüber, wie gut die (vom GA-Solver) gewählten Argumente eine Berechnung der tatsächlichen hopCounts ermöglichen.

4.2.1.2 Berechnung der Boundaries (Grenzwerte)

Wie bereits in Abschnitt 4.2.1 (Seite 68) beschrieben werden dem GA-Solver auch Boundaries (Grenzwerte), welche für die Argumente gelten, übergeben. Da die Werte aller Parameter (distance, nrOfBusBars, nrOfConnections) der Input Daten nicht negativ sind und somit die Anzahl der hopCounts nur positiv beeinflussen können, werden die unteren Grenzwerte (LB) für alle Argumente mit 0 festgelegt. Die Festlegung der oberen Grenzwerte (UB) gestaltet sich etwas aufwändiger. Da die maximale Anzahl des hopCount 8 beträgt und die *Parameterized Fitness Functions (PFF)* so aufgebaut ist, dass die Werte der Parameter mit je einem der Argumente multipliziert werden, kann das Ergebnis dieser Multiplikation maximal 8 betragen. Der Grenzfall, bei welchem als Ergebnis 8 auftreten kann ist jener, bei dem nur ein Argument ungleich 0 ist und der Wert des zugehörigen Parameters maximal ist. Der obere Grenzwert (UB) für jedes Argument wird deshalb als Kehrwert des maximalen Wertes des zugehörigen Parameters mal 8 festgelegt:

$$\begin{aligned} UB_{arg1} &= 8 \times \frac{1}{\max(\text{distance})} \\ UB_{arg2} &= 8 \times \frac{1}{\max(\text{nrOfBusBars})} \\ UB_{arg3} &= 8 \times \frac{1}{\max(\text{nrOfCon})} \\ UB_{arg4} &= 8 \times \frac{1}{\max(\text{distance})^2} \\ UB_{arg5} &= 8 \times \frac{1}{\max(\text{nrOfBusBars})^2} \\ UB_{arg6} &= 8 \times \frac{1}{\max(\text{nrOfCon})^2} \\ UB_{arg7} &= 8 \times \frac{1}{\max(\text{distance}) \times \max(\text{nrOfBusBars})} \\ UB_{arg8} &= 8 \times \frac{1}{\max(\text{distance}) \times \max(\text{nrOfCon})} \\ UB_{arg9} &= 8 \times \frac{1}{\max(\text{nrOfBusBars}) \times \max(\text{nrOfCon})} \end{aligned}$$

⁷MATLAB Operator .* anstatt *

4.2.2 Implementierung in MATLAB

In Abbildung 55 (Seite 72) ist die Funktionsweise des implementierten MATLAB Skripts `calculateParamsGenetic.m` ersichtlich. Dieses Skript stellt die Implementierung der bisher in Kapitel 4.2 beschriebenen heuristischen Analyse dar.

Zuerst werden die Parameter, welche der GA-Solver für eine Ausführung benötigt, konfiguriert. Danach werden die Input Daten geladen, welche zuvor mit dem MATLAB Skript `prepareDcDatasetGenetic.m` erzeugt wurden. Dieses Skript löscht alle Broadcasts und weiters alle ungültigen Einträge der gewünschten Datenbasis und speichert diese dann als *MAT-file*⁸ ab. Diese Vorgehensweise (Laden von vorbereiteten Input Daten) dient dazu, die Ausführungszeit des Skriptes `calculateParamsGenetic.m` zu verringern. Nach dem Laden der Input Daten erfolgt optional eine Filterung/Gruppierung der Input Daten.

Die Speicherung der Input Daten und der maximalen Werte davon, welche für die später Berechnung der Boundaries (Grenzwert) notwendig sind, erfolgt in einer Schleife für alle Blätter des Baumes vom Typ `struct.rbrDatasetTree`⁹. Bei jedem Durchlauf können optional durch einen weiteren Preprocessing-Schritt alle Reihen mit selten vorkommenden `hopCount` Werten, wie bereits in Kapitel 4.1.2.3 (Seite 61) genau beschrieben, in den Daten des aktuellen Blattes gelöscht werden.

Die eigentliche Berechnung der Werte der Parameter für die gewünschte *Parameterized Fitness Functions (PFF)* erfolgt in einer weiteren Schleife. Zuerst werden in dieser die tatsächlichen Boundaries (Grenzwert) für die Werte der Argumente, welche durch den GA-Solver festgelegt werden, berechnet (Details dazu siehe 4.2.1.2 auf Seite 70). Anschließend wird die zu Beginn des Skriptes `calculateParamsGenetic.m` bereits konfigurierte *Parameterized Fitness Functions (PFF)* mit den für diesen Schleifendurchlauf gültigen Input Daten für eine Verwendung durch den GA-Solver, wie in [MATb] beschrieben, vorbereitet. Da nun alle vom GA-Solver benötigten Daten zur Verfügung stehen, kann die Berechnung der Werte der Parameter für die gewünschte *Parameterized Fitness Functions (PFF)* erfolgen. Bei jedem Schleifendurchlauf werden die optimalen Werte und die zugehörige Summe der Abweichungen (vom `hopCount` der Input Daten) abgespeichert.

Nach Berechnung der Werte für alle Blätter werden die Abweichungen aller Schleifendurchläufe aufsummiert. Damit erhält man eine Aussage über die Vorhersagbarkeit des `hopCount` für die gewählte *Parameterized Fitness Functions (PFF)* mit den berechneten (optimierten) Werten der zugehörigen Parameter. Je niedriger diese Summe der Abweichungen ist, desto besser. Ein Wert von Null würde bedeuten, dass der `hopCount` aller Zeilen der Input Daten richtig berechnet werden könnte.

⁸MATLAB Datendatei mit der Endung `.mat` in welcher Variablen gespeichert werden können

⁹Falls keine Filterung/Gruppierung erfolgt ist existiert nur ein Blatt mit allen Daten.

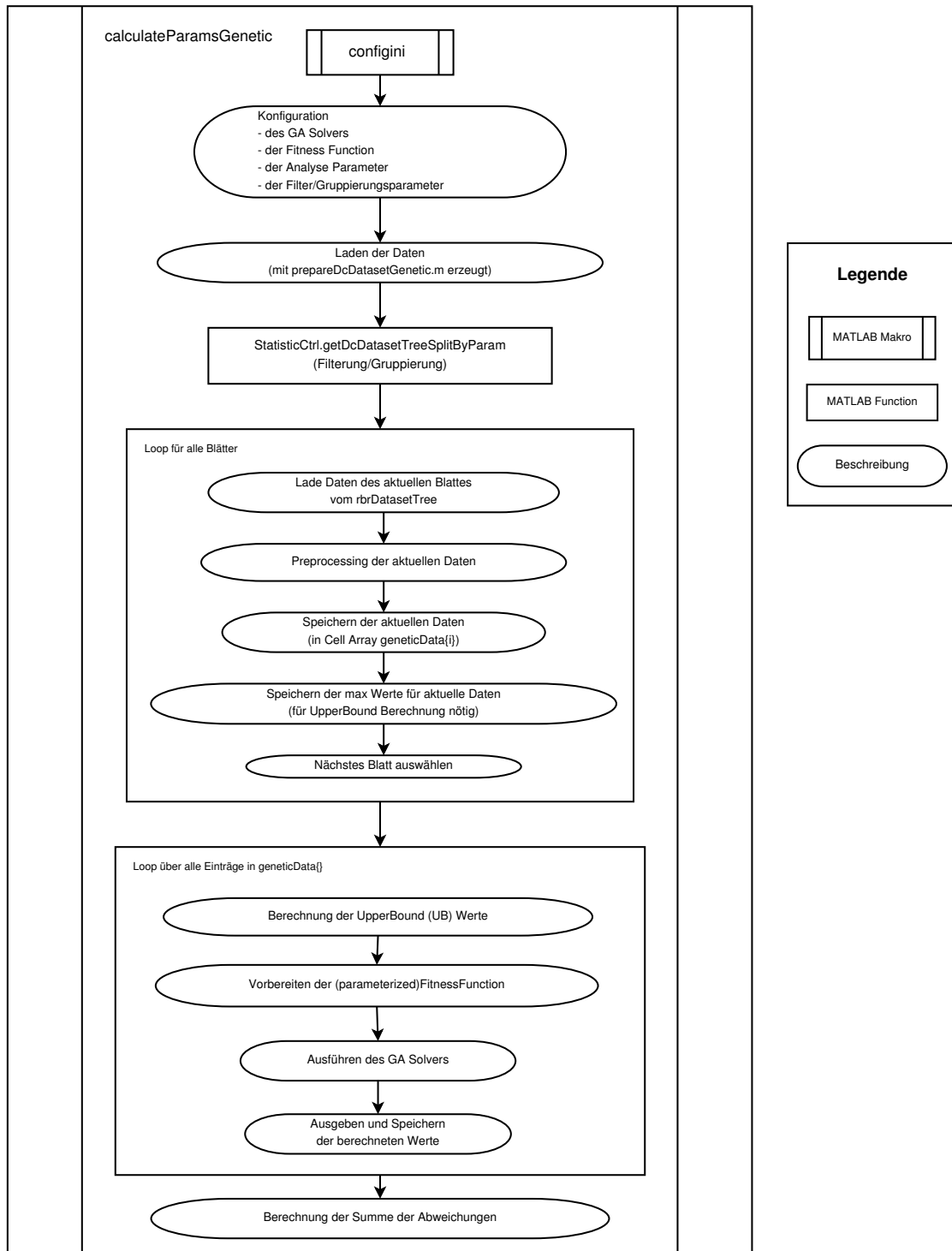


Abbildung 55: Ablauf des Skripts zur heuristischen Berechnung der Gewichtungparameter

4.2.3 Ergebnis der heuristischen Analyse

Die für die Testnetze EBERSTALZELL und LITTRING durchgeführte heuristische Analyse brachte die in Tabelle 15 (Seite 73) zusammengefassten Ergebnisse. Bezüglich der Summe der minimalen Abweichungen, welche für eine optimale Vorhersage des hopCount Null sein sollte, zeigt sich ein deutlicher Unterschied zwischen den beiden Testnetzen. Da diese beiden Testnetze bezüglich ihrer Netzstruktur (distance, nrOfBusBars, nrOfConnections) sehr unterschiedlich aufgebaut sind kann man daraus folgern, dass eine allgemeine Vorhersage des zu erwartenden hopCount bei Kenntnis der verfügbaren Informationen über die Netzstruktur für die untersuchten *Parameterized Fitness Functions* (→ Ausbreitungsmodell) nicht möglich ist.

Preprocessing der Input Daten	EBERSTALZELL	LITTRING
—	220 000	12 000
Grenze 10%	190 000	7 000

Tabelle 15: Minimale Summe der Abweichungen des hopCount der heuristischen Analyse

4.3 Verbesserung des Kommunikationssimulators

Da es durch die bisher erfolgten Analysen nicht gelang, ein brauchbares Modell für die allgemeine Vorhersage der AMIS-Kommunikationsparameter anhand der vorhandenen Daten über die Netzstruktur zu erstellen, wird nun das Ziel neu festgelegt. Es wird nunmehr keine allgemeine Vorhersage für allgemeine, neue Netze mehr angestrebt, sondern eine Verbesserung der Simulation bekannter Netze, für welche bereits AMIS-Messdaten vorliegen. Durch eine Erweiterung und Verbesserung der dem ICT zur Verfügung stehenden statistischen Auswertemöglichkeiten von AMIS-Messdaten kann mit dem Kommunikationssimulator ein genaueres Simulationsergebnis erreicht werden.

4.3.1 Aufbau der Input Datei

Als Input Datei, welche von dem am ICT entwickelten Kommunikationssimulator für die Simulation der Ausbreitungseigenschaften von AMIS-Paketen benötigt wird, wird eine csv-Datei verwendet. Ein beispielhafte Auszug aus solch einer (bisher verwendeten) csv-Datei sieht wie folgt aus:

```
Prior,Mode,Hops,Delay_min,Delay_mid,Delay_max,Sum_Acked,Lost_no,Lost_delay
27,0,1,0,0,0,0,0,0
27,0,2,0,0,0,0,0,0
...
...
27,4,4,0,0,0,0,0,0
27,6,7,0,0,0,0,0,0
27,9,0,28,38,59,29260,14240,63
```


27,10,6,0,0,0,0,0,0
 27,10,7,0,0,0,0,0,0
 27,10,8,0,0,0,0,0,0
 27,11,0,19,28,43,53807,32263,46
 27,11,1,46,52,65,169,559,64
 27,11,2,68,73,94,84,268,92
 27,11,3,90,106,122,88,239,121
 27,11,4,115,132,151,52,149,149
 27,11,5,137,157,179,35,78,178
 27,11,6,0,0,0,0,0,0
 27,11,7,0,0,0,0,0,0
 ...
 ...
 27,11,8,0,0,0,0,0,0

In Tabelle 16 (Seite 74) ist die Bedeutung jeder Spalte einer solchen csv-Datei beschrieben. Sind für ein Tupel der Filterparameter keine AMIS-Messwerte in der Datenbasis vorhanden, wird für alle statistischen Parameter der Wert Null in die csv-Datei geschrieben.

Spaltenname	Beschreibung
FILTERPARAMETER FÜR BERECHNUNG	
connectionObjectNr (NEU)	Nummer des Hausanschlusses, von EnergieAG Netz OÖ GmbH vergeben. Konnte in den bisherigen Auswertungen des ICT nicht berücksichtigt werden!!
Prior (priorNr)	beschreibt die Funktion, z.B. STATIONS Suche oder STATUS Abfrage
Mode (mode)	Übertragungsmodus, verschiedene Bitraten und Modulationen [AMI11, S. 15,41]
Hops (hopCount)	Anzahl der Stationen zwischen DC (Master) und Endgerät (Slave)
BERECHNETE STATISTISCHE PARAMETER (FÜR JEDES FILTERTUPEL)	
Delay_min	Minimum des delay
Delay_mid	Median des delay
Delay_max	Maximum des delay
Sum_Acked	Anzahl der gültigen Pakete
Lost_no	Anzahl der verloren gegangenen Pakete
Lost_delay	Maximales delay der verloren gegangenen Pakete

Tabelle 16: Beschreibung der Attribute einer csv-Datei mit statistischer Auswertung

4.3.2 Erstellung der Input Datei

Zur Erstellung der zuvor beschriebenen csv-Datei wurde für diese Diplomarbeit ein MATLAB Skript geschrieben. Dieses Skript `generateStatisticCsvForNetworkSimulator` verwendet einige bereits für die Aufbereitung und Analyse der Datenbasis verwendeten Methoden. Weiters wurde eine zusätzlich Methode implementiert welche statistische Berechnungen durchführt (`MainCtrl.calculateStatisticalParams`).

Im MATLAB Skript `generateStatisticCsvForNetworkSimulator` wird nach der Konfiguration der benötigten Variablen zuerst die zu verwendende Datenbasis geladen. Anschließend wird die MATLAB Methode `MainCtrl.calculateStatisticalParams` ausgeführt, welche die Berechnung der statistischen Parameter (siehe Tabelle 16, Seite 74) für eine übergebene Datenbasis durchführt und als Rückgabewert ein Array mit den berechneten statistischen Werten liefert. Dieses Array wird abschließend in eine csv-Datei geschrieben.

Von der implementierten MATLAB Methode `MainCtrl.calculateStatisticalParams` wird die MATLAB Methode `myStatisticCtrl.getDcDatasetTreeSplitByParam` verwendet um alle möglichen Kombinationen der Filterparameter zu berechnen. Jedes Blatt des von der MATLAB Methode `myStatisticCtrl.getDcDatasetTreeSplitByParam` zurückgelieferten Datenstruktur vom Typ `rbrDatasetTree` beinhaltet nur Daten eines Filtertupels. Somit können diese Daten als Inputdaten für die statistischen Berechnung für eine Zeile in der zu erzeugenden csv-Datei verwendet werden. In einer Schleife wird über alle Blätter iteriert und die statistischen Werte für alle möglichen Kombination der Filterparameter berechnet und in einem Array gespeichert. Nach Berechnung der statistischen Werte für das letzte Blatt wird das Array an den Aufrufer (in diesem Fall das MATLAB Skript `generateStatisticCsvForNetworkSimulator`) zurückgegeben.

4.4 Ergebnis der Verbesserung des Kommunikationssimulators

Durch die Erweiterung der Input-Datei des Kommunikationssimulators um den Parameter `connectionObjectNr` ist nunmehr eine viel detailliertere und damit genauere Simulation von Testnetzen, für welche AMIS-Messdaten und Daten über die Netzstruktur vorliegen, möglich. Durch die, dem ICT bisher zur Verfügung stehenden, statistischen Auswertungen war es lediglich möglich, die Verteilung der AMIS-Kommunikationsparameter der gesamten AMIS-Messdaten eines Testnetzes zu berechnen. Durch die nunmehr mögliche Zuordnung der `connectionObjectNr` zu den AMIS-Messdaten kann diese Verteilung für jedes AMIS-Endgerät individuell bestimmt werden, für welches AMIS-Messdaten vorliegen.

5 Diskussion

Es erfolgt nun eine Betrachtung und kritische Diskussion der durch diese Diplomarbeit hervorgebrachten Resultate und Erkenntnisse aus verschiedenen Blickwinkeln. Zuerst erfolgt dies für den Datenaufbereitungsteil (Kapitel 3, ab Seite 18), danach wird der Analyseteil (Kapitel 4, ab Seite 50) behandelt und abschließend erfolgt eine Auseinandersetzung mit der erfolgten Verbesserung des Kommunikationssimulators (Kapitel 4.3, ab Seite 73).

5.1 Diskussion der Datenaufbereitung

Die im Kapitel 3 (ab Seite 18) beschriebene Durchführung der, für die Analyse unbedingt erforderlichen, Datenaufbereitung, welche schließlich die Datenbasis für die Analyse liefern muss, ist ein aufwändiger Vorgang. Dies ist vor allem bedingt durch die Inhomogenität der verschiedenen benötigten Datenquellen. Diese unterscheiden sich sowohl im Dateiformat (csv-Dateien, xml-Datei, DIGSILENT-GIS-SCADA (dgs)-Datei, ...), als auch durch einen inkonsistenten Aufbau der Datenquellen. So wird teilweise je nach vorliegender Version der Datenquelle für die selbe Information in csv-Dateien eine andere Bezeichnung und/oder Position der Spalten verwendet. Aufgrund dieser, im Vorhinein nicht erwarteten, Komplexität beim Erstellen der Datenbasis für die Analyse, ist der Prozess selbst wie auch das dazu implementierte MATLAB Programm aufwändig und umfangreich.

Aufgrund der evolutionären Entwicklung (→ bei Auftreten einer neuen Datenquelle bzw. Version hinzufügen von Programmcode bzw. einer neuen Methode) des implementierten MATLAB Programms ist die derzeitige Struktur bzw. Aufteilung der Funktionalität nicht mehr optimal und es sollte, spätestens vor der Implementierung neuer Funktionalitäten, ein Redesign durchgeführt werden. So ist z.B. die Klasse `StatisticCtrl.m` ursprünglich für rein statistische Methoden gedacht gewesen, jedoch erweiterte sich die Funktionalität der darin enthaltenen Methoden im Laufe der Zeit zusehends. Besser wäre es mittlerweile einige der Methoden in eine eigene Klasse auszulagern, z.B. mit dem Namen `DataProcessingCtrl`. Da dies jedoch lediglich den logischen Aufbau betrifft und relativ viel Aufwand bedeuten würde, wurde dieser Schritt in der derzeitigen Version der MATLAB Implementierung noch nicht umgesetzt.

Die Funktionalität zur Erstellung der Datenbasis für die Analyse besteht im Wesentlichen aus dem Einlesen und Zusammenführen mehrerer, inhomogener Datenquellen in MATLAB. Die Konfiguration des Aufbaus der bereits definierten Datenquellen (z.B. in welcher Spalte welche Daten vorhanden sind) ist nicht im Programm selbst codiert (→ nicht hardgecoded), sondern durch Anpassung von MATLAB Variablen (z.B. durch MATLAB Konfigurationsskripte) an eventuell auftretende neue Versionen der Datenquellen anpassbar.

Um die Verwendung der erstellten Datenbasis für die Analyse (→ Datenstruktur vom

Typ `rbrDataset`) flexibel zu ermöglichen, werden die Daten dieser in einer Datenstruktur vom Typ `rbrDatasetTree` (siehe Abbildung 31, Seite 41) gespeichert. Die für diese Diplomarbeit in MATLAB implementierten Methoden, welche diese Datenstruktur erzeugen und die darin gespeicherten Daten manipulieren, erlauben trotz der Komplexität der verwendeten Algorithmen eine komfortable Verwendung dieser Datenstruktur. Besonders hilfreich ist die Möglichkeit, damit die Daten einer vorhandenen Datenbasis zu gruppieren bzw. zu filtern (siehe Abbildung 32, Seite 42). Allerdings kann die dafür benötigte Zeit bei einer entsprechend großen Datenbasis (mehrere Millionen Zeilen) einige Minuten betragen¹⁰.

5.2 Diskussion der Analyse

Wie bereits im Kapitel 4.1.3 (Seite 62) erwähnt, war die erste Herausforderung, die für die Analyse benötigten Daten überhaupt zu bekommen. Es konnten aufgrund fehlender Daten die Analysen nur für die beiden Testnetze LITTRING und EBERSTALZELL durchgeführt werden, da nur für diese beiden Testnetze eine vollständige, und damit sinnvoll verwendbare, Datenbasis erstellt werden konnte. Für die restlichen Testnetze (ATTNANG-PUCHHEIM und VÖCKLABRUCK) lagen keine zu den vorliegenden Automated Metering and Information System (AMIS)-Messdaten passenden Datensätze über die Netzstruktur vor, weshalb die Zuordnung der `connectionObjectNr` (und damit in weiterer Folge der `distance`) nicht möglich war und somit die Datenbasen unvollständig blieben. Da für diese unvollständigen Datenbasen durch die geplanten Analysen keine neuen Erkenntnisse zu erwarten waren, wurde die Analyse auf die vollständigen Datenbasen der Testnetze LITTRING und EBERSTALZELL beschränkt.

Durch die durchgeführte grafische Analyse (Histogramm, XY-Diagramm (mit und ohne zur Anzeige hinzugefügtem Jitter), Korrelationsmatrixdiagramm und AMIS-Parameter Matrix) der beiden Datenbasen zeigte sich, dass der gemessene `hopCount` der AMIS-Kommunikation von einigen Parametern gleichzeitig beeinflusst wird. Es konnte jedoch kein einfacher und eindeutiger (z.B. linearer) Zusammenhang des Parameters `hopCount` mit den anderen Parametern, welche ebenfalls in der Datenbasis zur Verfügung standen (`distance`, `SNR`, `nrOfBusBars`, `nrOfConnections`, ...), gefunden werden. Somit war es auch nicht möglich, ein einfaches Modell für die Vorhersage des `hopCount` auf Grund der Kenntnisse über die Netzstruktur zu erstellen.

Wegen der nun zu erwartenden Komplexität des Modells für die Vorhersage des `hopCount` wurde nach Vorliegen der Ergebnisse der grafischen Analyse ein heuristischer Ansatz für die Analyse gewählt. Die für die Testnetze EBERSTALZELL und LITTRING durchgeführte heuristische Analyse führte jedoch zu dem Ergebnis, dass mit dem gewählten Lösungsansatz (\rightarrow *Parameterized Fitness Function (PFF)*) das Finden eines Modells, welches die Vorhersage des `hopCount` für eine allgemeine Netzstruktur ermöglicht, nicht möglich ist bzw. einen extrem hohen Aufwand bedeuten würde. Das Problem bei dem

¹⁰verwendeter Computer: AMD64 Quad-Core System mit 8GB RAM

gewählten Ansatz liegt darin, dass dabei die Struktur der Formel (und damit das Modell), welche für die Berechnung des *hopCount* verwendet wird, festgelegt werden muss und lediglich Gewichtungparameter der einzelnen Faktoren durch die vorhandene Datenbasis angepasst/optimiert werden. Besser wäre daher wahrscheinlich ein Ansatz, bei welchem im Vorhinein keinerlei Wissen über die Ausprägung der Auswirkungen der Parameter der Netzstruktur auf den *hopCount* sowie die Ausprägung der Wechselwirkungen der Parameter der Netzstruktur untereinander benötigt wird. Dies wäre z.B. bei Ansätzen, welche Neuronale Netze oder Genetische Programmierung verwenden, der Fall. Durch Untersuchung der bereits vorhandenen sowie zukünftig verfügbarer Datenbasen von weiteren Testnetzen mit diesen Methoden ließe sich vielleicht ein Modell zur allgemeine Vorhersage des *hopCount* entwickeln.

5.3 Diskussion der Verbesserung des Kommunikationssimulators

Die Verbesserung des Kommunikationssimulators erfolgte durch die Erweiterung der Input-Datei (\rightarrow csv-Datei) des Kommunikationssimulators um Werte für den Parameter *connectionObjectNr* (siehe Kapitel 4.3 auf Seite 73). Da dabei die Verteilung der AMIS-Kommunikationsparameter für jedes AMIS-Endgerät einzeln berechnet wird, ist diese Beschreibung viel wirklichkeitsgetreuer als die bisher mögliche Berechnung der Verteilung der AMIS-Kommunikationsparameter für das gesamte Testnetz¹¹. Dadurch ist eine genauere Simulation der AMIS-Kommunikationsparameter für bekannte Testnetze möglich, sobald für diese AMIS-Messdaten und Daten über die Netzstruktur vorliegen.

Durch den meist angewendeten generischen Aufbau der implementierten MATLAB Methoden sollte es relativ einfach möglich sein, die statistische Auswertung zu verfeinern bzw. bei Bedarf weitere Parameter auszuwerten und der Input-Datei des Kommunikationssimulators hinzuzufügen. Natürlich müssten diese etwaigen neuen Parameter auch durch den Kommunikationssimulator unterstützt werden, weshalb auch dieser erweitert werden müsste.

¹¹der bisherige Ansatz entspricht quasi einer Mittelung über alle AMIS-Messdaten

6 Zusammenfassung und Ausblick

Das Ziel dieser Diplomarbeit war es, ein Modell für die Vorhersage von AMIS-Kommunikationsparametern in Niederspannungsnetzen zu entwickeln, welches vom am ICT der TU-Wien entwickelten Kommunikationssimulator verwendet werden kann. Dieses Modell sollte anhand von Daten über die Netzstruktur (Leitungslänge, Anzahl der Abzweigungen, Anzahl der Sammelschienen) den auftretenden *hopCount* vorhersagen können. Benötigt wird diese Funktionalität unter anderem für eine Abschätzung des Aufwandes bei Netzaus- und Netzbauten oder für die Analyse von fiktiven bzw. in Planung befindlichen Netzen.

Dazu mussten zuerst mehrere Datensätze mit unterschiedlichen Informationen von verschiedenen Dateien (csv-Dateien, xlsx-Dateien, xml-Dateien, dgs-Dateien) zu einer gemeinsamen Datenbasis, welche alle benötigten Informationen enthält, zusammengeführt werden. Dazu wurde ein Programm in MATLAB implementiert, welches die Daten aus csv-Dateien einlesen und die darin enthaltenen Informationen richtig verknüpfen kann. Von jenen Datenquellen, welche nicht im csv-Format vorlagen, musste zuerst mit Hilfe anderer Programme (PowerFactory von DIGSILENT; Smart Grids Topologie Simulator aus einer Bachelor Arbeit am ICT) eine csv-Datei erzeugt werden.

Die für zwei verschiedene Testnetze (EBERSTALZELL und LITTRING) erstellten Datenbasen wurden zuerst einer grafischen Analyse unterzogen. Die dazu benötigten Diagramme und Tabellen wurden durch selbst implementierte MATLAB Methoden erzeugt. Das Ergebnis dieser ersten Analyse war, dass mehrere verschiedene Faktoren (z.B. Leitungslänge, Übertragungsmodus, Tageszeit) einen Einfluss auf den resultierenden *hopCount* haben. Es konnte jedoch kein vorherrschender und damit direkt ersichtlicher Zusammenhang zwischen einem Parameter der Netzstruktur und des resultierenden *hopCount* gefunden werden.

Als Nächstes wurde versucht, mit Hilfe einer Heuristik, ein Modell für die Vorhersage des *hopCount* zu entwickeln. Es wurden dazu die gleichen Datenbasen wie für die grafische Analyse verwendete. Wiederum wurde dazu ein MATLAB Programm implementiert, wobei der in MATLAB bereits vorhandene *Genetic Algorithm Solver (GA-Solver)* mit *Parameterized Fitness Function* verwendet wurde. Jedoch konnte durch diesen Ansatz ebenfalls kein brauchbares Modell entwickelt werden.

Da die Entwicklung eines Modells für die Vorhersage des *hopCount* in Niederspannungsnetzen anhand der vorhandenen Parameter über die Netzstruktur nicht erreicht werden konnte, wurde die Zielsetzung abgeändert. Es wurde nun eine Verbesserung des Kommunikationssimulators derart angestrebt, dass die Vorhersage des *hopCount* für Niederspannungsnetze, für welche bereits AMIS-Messdaten vorliegen, verbessert wird. Die vor der Verbesserung verwendete Vorhersage beruhte auf einer statistischen Mittelung der AMIS-Kommunikationsdaten des gesamten Testnetzes, also aller AMIS-Messdaten, welche von einem Data Concentrator (DC) gespeichert werden. Die im Zuge dieser Di-

plomarbeit entwickelte Verbesserung besteht darin, dass nun eine statistische Auswertung der AMIS-Kommunikationsdaten für jedes einzelne AMIS-Endgerät erfolgen kann. Dies wird erst durch die Zuordnung der Hausanschlüsse (*connectionObjectNr*) zu den AMIS-Messdaten, welche im Zuge der Erstellung der Datenbasis erfolgt, ermöglicht, da diese Information in den AMIS-Messdaten nicht direkt enthalten ist. Die Verwendung der nun möglichen, in MATLAB implementierten, genaueren statistischen Auswertung durch den am ICT entwickelten Kommunikationssimulator wird durch Speicherung der Auswertungsergebnisse in einer csv-Datei sichergestellt.

Um das ursprüngliche Ziel dieser Diplomarbeit (Entwicklung eines Modells für die Vorhersage des *hopCount* in Niederspannungsnetzen bei Vorliegen von Daten über die Netzstruktur) eventuell doch noch zu erreichen, wäre wahrscheinlich eine Erweiterung der Datenbasis um weitere Informationen über die Netzstruktur, wie z.B. ein Parameter welcher den Leitungstyp beschreibt und ein Parameter wie die Leitung verlegt ist (z.B. Freileitung oder Erdkabel), notwendig. Ein Problem dabei stellt die Inhomogenität der Leitungen zwischen dem DC und den AMIS-Endgeräten und auch zwischen den einzelnen AMIS-Endgeräten selbst dar, weshalb eine Beschreibung entweder stark vereinfacht werden müsste oder sehr aufwändig wäre.

Um das Modell zu erstellen sollte außerdem der in dieser Diplomarbeit verwendete Ansatz eines *GA-Solver* mit *Parameterized Fitness Function* nicht verwendet werden. Stattdessen wären Ansätze, wie z.B. Neuronale Netzwerke oder Genetische Programmierung, bei welchen die Auswirkungen der einzelnen Parameter auf den *hopCount* und die Wechselwirkungen zwischen den Parametern im Vorhinein nicht festgelegt werden müssen, mit großer Wahrscheinlichkeit erfolgversprechender.

Weiters könnte auch das im Zuge dieser Diplomarbeit erstellte MATLAB Programm durch das Hinzufügen weiterer statistischer Auswertemöglichkeiten, z.B. Erstellung einer AMIS-Parameter Matrix für andere Parameterkombinationen als *hopCount* vs. *mode* und *hopCount* vs. *phase*, verbessert werden. Dies würde eine noch genauere statistische Auswertung der AMIS-Messwerte bekannter Testnetze ermöglichen, wodurch wiederum die Genauigkeit der Simulation dieser bekannten Testnetze weiter verbessert werden würde.

Literaturverzeichnis

- [Aic11] AICHELE, Christian: *SmartEnergy*. 2011. – ISBN 978-3-8348-1570-5. – eBook
- [AMI06] Siemens AG: *AMIS System Überblick*. 04.04.2006. – page 6
- [AMI11] Siemens AG: *Spezifikation von AMIS CX1-Profil*. 09 2011
- [DIg] DIGSILENT GMBH, GERMANY, publisher=DIgSILENT GmbH, Germany year=2011: *DIgSiILENT Interface Documentation*
- [FD11] FASCHANG, Mario ; DIMITRIOU, Pavlos: *Modellierung der AMIS-Kommunikation*. 2011. – Technischer Bericht; Projekt DG DemoNetz - Smart LV-Grid; Revision 2
- [FDLK12] FASCHANG, Mario ; DIMITRIOU, Pavlos ; LEBER, Thomas ; KUPZOG, Friedrich: *Coupled Simulation*. 11 2012. – unpublished
- [Ker06] KERSTING, William H.: *Distribution System Modeling and Analysis*. 2006. – ISBN 1439856222
- [Knu06] KNUTH, D.E.: *Informations- und Kommunikationstechnologien für die Energieversorgung von morgen*. 2006. – 403–422 S.
- [MK06] MAENOU, Takatoshi ; KATAYAMA, Masaaki: *Study on Signal Attenuation Characteristics in Power Line Communications*. 2006
- [P.02] P., Angelov P.: *Evolving Rule-Based Models*. 2002. – ISBN 3-7908-1457-1
- [SEB06] SOLARE ENERGIEVERSORGUNGSTECHNIK, Institut für ; BEZIRKSVEREIN, Verband Deutscher E.: *Informations- und Kommunikationstechnologien für die Energieversorgung von morgen: Elftes Kasseler Symposium Energie-Systemtechnik ; [9. - 10. November 2006 ; Tagungsband/Report]*. Institut für Solare Energieversorgungstechnik, 2006 <http://books.google.at/books?id=J2ayPgAACAAJ>

- [Spi05] SPIES, Marcus: *Unsicheres Wissen*. 2005. – ISBN 3-86025-006-X
- [Tom05] TOMASSINI, Marco: *Spatially Structured Evolutionary Algorithms*. 2005. – ISBN 3-540-24193-0
- [Zil] ZILLNER, Stefan: *Smart Grids Topologie Simulator, Bachelor Arbeit*. – unpublished

Internet Referenzen

- [DGDa] *DG DemoNet – Smart LV Grid.* <http://www.ait.ac.at/departments/energy/research-areas/electric-energy-infrastructure/smart-grids/dg-demonet-smart-lv-grid/?L=1>, . – Accessed: 2013-11-02
- [DGDb] *Projektbeschreibung von DG DemoNet SmartLVGrid der Energy & IT Group.* <http://energyit.ict.tuwien.ac.at/index.php/en/projects/completed/dg-demonet>, . – Accessed: 2013-11-01
- [dig] *DIgSILENT PowerFactory.* <http://www.digsilent.de/index.php/products-powerfactory.html>, . – Accessed: 2014-05-15
- [EuZ] *20-20-20 Ziele der Europäischen Union.* <http://www.e-control.at/de/konsumenten/oeko-energie/klima-und-umwelt/20-20-20-ziele>, . – Accessed: 2014-04-08
- [MATa] *Description of MATLAB Genetic Algorithm Parameter nonlcon (Non Linear Constraint).* http://www.mathworks.de/de/help/gads/ga.html?searchHighlight=gasolver#inputarg_nonlcon, . – Accessed: 2014-06-14
- [MATb] *Description of MATLAB Genetic Algorithm with Fitness Function.* <http://www.mathworks.de/de/help/gads/examples/coding-and-minimizing-a-fitness-function-using-the-genetic-algorithm.html#zmw57dd0e1855>, . – Accessed: 2014-06-12
- [Smaa] *Smart Grid Definition von <http://www.smartgrids.at>.* <http://www.smartgrids.at/smart-grids/#smart-grids--intelligente-stromnetze>, . – Accessed: 2013-12-14
- [Smab] *Smart Grid Definition von <http://www.smartgrids.eu>.* <http://www.smartgrids.eu/node/56#12>, . – Accessed: 2014-02-21