

DIPLOMARBEIT

**Prüfung von Schutzrelais unter Verwendung
eines Real-Time-Simulators**

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines

Diplom-Ingenieurs (Dipl.-Ing.)

unter der Anleitung von

Prof. Dr.-Ing. **Wolfgang GAWLIK**

Univ. Ass. Dipl.-Ing. **Benjamin COX**, BSc

Institut für Energiesysteme und elektrische Antriebe

Eingereicht an der

Technischen Universität Wien,

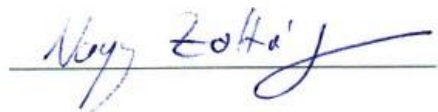
Fakultät für Elektrotechnik und Informationstechnik

von

Zoltán Simon Nagy, BSc

Matr.Nr.: 11722568

Wien, im April 2021



Danksagung

An dieser Stelle möchte ich mich gerne bei all jenen bedanken, die mich während meines Studiums und bei dem Verfassen der vorliegenden Arbeit unterstützt haben.

In erster Linie möchte ich mich bei Prof. Dr.-Ing. Wolfgang Gawlik dafür bedanken, dass er mir ermöglicht hat unter seiner Leitung diese Abschlussarbeit zu verfassen. Außerdem gilt Dank ihm, da er viele Hilfe nicht nur bei fachlichen Fragen, sondern auch bei organisatorischen Problemen geboten hat. Zweitens möchte ich mich bei Dipl.-Ing. Benjamin Cox für seine Geduld und Hilfsbereitschaft bedanken, mit denen er mich während dieses Projektes unterstützt hat. Mein Dank gilt ebenfalls Frau Dietlinde Egger, wer immer schnell und nett zur Verfügung der Studierende bei organisatorischen Fragen steht.

Mein spezieller Dank gilt meiner Familie für den festen Rückhalt in meinem ganzen Leben. Ich bedanke mich bei meinen Eltern, die mir mein Studium ermöglicht und mich immer unterstützt haben meine Träume zu verwirklichen.

Zu guter Letzt möchte ich mich bei meiner Ehefrau, die die wichtigste Person in meinem Leben ist, dafür bedanken, dass sie schon jahrelang neben mir steht und mich fest unterstützt. Wie der Prediger sagt:

*„Zwei sind besser daran als ein Einzelner,
weil sie einen guten Lohn für ihre Mühe haben.“*

[Prediger 4,9 ELB]

Kurzfassung

Die Netzführung der elektrischen Energienetze begegnet immer neueren Herausforderungen wegen der immer größeren Menge an leistungselektronischen Elementen und dezentraler Energieerzeugung. Um die sichere und unterbrechungslose Lieferung der elektrischen Energie gewährleisten zu können, ist die präzise Einstellung der Schutzeinrichtungen ein unerlässliches Minimum.

Obwohl heutzutage zahlreiche Testgeräte und Simulationsprogramme vorhanden sind, mit denen eine Schutzeinrichtung parametrisiert werden kann, fehlt die nötige Geschwindigkeit und Genauigkeit in vielen Fällen. In dieser Arbeit wird der Aufbau eines Prüfstandes vorgestellt, mit dem ein Schutzrelais durch Verwendung der Technologie von Real-Time-Simulation getestet werden kann. Das Hauptelement dieses Projektes ist der Echtzeitsimulator OP5600 von der Firma OPAL-RT, mit dem ein Schutzrelais SIPROTEC 7SL87 von Siemens geprüft wird. Diese Arbeit legt den Fokus auf die Realisierung dieses Messsystems sowohl software- als auch hardwareseitig. Das Modell, das die Basis für die Simulationen bildet, stellt das Verhalten eines Offshore-Windparks bei einem Fehlerfall dar. Dieses Modell wurde während des Projektes umgebaut, um der verwendeten Real-Time-Technologie zu entsprechen. Bei der Erstellung des Prüfstandes war ein wichtiger Eckpunkt, wie eine Großzahl von Simulationen eines Kurzschlussfalles mit unterschiedlichen Parametern durchgeführt werden kann.

Das System wurde für einpoligen Erdschluss und zweipoligen Kurzschluss getestet. Um die Authentizität der Messergebnisse mit OP5600 überprüfen und einen möglichen Messfehler wegen der ungenauen Relaisereinstellungen ausschließen zu können, wurde das System auch mit dem Prüfgerät CMC 356 von Firma Omicron getestet. Schließlich wurden die Ergebnisse beider Methoden verglichen.

Abstract

The grid control of the electrical energy networks is facing more and more new challenges due to the increasing amount of power electronic elements and decentralized energy generation. In order to be able to guarantee the safe and uninterrupted supply of electrical energy, the precise setting of the protective devices is an essential minimum.

Although nowadays there are numerous test devices and simulation programs with which a protective device can be parameterized, the necessary speed and accuracy are lacking in many cases. In this thesis the construction of a test stand is presented, with which a protective relay can be tested by using the technology of real-time simulation. The main element of this project is the real-time simulator OP5600 from OPAL-RT, with which a protection relay SIPROTEC 7SL87 from Siemens is tested. This thesis focuses on the implementation of this measurement system on the software as well as the hardware side. The model, which forms the basis for the simulations, shows the behaviour of an offshore wind farm in the event of a fault. This model was modified during the project to correspond to the real-time technology used. When creating the test stand, an important cornerstone was how a large number of simulations of a short circuit case can be carried out with different parameters.

The system has been tested for single-pole earth fault and double-pole short circuit. In order to check the authenticity of the measurement results with the OP5600 and to be able to rule out a possible measurement error due to the imprecise relay settings, the system was also tested with the CMC 356 test device from Omicron. Finally, the results of both methods were compared.

Inhaltsverzeichnis

1	Einführung.....	1
1.1	Ausgangspunkt des Projektes	1
2	Theorie	4
2.1	Kurz- und Erdschlüsse im elektrischen Energienetz	4
2.1.1	Grundlagen	4
2.1.2	Einpoliger Kurzschluss	9
2.1.3	Zweipoliger Kurzschluss.....	12
2.2	Netzschutz	14
2.2.1	Grundlagen	14
2.2.2	Typen der Schutzeinrichtungen.....	14
2.2.3	Distanzschutz	15
3	Methodik	20
3.1	Aufbau des Messsystems	20
3.2	Projektphasen.....	21
4	Verwendete Programme und Geräte	22
4.1	MATLAB und Simulink.....	22
4.1.1	SimPowerSystems	23
4.2	OPAL-RT und RT-LAB	23
4.2.1	RT-LAB	24
4.2.2	OP5600 Real-Time-Simulator.....	25
4.3	Doble F6350e	28
4.4	Siemens SIPROTEC 7SL87	29
4.5	Omicron CMC 356	29
5	Messaufbau.....	31
5.1	Umstellung des Modelles	31
5.2	Aufbau des Messsystems.....	33
5.3	Lösung der Simulationsautomatisierung	35
5.3.1	Lichtbogenwiderstad	35
5.3.2	Phasenwinkel.....	37

5.3.3	Fehlerort	40
5.3.4	Reaktion des Schutzrelais.....	42
5.3.5	Automatisierung mit Python-Skript	44
6	Ergebnisse	47
6.1	Test des Schutzrelais	47
6.2	Testen mit Omicron CMC 356	50
6.2.1	Zweipoliger Kurzschluss.....	51
6.2.2	Einpoliger Fehler.....	55
7	Ausblick	59
7.1	Verwendung mehrerer Prozessorkerne.....	60
7.2	ARTEMiS.....	60
8	Zusammenfassung.....	63
	Literaturverzeichnis.....	65
	Abbildungsverzeichnis	67
	Tabellenverzeichnis.....	69
	Abkürzungsverzeichnis	69
Anhang A.:	Schaltplan der Testeinrichtung von SIPROTEC 7SL87.....	70
Anhang B.:	Subsystem SM_MASTER V.1.....	71
Anhang C.:	Kommunikationskanälen	72
Anhang D.:	Winkelmessungsproblem.....	73
Anhang E.:	Subsystem SC_GUI V.2	74
Anhang F.:	Subsystem SM_MASTER V2	75
Anhang G.:	Python-Skript V.1.....	76
Anhang H.:	Python-Skript V.2.....	78

1 Einführung

Wegen der immer größeren Menge an leistungselektronischen Elementen und dezentraler Energieerzeugung bedeutet die Netzführung heutzutage immer größere Herausforderung. Die Netzmodellierung und Simulation sind zwei unerlässliche Komponenten der Energietechnik. Die richtige Einstellung von Schutzgeräten ist heute unvorstellbar ohne diese Technologie. Die präzise Darstellung des modellierten Netzes ist nicht mehr die einzige Anforderung, die diese Modelle und Simulationen erfüllen müssen. Die Simulationen sollen ebenfalls schnell und genau wiederholbar sein.

Das Ziel dieser Arbeit ist die Vorstellung des Aufbaues eines Prüfstandes, mit dem ein Schutzrelais durch Verwendung der Technologie von Real-Time-Simulation getestet werden kann. Das Hauptelement dieses Projektes ist der Echtzeitsimulator OP5600 von der Firma OPAL-RT, mit dem ein Schutzrelais SIPROTEC 7SL87 von Siemens geprüft wird. Alle während des Projektes verwendeten Geräte und Programme werden im Kapitel 4 betrachtet.

Diese Arbeit legt den Fokus auf die Realisierung dieses Messsystems sowohl software- als auch hardwareseitig. Die Erstellungsphase des Messsystems wird in Kapitel 5 beschrieben. Hier ist es ein wichtiger Eckpunkt, wie eine Großzahl von Simulationen eines Kurzschlussfalles mit unterschiedlichen Parametern durchgeführt werden kann.

1.1 Ausgangspunkt des Projektes

Das Modell, das während des Projektes als die Basis der Simulationen gilt, wird in der Abbildung 1.1 dargestellt. Dieses Modell wurde an der Technischen Universität Wien an dem Institut für Energiesysteme und elektrische Antriebe eingestellt, um das folgende Phänomen untersuchen zu können.

In einem norddeutschen Windpark ist das Problem aufgetaucht, als ein einpoliger Fehler in einem Zweig vorkam, dass der Distanzschutz auf dem Nachbarzweig schneller ausgelöst hat als auf der fehlerbehafteten Leitung. Also in diesem Fall konnte die selektive Abschaltung der Netzteile nicht erfüllt werden. Das Ziel bei der Erstellung dieses Modelles war der Erhalt möglichst gleicher Messdaten während der Simulation als in dem realen System.

Das Modell besteht aus den folgenden Elementen: Die Komponenten im obigen Bereich der Abbildung sind für die Darstellung und Normierung der Messdaten verantwortlich. Sie sind für das Projekt nicht relevant und deswegen werden sie in späteren Versionen weggelassen. Die Windkraftanlagen werden als konzentrierte Einspeisungen am Ende der sechs Zweige (unten) dargestellt, die durch drei phasenrichtige Stromquellen modelliert werden. Die Leitungen von den Einspeisungen bis zur Sammelschiene haben unterschiedliche Längen und Parameter. Zwischen der Sammelschiene der Abzweigungen und dem Drehstromtransformator liegt noch ein kurzes 0,1 km langes Leitungsstück. Die Schaltgruppe des Transformators ist Ynd mit den Spannungsniveaus 155 kV/ 34 kV. Hochspannungsseitig ist er mit einer idealen HGÜ-Einspeisung verbunden. Mittelspannungsseitig gewährleistet ein Sternpunktbildner das Nullpotential. Der Sternpunkt ist mit 25Ohm geerdet (RA001).

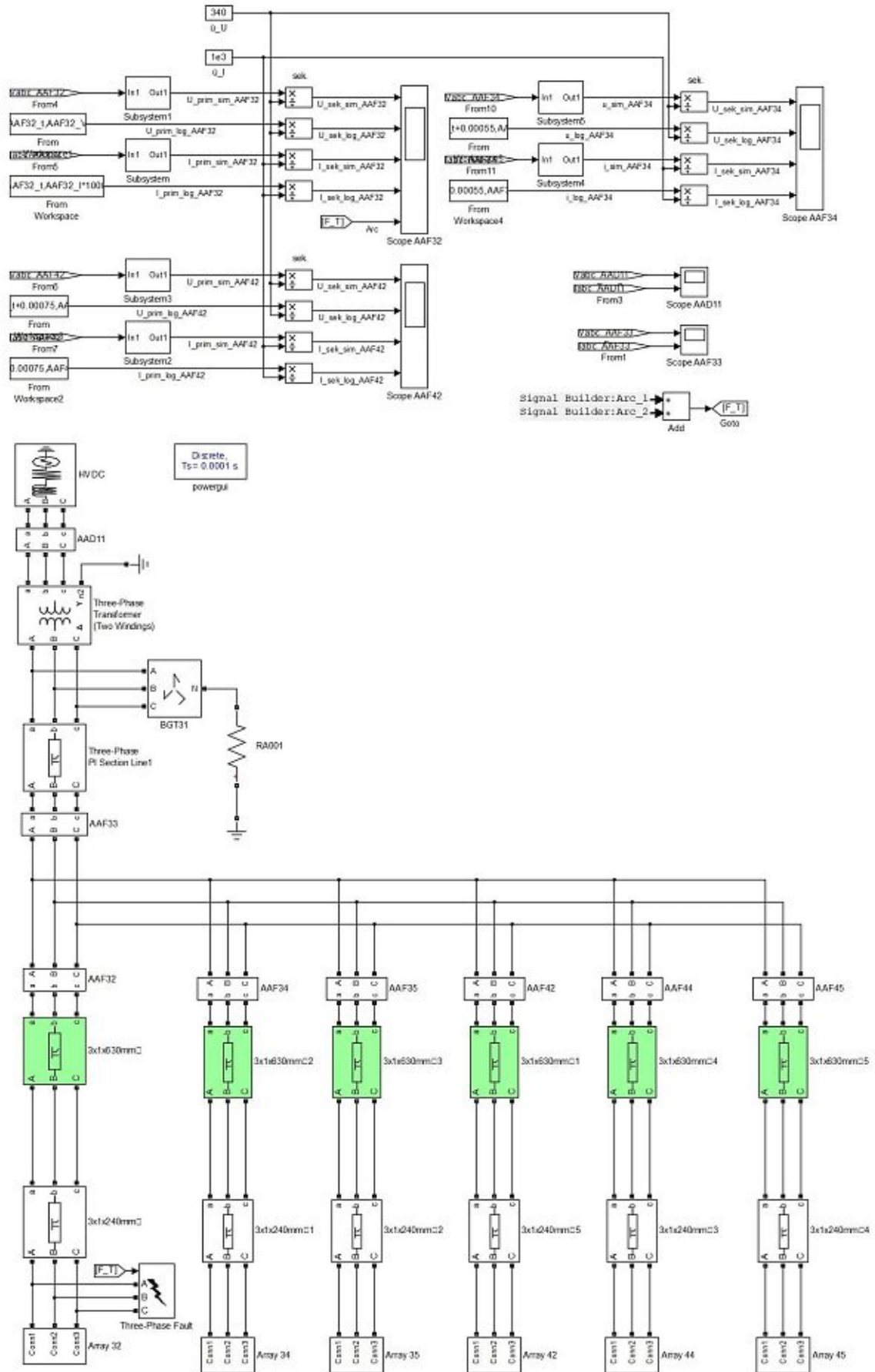


Abbildung 1.1: Original Modell des Windparks

Auf den Abzweigungen befinden sich mehrere Messstationen (AAFxx), die grundsätzlich als Messpunkte des Distanzschutzes dienen. Ein Three-Phase Fault Block wird am Ende der ersten Leitung (AAF32) hinzugefügt, damit der einpolige Erdschluss simuliert wird. Der Selektivitätsproblem entsteht also zwischen den Distanzschutzeinrichtungen AAF32 und AAF34.

Nichtdestotrotz ist die Zielsetzung dieses Projektes nicht die Untersuchung des oben erwähnten Phänomens, sondern die Erstellung eines Mess- und Simulationssystems, mit dem dieses oder ein ähnliches Problem erforscht und gelöst werden kann.

2 Theorie

In diesem Kapitel werden die wichtigsten, mit diesem Projekt verbundenen theoretischen und technischen Hintergründe zusammengefasst.

2.1 Kurz- und Erdschlüsse im elektrischen Energienetz

Dieses Kapitel gibt einen Überblick über den Themenbereich Kurz- und Erdschlüsse. Es beschäftigt sich detaillierter mit der Beschreibung des ein- und zweipoligen Fehlers, da das aufgebaute Messsystem durch diese zwei Fehlerarten getestet wird.

2.1.1 Grundlagen

Grundsätzlich können zwei Fehlerarten im elektrischen Netz unterschieden werden, nämlich Querfehler (also Kurz- und Erdschluss) oder Längsfehler (also Phasenunterbrechung). Diese Fehler können ein-, zwei oder dreipolig mit oder ohne Erdberührung erfolgen. Sie können in einem Netz nicht nur als Einfachfehler sondern auch als Mehrfachfehler (gleichzeitig mehrere Fehler in unterschiedlichen Orten) entstehen und die Kombination dieser Fehlerarten ist auch vorstellbar (z.B.: Einpoliger Erdschluss mit Kabelunterbrechung). Als Ursachen dieser Vorgänge können unter anderem atmosphärische Einwirkungen (Blitz, Sturm, Eis), Schaltüberspannungen, Alterung der Isolation, fremde Einwirkungen (Baumfalle, Bauarbeiten) oder Fehlbedienung benannt werden. [1] [2]

Querfehler in einem System können Kurzschluss (zwischen zwei Phasen), Erdschluss oder Erdkurzschluss (zwischen Erde und Phase(n)) sein. Ob man von einem Erdschluss oder einem Erdkurzschluss spricht, hängt davon ab, wie groß der Kurzschlussstrom während des Fehlervorganges ist. Wenn es kein Sternpunktterdung im System gibt, kann kein Nullsystem im Sternpunkt der Einspeisung gebildet werden. In diesem Fall fließt nur geringer Fehlerstrom wegen der Leitungskapazität, es nennt man Erdschluss. Wenn das Netz starr oder niederohmig geerdet ist, fließt ein deutlich höher Fehlerstrom. In diesem Fall spricht man von Erdkurzschluss. [1] [2]

Wenn man den zeitlichen Verlauf eines Kurzschlussstromes beobachtet, können die folgenden Eigenschaften bemerkt werden. Der Kurzschlussstromzeitverlauf entsteht durch einen abklingenden Wechselstromanteil und einen abklingenden Gleichstromanteil. Der Grund dieses Zeitverlaufes sind die Ausgleichvorgänge der Generatoren, die ins Netz einspeisen. Deswegen ist die Entfernung des Kurzschlussortes zum Generator ein wichtiger Einflussfaktor. Der Ablauf der Ausgleichvorgängen hängt von den Eigenschaften des Generators wie transiente T' und subtransiente T'' Zeitkonstante ab. Eine allgemeine Beschreibung des Kurzschlussstromzeitverlaufes ergibt sich nach [2]:

$$i_k(t) = \underbrace{\sqrt{2} \left[(I_k'' - I_k') e^{-\frac{t}{T_d''}} + (I_k' - I_k) e^{-\frac{t}{T_d'}} + I_k \right]}_{\text{Wechselstromanteil}} \cos(\omega t + \alpha) - \underbrace{\sqrt{2} I_k'' \cos \alpha \cdot e^{-\frac{t}{T_g}}}_{\text{Gleichstromanteil}} \quad (2.1)$$

Wo:

- I_k'' – Anfangskurzschlusswechselstrom
- I_k' – transienter Kurzschlussstrom
- I_k – Dauerkurzschlussstrom
- T_d'' – subtransiente Zeitkonstante
- T_d' – transiente Zeitkonstante
- T_g – Gleichstromzeitkonstante
- α – Nullphasenwinkel des Kurzschlussstromes $i_k(t)$

Wenn der Kurzschluss nicht unmittelbar an den Klemmen der Statorwicklung auftaucht, wird er ebenfalls durch die Impedanz der elektrischen Komponenten des Netzes wie Leitungen und Transformatoren beeinflusst. Deswegen ergibt sich die transiente und subtransiente Induktivität aus der Induktivität des Generators und des Netzes nach [3]:

$$l'' = l_g'' + l_{Netz} \quad l' = l_g' + l_{Netz} \quad l = l_g + l_{Netz} \quad (2.2)$$

Folglich wird die Differenz zwischen dem transienten, subtransienten und Dauerkurzschlussstrom mit Erhöhung der Entfernung von Generator immer kleiner sein. Deshalb kann zwischen generatornahen und generatorfernen Kurzschlüssen unterschieden werden. Der Hauptunterschied zwischen beiden Fällen ist, dass der Wechselstromanteil bei einem generatornahen Kurzschluss mit der Zeit abklingt ($I_k'' > I_k$). Im Fall von einem generatorfernen Kurzschluss bleibt dieser Anteil entlang den ganzen Kurzschlusszeitvorgang fast gleich ($I_k'' \approx I_k$).

Mit diesen Bedingungen kann ein Kurzschlussfall auf folgende Weise modelliert werden (siehe Abbildung 2.1):

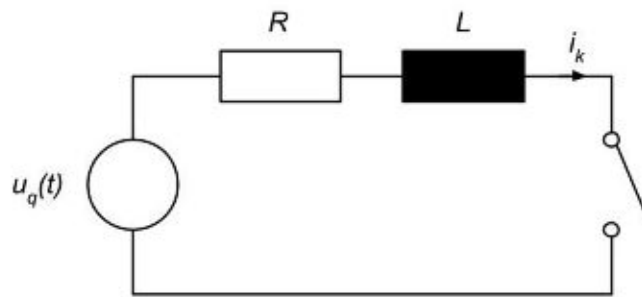


Abbildung 2.1: Modell eines Kurzschlussvorganges [2]

Nach der Abbildung 2.1 kann die vereinfachte Formel des Verlaufes des Kurzschlussstromes aufgeschrieben werden [2]:

$$i_k(t) = \sqrt{2} \frac{U_q}{Z} \cos(\omega t + \varphi_u - \varphi_Z) + k e^{-\frac{t}{T}} \quad (2.3)$$

mit:

$$Z = \sqrt{R^2 + (\omega L)^2} \quad \varphi_Z = \arctan\left(\frac{\omega L}{R}\right) \quad T = \frac{L}{R} \quad (2.4)$$

und

- φ_u – Nullphasenwinkel der Spannung am Kurzschlussort
- φ_Z – Winkel der Kurzschlussimpedanz an der Kurzschlussstelle
- k – Stossfaktor

Die Konstante k kann über die Anfangsbedingungen bestimmt werden. Da der Strom sich an einer Induktivität nicht sprunghaft, sondern nur stetig ändern kann, nimmt der Fehlerstrom den gleichen Betrag nach dem Kurzschluss wie davor auf. Dieser Wert ist gleich null, deshalb ergibt sich der Konstante k nach [2]:

$$k = -\sqrt{2} \frac{U_q}{Z} \cos(\varphi_u - \varphi_Z) \quad (2.5)$$

So ist der zeitliche Verlauf des Kurzschlussstromes:

$$i_k(t) = \sqrt{2} \frac{U_q}{Z} \left[\cos(\omega t + \varphi_u - \varphi_Z) - \cos(\varphi_u - \varphi_Z) e^{-\frac{t}{T}} \right] \quad (2.6)$$

Der Stromverlauf bei einem generatornahen Kurzschluss wird in der Abbildung 2.2 sowie bei einem generatorfernen Kurzschluss in der Abbildung 2.3 dargestellt. Solange der Gleichstromanteil nicht abklingt, besteht der Anfangskurzschlussstrom aus der Summe beider Komponenten. Wenn der Stossfaktor gleich null ist, entsteht kein Gleichstromanteil. Der Konstant k und deswegen das Maximum $I_{g \max}$ des Gleichstromgliedes hängt von der Phasenlage des Stromes im Zeitpunkt des Kurzschlusses ab [1] [4]:

- $\alpha - \varphi_Z = 0$: Kein Gleichstromglied ist vorhanden. Der Kurzschlussstrom hat nur Wechselkomponente.
- $\alpha - \varphi_Z = 90^\circ$: Die größtmögliche Gleichstromkomponente entsteht.

Hier ist es wichtig zu bemerken, dass kein Gleichstromglied bei dem simulierten Kurzschluss in diesem Projekt entsteht, da die Einspeisung in dem Modell einerseits durch einen idealen HGÜ-Netzanschluss, andererseits durch drei ideale AC-Stromquellen erfolgt. Deswegen kann diese Erscheinung im Verlauf der simulierten Kurzschlussströme nicht erkannt werden.

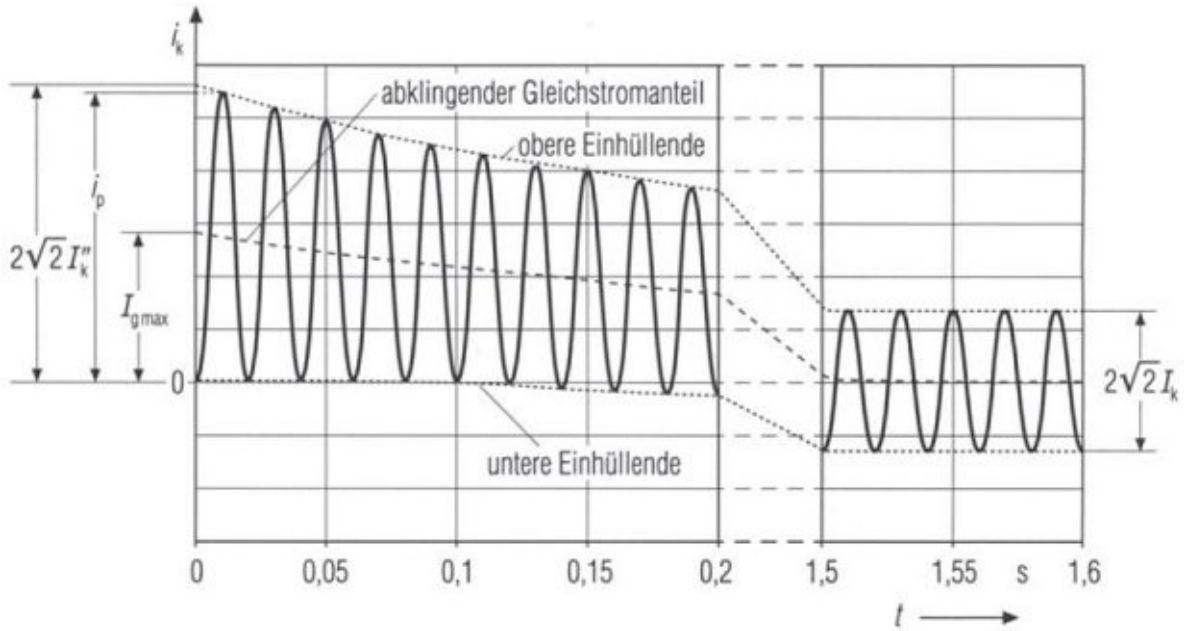


Abbildung 2.2: Stromverlauf bei einem generatornahen Kurzschluss [4]

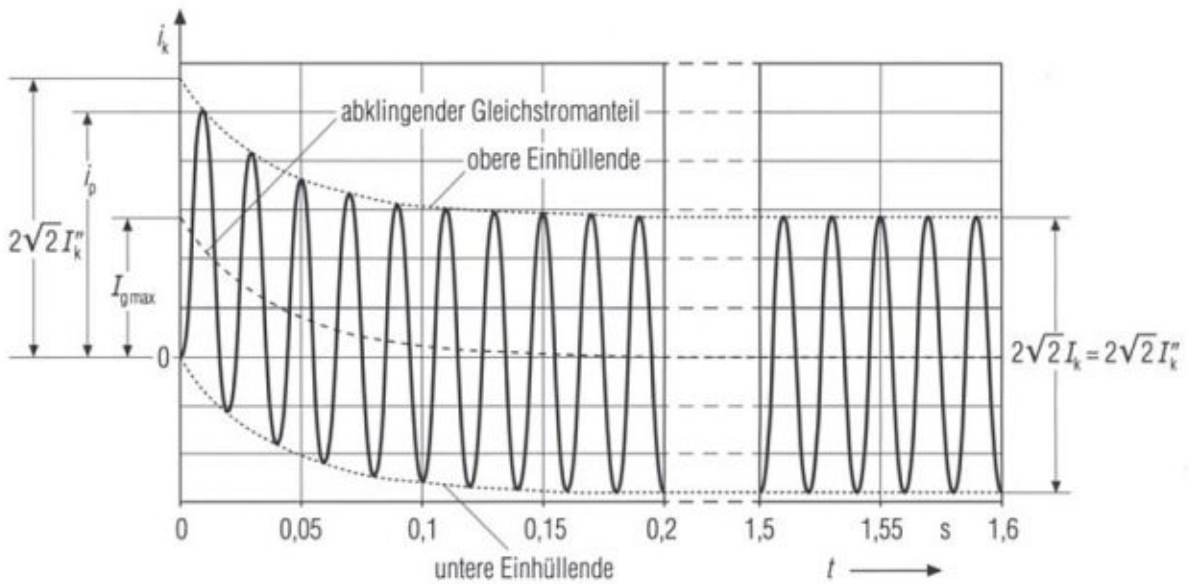


Abbildung 2.3: Stromverlauf bei einem generatorfernen Kurzschluss [4]

Die zur Bestimmung der Kurzschlussstromzeitverläufe verwendeten Kenngrößen sind die folgenden [1] [2]:

- **Anfangskurzschlusswechselstrom I_k''**

Er ist eine fiktive Rechengröße, die praktisch nicht messbar ist. Sein Betrag ist der Effektivwert des Wechselstromanteils im Augenblick des Kurzschlusseintritts. Es kann mit Hilfe der in der Abbildung 2.2 oder Abbildung 2.3 eingezeichneten Hüllkurven des Kurzschlussstromverlaufs bestimmt werden. Der zum Zeitpunkt $t = 0$ gehörige Betrag der oberen Hüllkurve ist gleich mit der doppelten Amplitude des Wechselstromanteils, d.h. $2\sqrt{2}I_k''$.

- **Stoßkurzschlussstrom i_p**

Er stellt den größtmöglichen Momentanwert des Kurzschlussstromes dar. Dieser Wert ist für die Überprüfung der mechanischen Kurzschlussfestigkeit der Netzanteile relevant. Der Stoßkurzschlussstrom kann mit Hilfe des Faktors k bestimmt werden:

$$i_p = \sqrt{2}kI_k'' \quad (2.7)$$

mit:

$$k = 1,02 + 0,98e^{-3R/X} \quad (2.8)$$

- **Ausschaltwechselstrom I_b**

Dieser Wert kann für die Dimensionierung der Leistungsschalter verwendet werden. Er stellt den Effektivwert des Kurzschlussstromes unmittelbar vor der Abschaltung dar. Bei den heutigen Hochleistungsschaltern hat er immer weniger Bedeutung, da diese Geräte auch den großen Anfangskurzschlusswechselstrom unterbrechen können.

- **Dauerkurzschlussstrom I_k**

Er repräsentiert den Effektivwert des Kurzschlussstromes nach dem Abklingen aller Ausgleichsvorgänge. Er hat in Praxis nur geringe Bedeutung, da die Kurzschlüsse im Allgemeinen vor dem Erreichen dieses Wertes unterbrochen werden. Zwei Typen des Dauerkurzschlussstromes können aber unterschieden werden:

- **Größter Dauerkurzschlussstrom I_{kmax}**

Er wird bei der Dimensionierung von Sicherungen und Schutzgeräten mit Überstromauslösung genutzt, um eine thermische Überlastung der Betriebsmittel bei langen Kurzschlussdauern zu vermeiden.

- **Kleinster Dauerkurzschlussstrom I_{kmin}**

Er wird bei Einstellung der Empfindlichkeit von Schutzeinrichtungen verwendet, um eine sichere Abschaltung von Fehlern zu ermöglichen.

- **Thermisch gleichwertiger Kurzschlussstrom I_{th}**

Er ist für die Beurteilung der thermischen Kurzschlussfestigkeit von elektrischen Betriebsmitteln verantwortlich. Er stellt den Effektivwert des Kurzschlussstromes dar, der während der Kurzschlussdauer mit dem abklingenden Kurzschlussstrom die gleiche Wärmewirkung hat. Er ergibt sich nach [2]:

$$I_{th} = \sqrt{m + n} \cdot I_k'' \quad (2.9)$$

wo die Faktoren m und n die Wärmeeffekte der zeitlich abklingenden Wechsel- und Gleichstromanteile berücksichtigen.

2.1.2 Einpoliger Kurzschluss

In den Hoch- und Höchstspannungsnetzen ist die am häufigsten auftretende Fehlerart der einpolige Querfehler (ca. 80%), wovon ein ca. 70-80-prozentiger Anteil in Netzen mit isoliertem Sternpunkt oder mit Resonanzpunkterdung als Erdschlüsse vorkommt. [2]

Starre Erdung

Zur Bestimmung des Kurzschlussstromes bei einpoliger Erdberührung ist es sinnvoll ihn erstens in einem starr geerdeten Netz zu betrachten. Als weitere Vereinfachung wird die Fehlerimpedanz (sowie Lichtbogenwiderstand) vernachlässigt. In der Abbildung 2.4 wird dieser Fehlerfall in einem vereinfachten Ersatzschaltbild einer Drehstromquelle mit Leitung sowie in symmetrischen Komponenten dargestellt.

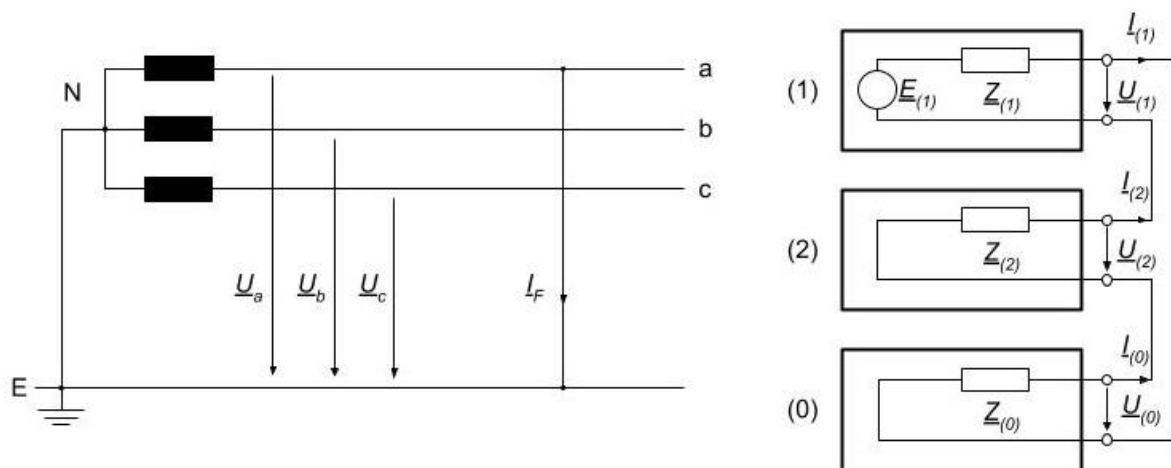


Abbildung 2.4: Einpoliger Kurzschluss im starr geerdeten Netz [1]

Die Fehlerbedingungen sind nach [1] die folgenden:

$$\begin{aligned} \underline{I}_{b,F} &= 0 \\ \underline{I}_{c,F} &= 0 \\ \underline{U}_{a,F} &= 0 \end{aligned} \quad \leftrightarrow \quad \begin{aligned} \underline{I}_{(0)} &= \underline{I}_{(1)} = \underline{I}_{(2)} \\ \underline{U}_{(0)} + \underline{U}_{(1)} + \underline{U}_{(2)} &= 0 \end{aligned} \quad (2.10)$$

Der Fehlerstrom \underline{I}_F fließt nur in Phase a , in Phase b und c kann kein Fehlerstrom fließen. Da die Leitung von Phase a unmittelbar mit der Erde kurzgeschlossen wird, ist die Spannung $\underline{U}_{a,F}$ ebenfalls gleich null. Wie es in der Abbildung 2.4 sichtbar ist, sind die Impedanzen des Mit-, Gegen- und Nullsystems in Reihe geschaltet, deswegen sind ihre Ströme identisch [1]:

$$\underline{I}_{(0)} = \underline{I}_{(1)} = \underline{I}_{(2)} = \frac{\underline{E}_{(1)}}{\underline{Z}_{(0)} + \underline{Z}_{(1)} + \underline{Z}_{(2)}} \quad (2.11)$$

Mit der Rücktransformation der Komponentenströme in Originalbereich erhält man den Erdkurzschlussstrom, der weiterhin in Phase a fließt [1]:

$$\underline{I}_a = \underline{I}_{(0)} = \underline{I}_{(1)} = \underline{I}_{(2)} = \frac{\underline{E}_{(1)}}{\underline{Z}_{(0)} + \underline{Z}_{(1)} + \underline{Z}_{(2)}} \quad (2.12)$$

Da die Mit- und Gegenimpedanzen im passiven Netz im Allgemeinen gleich sind, können sie als $2\underline{Z}_{(1)}$ vereinfacht werden. Der Kurzschlussstrom kann also mit Nennspannung (verkettete Spannung) in folgender Weise ausgedrückt werden [1]:

$$I''_{k1p} = \frac{\sqrt{3}cU_n}{|\underline{Z}_{(0)} + 2\underline{Z}_{(1)}|} \quad (2.13)$$

wo c der Spannungsfaktor (zwischen 0,95...1,10) ist.

Die Spannungen in den nicht fehlerbehafteten Phasen ergeben sich nach [1]:

$$\underline{U}_b = j\sqrt{3}\underline{E}_{(1)} \frac{\underline{a} \cdot \underline{Z}_{(0)} - \underline{Z}_{(2)}}{\underline{Z}_{(0)} + \underline{Z}_{(1)} + \underline{Z}_{(2)}} \quad (2.14)$$

$$\underline{U}_c = j\sqrt{3}\underline{E}_{(1)} \frac{\underline{Z}_{(2)} - \underline{a}^2 \cdot \underline{Z}_{(0)}}{\underline{Z}_{(0)} + \underline{Z}_{(1)} + \underline{Z}_{(2)}} \quad (2.15)$$

wo:

$$\underline{a} = e^{\frac{j2\pi}{3}} = -\frac{1}{2} + j\frac{\sqrt{3}}{2} \quad (2.16)$$

$$\underline{a}^2 = \underline{a}^* = e^{j4\pi/3} = -\frac{1}{2} - j\frac{\sqrt{3}}{2} \quad (2.17)$$

Isolierter Sternpunkt

Eine ganz andere Situation entsteht im isolierten Netz also, wenn keine Sternpunkte geerdet sind. In diesem Fall kann kein Fehlerstrom über den Sternpunkt in die Erde fließen. Das bedeutet, dass kein Nullsystem gebildet werden kann. In die Darstellung der symmetrischen Komponenten wird es mit einem geöffneten Schalter im Nullsystem (siehe Abbildung 2.5) symbolisiert. [5]

Da kein Fehlerstrom durch den Sternpunkt fließen kann, kann der Stromkreis nur durch die Kapazität der Leitung geschlossen werden. In diesem Fall müssen also die Querleitwerte der Leitung ebenfalls berücksichtigt werden, obwohl sie bisher grundsätzlich kurzgeschlossen waren und deswegen vernachlässigt werden konnten. [1]

Die Nullkapazität ergibt sich aus der Summe der Erdkapazitäten aller Leitungen und Kabel im betrachteten Netz. Da die Querimpedanz – die durch die Erdkapazitäten gebildet wird –, deutlich größer als die Längsimpedanzen ist, kann man sie vernachlässigen und den Fehlerstrom einfach durch die Querimpedanz bestimmen [1]:

$$\underline{Z}_{(0)} + \underline{Z}_{(1)} + \underline{Z}_{(2)} \ll \frac{1}{j\omega\underline{C}_{(0)}} \quad (2.18)$$

$$\underline{I}_a = \underline{I}_{(0)} = \underline{I}_{(1)} = \underline{I}_{(2)} \approx \frac{\underline{E}_{(1)}}{\frac{1}{j\omega\underline{C}_{(0)}}} = j\omega\underline{C}_{(0)}\underline{E}_{(1)} \quad (2.19)$$

$$\underline{I}_{k1p}'' = \underline{I}_a = j\omega\underline{C}_{(0)}\underline{E}_{(1)} = j\omega\sqrt{3}\underline{C}_{(0)} \cdot cU_n \quad (2.20)$$

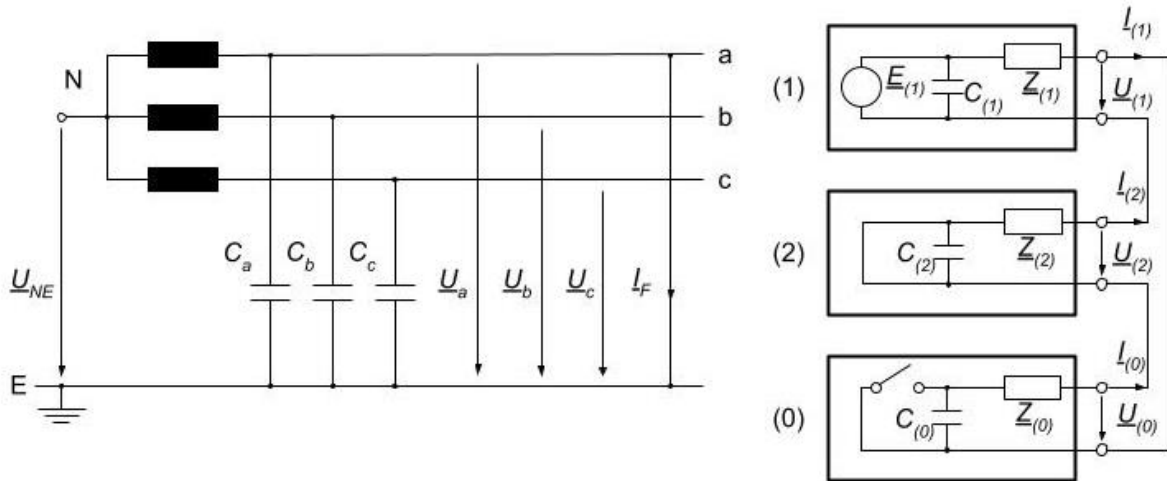


Abbildung 2.5: Einpoliger Kurzschluss im isolierten Netz [1]

Da die Fehlerimpedanz durch die Erdkapazität bestimmt wird und in diesem Fall deutlich größer als bei starr geerdetem Netz ist, wird der Kurzschlussstrom kleiner. Es ist aber nicht unbedingt vorteilhaft. Aus der Sicht von Schutztechnik kann ein kleiner Fehlerstrom ungünstig sein, weil er schwieriger detektiert werden kann, da die Stromaufnahme dieses Fehlerfalles ganz nahe dem normalen Betriebsfall liegt. Ein weiteres Merkmal dieses Fehlerfalles ist die Spannungsverlagerung. Die Phasenspannungen können in folgender Weise bestimmt werden [1]:

$$\underline{U}_{(0)} \approx -\underline{E}_{(1)} \quad \underline{U}_{(1)} \approx \underline{E}_{(1)} \quad \underline{U}_{(2)} \approx 0 \quad (2.21)$$

$$\underline{U}_b = (\underline{a}^2 - 1) \cdot \underline{E}_{(1)} \quad (2.22)$$

$$\underline{U}_c = (\underline{a} - 1) \cdot \underline{E}_{(1)} \quad (2.23)$$

Wo:

$$|\underline{a}^2 - 1| = |\underline{a} - 1| = \sqrt{3} \quad (2.24)$$

Wie es aus Formel (2.21) - (2.24) sichtbar ist, erhöht sich die Spannung in den gesunden Phasen um $\sqrt{3}$, d.h. die Phasenspannungen werden so groß sein wie die verketteten Spannungen. So führt dieser Erdschluss zu unzulässigen Überspannungen.

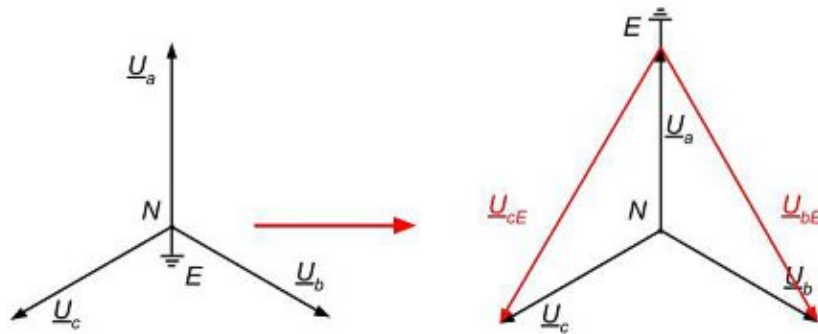


Abbildung 2.6: Spannungsverlagerung bei einpoligem Erdschluss [1]

In der Abbildung 2.6 werden die Spannungszeiger eines symmetrischen dreiphasigen Systems dargestellt. An der linken Seite sind die Zeiger unter normalem Zustand sichtbar. An der rechten Seite wird die Spannungsverlagerung wegen eines einpoligen Kurzschlusses im isolierten Netz dargestellt.

Es gibt eine weitere Ursache, weswegen dieser Fehlerart unerwünscht ist. Der Fehlerstrom ist stark kapazitiv, d.h. er fließt gegenüber der Spannung an der Fehlerstelle um 90° voreilend. Es erschwert das Löschen des Lichtbogens, da die Spannung beim Stromnulldurchgang den Maximalwert hat. Deswegen darf der Kurzschlussstrom 30 A nicht überschreiten, was die Ausdehnung des Netzes mit isoliertem Sternpunkt begrenzt. [1]

Wenn ein Netz hochohmig geerdet wird, verhält das System bei einem einpoligen Erdschluss fast genauso wie mit isoliertem Sternpunkt.

2.1.3 Zweipoliger Kurzschluss

Die zweite Fehlerart, die in diesem Projekt vorgekommen ist, ist der zweipolige Kurzschluss ohne Erdberührung. Diese Fehlerart ist ideal für die Untersuchung und das Testen der Simulationen und Messsysteme, da die Nullimpedanz in diesem Fall gegenüber dem einpoligen Erdschluss den Fehlerstrom nicht beeinflusst. Deswegen kann die Fehlerimpedanz und der Fehlerabstand einfacher bestimmt werden. Außerdem ist der Fehlerstrom deutlich größer, was das Detektieren des Fehlerfalles erleichtert. Die Fehlerbedingungen des zweipoligen Kurzschlusses sind folgenden nach [1]:

$$\begin{aligned}
 \underline{I}_{a,F} &= 0 & \underline{I}_{(0)} &= 0 \\
 \underline{I}_{b,F} + \underline{I}_{c,F} &= 0 & \Leftrightarrow & \underline{I}_{(2)} = -\underline{I}_{(1)} \\
 \underline{U}_{b,F} &= \underline{U}_{c,F} & & \underline{U}_{(2)} = \underline{U}_{(1)}
 \end{aligned}
 \tag{2.25}$$

Die Querimpedanzen können in diesem Fall vernachlässigt werden. Der Mitsystemstrom, der gleich mit dem negativen Gegensystemstrom ist, kann einfach durch die Impedanzen des Mit- und Gegensystems bestimmt werden, wie es in Abbildung 2.7 ebenfalls sichtbar ist.

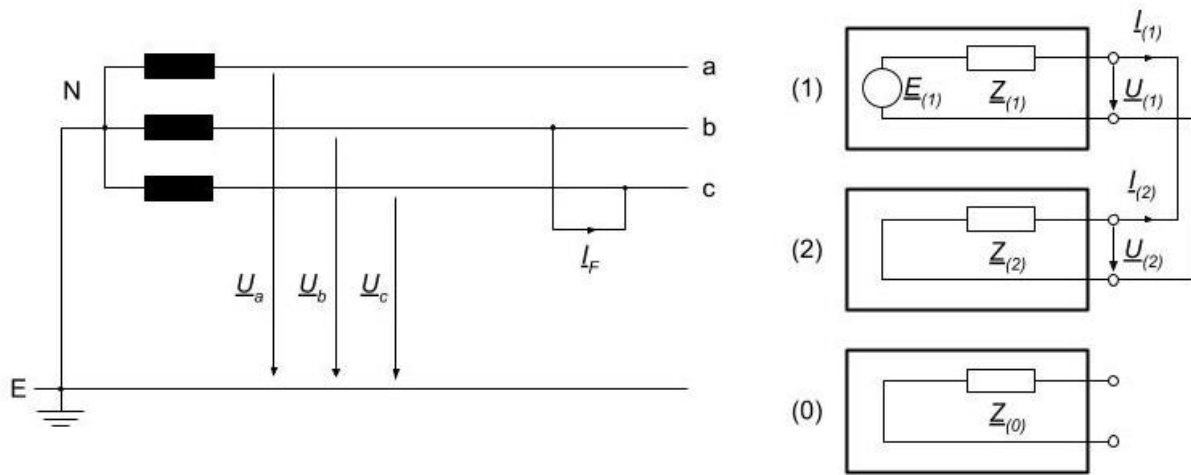


Abbildung 2.7: Zweipoliger Kurzschluss ohne Erdberührung [1]

Der Strom des Mit- und Gegensystems ist nach [1]:

$$\underline{I}_{(1)} = -\underline{I}_{(2)} = \frac{\underline{E}_{(1)}}{\underline{Z}_{(1)} + \underline{Z}_{(2)}} = \frac{c \cdot U_n}{\sqrt{3}(\underline{Z}_{(1)} + \underline{Z}_{(2)})} \quad (2.26)$$

Nach der Rücktransformation in Phasengrößen ergibt sich der Kurzschlussstrom:

$$\underline{I}_b = -\underline{I}_c = -j\sqrt{3}\underline{I}_{(1)} = -j\frac{c \cdot U_n}{\underline{Z}_{(1)} + \underline{Z}_{(2)}} \quad (2.27)$$

Da die Mit- und Gegenimpedanzen im passiven Netz im Allgemeinen gleich oder fast gleich genommen sind, kann die Formel bei Bestimmung des Betrages des Kurzschlussstromes vereinfacht werden:

$$I_{k2p}'' = \frac{c \cdot U_n}{|\underline{Z}_{(1)} + \underline{Z}_{(2)}|} = \frac{c \cdot U_n}{2 \cdot |\underline{Z}_{(1)}|} \quad (2.28)$$

Der Kurzschlussstrom im Fall eines zweipoligen Kurzschlusses ohne Erdberührung ist immer kleiner als im Fall eines dreipoligen Fehlers. Der Zusammenhang zwischen den beiden Größen ist:

$$\frac{I_{k2p}''}{I_{k3p}''} = \frac{\sqrt{3}}{2} = 0,866 \quad (2.29)$$

2.2 Netzschutz

2.2.1 Grundlagen

Ein Netz muss vor unzulässigen Betriebszuständen geschützt werden. Nach dem Typ der Störung wird zwischen Überlastschutz, Kurzschlusschutz, Erdschlusschutz, Über- und Unterspannungsschutz sowie Über- und Unterfrequenzschutz unterschieden. Die Schutzeinrichtungen, die diese Aufgaben erfüllen, können nach Aufbau und Schutzverfahren differenziert werden. Sie müssen jedoch solche Anforderungen bestmöglich erfüllen, wie

- keine Überfunktion,
- keine Unterfunktion,
- Selektivität,
- Schnelligkeit,
- Genauigkeit,
- Selbstständigkeit,
- wirtschaftliche Angemessenheit,

um einen sicheren und wirtschaftlichen Netzbetrieb gewährleisten zu können. Wie es im Kapitel 1 sichtbar ist, erfüllt sich die Selektivität als Anforderung im untersuchten System nicht, da ein fehlerfreier Zweig auch abgeschaltet wird. [6]

2.2.2 Typen der Schutzeinrichtungen

Es gibt zwei Hauptgruppen der Schutzeinrichtungen, nämlich Primär- und Sekundärschutzeinrichtungen. Primärschutzeinrichtungen befinden sich im Allgemeinen im Niederspannungsnetz. Sie arbeiten unmittelbar auf dem Potential des schützenden Netzes. Im Mittel-, Hoch-, und Höchstspannungsnetz müssen Schutz- und Schaltfunktionen wegen der großen Betriebsspannungen und -strömen voneinander getrennt werden. In diesem Fall spricht man von Sekundärschutzeinrichtungen. Bei diesen Anlagen werden die Primärgrößen (Spannung und Strom) durch einen Schutzwandler auf genormte Sekundärgrößen umgeformt, die typischerweise 100 V Spannung und 1 A Strom sind. [6] [7]

Die ältesten Schutzeinrichtungen sind die elektromechanischen Schalter, die auch noch heutzutage im Niederspannungsnetz eingesetzt werden. In diesen Geräten wird ein Relais so eingestellt, dass es nur bei (oder über) einer bestimmten Erregung nach Schutz-Auslösekriterium geschlossen (oder geöffnet) wird. [6]

Die nächste Generation der Schutzeinrichtungen sind die elektronischen Schutzeinrichtungen. Die Schutzfunktion in diesen Geräten wird durch analoge (Widerstände, Kondensatoren sowie leistungselektronische Elemente) und digitale Elemente erfüllt. Diese elektronischen Bauteile steuern dann das Schutzrelais. [6]

Heutzutage werden überwiegend digitale Schutzeinrichtungen in Netzen von Mittel- bis Höchstspannungsebene verwendet. Der Schutzschalter wird durch einen Mikroprozessor gesteuert. In einem EPROM kann der Schutzalgorithmus gespeichert werden. Der wesentlichste Vorteil dieser Schutzeinrichtung gegenüber den anderen ist, dass die Parameter der

Schutzfunktion leicht verändert werden können. Zu dieser Kategorie gehört einer der Gegenstände der Untersuchung dieser Arbeit, der ein Siemens SIPROTEC 7SL87 Schutzrelais ist (siehe Kapitel 4.4). [6]

2.2.3 Distanzschutz

Wie es schon vorher erwähnt wurde, können unter Schutzeinrichtungen nach Schutzverfahren Überstromschutz, Distanzschutz, Differentialschutz und Frequenzschutz unterschieden werden. Da ein Distanzschutz in dem untersuchten Netz existiert, wird in diesem Kapitel nur diese Schutzfunktion detaillierter betrachtet. [7]

Die Grundidee des Distanzschutzes ist, dass durch die Kurzschlussspannung und den Kurzschlussstrom die Fehlerimpedanz und dadurch den Abstand der Fehler bestimmt werden kann. Mit dieser Methode kann eine sehr gute Selektivität erreicht werden und können gleichzeitig die Nachteile des Überstromschutz durch langen und kurzschlussstromabhängigen Auslösezeiten vermieden werden.

Soll ein Netz- oder Leitungstück in mehrere Zonen aufgeteilt werden (siehe Abbildung 2.8), können diese Zonen im Fall eines Fehlers mit Hilfe eines Distanzschutzes in folgender Weise selektiv abgeschaltet werden: [8]

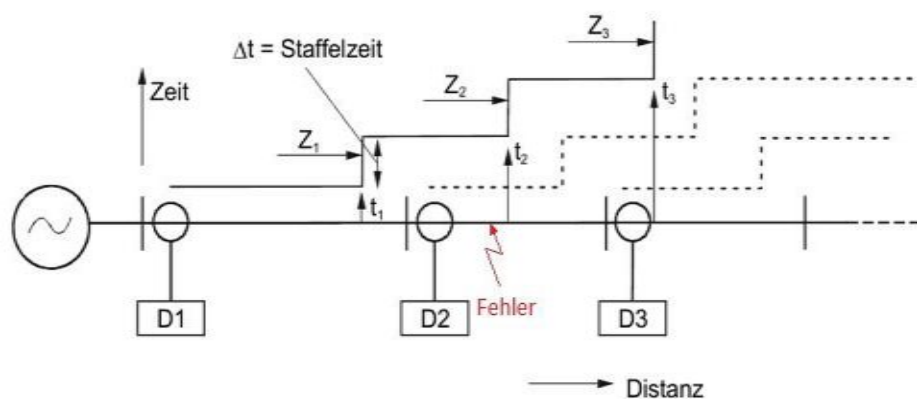


Abbildung 2.8: Netz mit Distanzschutz [8]

Wenn ein Fehler im System entsteht, schaltet das Schutzrelais ohne Verzögerung, in dessen Schutzzone der Fehler auftaucht. In dem obigen Beispiel (siehe Abbildung 2.8) befindet sich der Fehler zwischen Schutzrelais D2 und D3. Es bedeutet, dass D2 ohne Verzögerung abschalten muss. Es nennt man Schnellzeitstufe, in der die Abschaltzeit nur durch die Eigenzeit des Schutzrelais begrenzt wird. Wie in der Abbildung 2.8 sichtbar ist, schaltet der Schutzrelais D1 nur zeitverzögert t_2 , wenn D2 versagt. Diese Distanzstufen werden als Reserveschutz bezeichnet. Wegen der Ungenauigkeiten der Distanzmessung kann der Fehlerort nicht präzise bestimmt werden. Diese Unschärfe entsteht aufgrund von

- Fehler der Messwandler,
- Messfehler der Schutzeinrichtung,
- Schwankungen der Leitungsparameter,
- Schwankungen der Bodenparameter.

Deshalb wird die Schnellzeitzone eines Schutzrelais um 10...20 % kleiner definiert als seine reale Distanzzone. Es gilt ebenfalls für die weiteren Zeitstufen, wobei diese Sicherheitsabstände addiert werden. Demzufolge werden im Allgemeinen nur fünf Zeitstufen in einem System eingebaut, nämlich eine Schnellzeitstufe, zwei Reservestufen und zwei Endzeitstufen, die im Fall einer zweiseitigen Einspeisung gerichtet oder ungerichtet sein können. [6]

Eines der wichtigsten Werkzeuge von Distanzschutz ist das Impedanzdiagramm. Ein Beispiel für ein grundlegendes Impedanzdiagramm wird in der Abbildung 2.9 dargestellt. Das Diagramm wird in drei Bereiche aufgeteilt. Mit blauer Farbe wird der Lastbereich markiert. Unter Normalzustand liegt die gemessene Impedanz in diesem Bereich. [9]

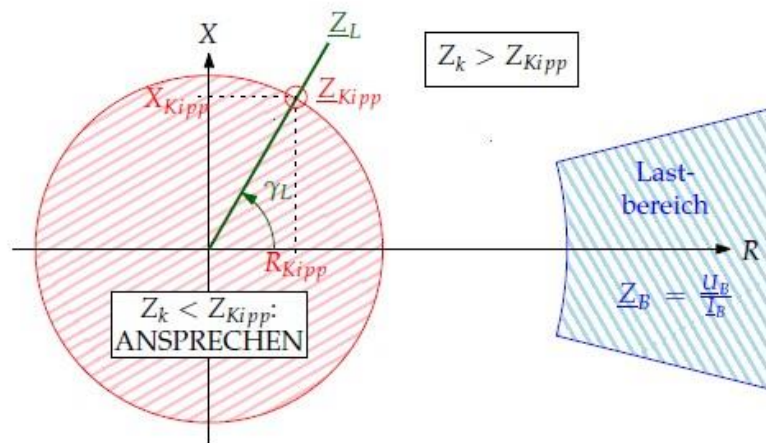


Abbildung 2.9: Darstellung der Distanzgrößen in der komplexen Impedanzebene [9]

Der rote Bereich ist die Ansprechzone. Wenn die Impedanz innerhalb des Kreises liegt, d.h. kleiner als die Kippimpedanz Z_{Kipp} ist, erkennt die Schutzeinrichtung, dass ein Fehler in dem System und in ihrer Schnellzeitzone entsteht. Folglich löst das Schutzrelais ohne Verzögerung aus. Wenn die Fehlerimpedanz weder in Ansprech- noch in Lastbereich liegt, wartet das Schutzrelais, bevor es anspricht. Die Reservestufen des Distanzschutzes können als konzentrische Kreise um die Ansprechzone vorgestellt werden. [9]

Ein weiterer wichtiger Parameter ist die Leitungsgerade Z_L mit dem dazugehörigen Leitungswinkel γ_L . Bei einem modernen Schutzrelais ist diese Gerade die Basis der Bestimmung der Distanzcharakteristik, da die Fehlerimpedanz sich bei einem Fehlerfall idealerweise auf dieser Gerade befindet. Jedoch wird die gemessene Impedanz durch den Lichtbogenwiderstand stark beeinflusst. Es ist ein ohmscher Widerstand, der schwierig messbar ist, da ein nichtlinearer Zusammenhang zwischen Kurzschlussstrom und -spannung entsteht. Deswegen kann dieser Widerstand nur abgeschätzt werden: [9]

$$R_{LiBo} = \frac{U'_L \cdot l_L}{I_k^k} \quad (2.30)$$

wo:

- U'_L – Lichtbogenspannung in V/m
- l_L – Lichtbogenlänge in m
- I_k – Lichtbogen-(Kurzschluss-)strom in A
- k – Exponentialfaktor (1...1,4)

In der Abbildung 2.10 wird es dargestellt, wie die gemessene Impedanz wegen des Lichtbogenwiderstandes von der Fehlerimpedanz abweicht. In diesem Beispiel verursacht diese Verschiebung eine falsche Zonenbestimmung, die zu einer verzögerten Auslösung führt.

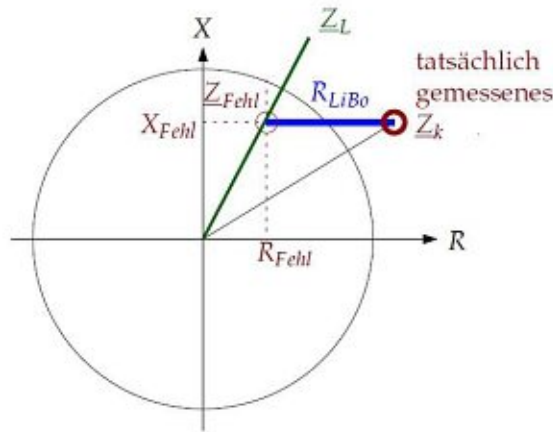


Abbildung 2.10: Einfluss des Lichtbogenwiderstandes [9]

Wegen der begrenzten Möglichkeiten der analogen Messtechnik sind die Relais-Impedanzkennlinien traditionell geometrische Formen, die aus Kreisen, Kreisabschnitten und Geraden zusammengesetzt werden. Zur Kompensation des Lichtbogenwiderstandes existieren mehrere verschiedene Impedanzcharakteristiken nämlich Mischimpedanz-Charakteristik, MHO-Charakteristik oder elliptische Charakteristik, aber eine wirklich effektive Kompensation kann bei heutigen digitalen Distanzschutzgeräten nur durch Polygon-Charakteristik erfolgen. Die vereinfachte Darstellung einer Polygon-Charakteristik ist in der Abbildung 2.11 sichtbar. [8]

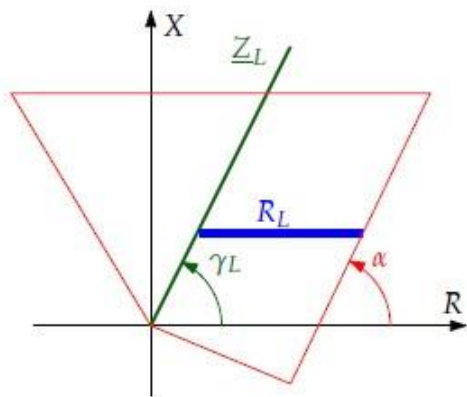


Abbildung 2.11: Polygon-Charakteristik [9]

Wenn der Neigungswinkel α dem Leitungswinkel γ_L gewählt wird, wird eine konstante Lichtbogenreserve R_L bis zur Kippreaktanz X_{Kipp} gewährleistet. [9]

Wenn ein Kurzschluss im Netz auftaucht, wird die Funktion *Anregung* aktiviert. Die Hauptaufgabe dieser Funktion ist den Fehler phasenrichtig zu erkennen und zu klassifizieren. Eine Phasenselektivität ist besonders wichtig bei einpoligen Fehlern, damit der Schalter nur in der fehlerbehafteten Phase ausgelöst wird. Gleichzeitig werden die Stufenzeiten gestartet und die Messschleife auch ausgewählt. Nach Anregekriterium können drei Arten von Anregungen unterschieden werden. [8]

- **Überstromanregung**

Diese Methode ist die einfachste und schnellste Fehlererkennungsart. Wenn der gemessene Strom einen Schwellwert I_{\gg} überschreitet, wird die Anregung aktiviert. Der Nachteil dieser Fehlererkennung ist, dass der Fehlerstrom mindestens zweimal größer als der maximale Laststrom sein muss, welche Voraussetzung bei einpoligem Erdschluss in einem isolierten Netz nicht erfüllt wird. [8]

$$k \cdot I_{B,max} < RV \cdot I_{F,min} \quad (2.31)$$

In der Praxis wird der Wert 2 für den Sicherheitsfaktor gewählt. RV ist das Rückfallverhältnis, im Allgemeinen gleich 0,95. [9]

- **U/I - Anregung**

Wenn der Fehlerstrom zu klein für eine Überstromanregung ist, kann die Überwachung des Spannungsniveaus für die Fehlererkennung verwendet werden. Wenn die Spannung unter einen bestimmten Schellwert ($0,7 \cdot U_N$) sinkt, erkennt die Schutzeinrichtung den Fehler in der gegebenen Phase. Um Fehlansprechen bei abgeschalteter (spannungsloser) Leitung vermeiden zu können, wird die Unterspannungsanregung nur über ein eingestellten Stromschwelle ($I_{>} \approx 0,2 \dots 0,5 \cdot I_N$) eingesetzt. [8]

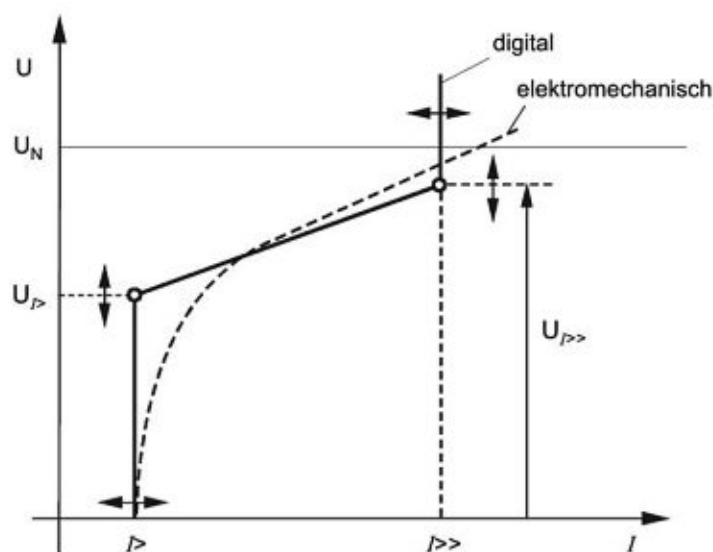


Abbildung 2.12: U/I – Kennlinie [8]

Diese Methode kann mit einer Winkelüberwachung ergänzt werden. Da der Kurzschlusswinkel ($> 70^\circ$) wesentlich größer als der Lastwinkel ($\pm 30^\circ$) ist, kann dieses Merkmal auch als Anregekriterium genutzt werden. [8]

- **Impedanzanregung**

Bei den heutigen modernen Schutzgeräten ist die Impedanzcharakteristik sehr gut geeignet zur Unterscheidung zwischen Fehler- und Lastzustand. Die Impedanzen müssen dazu in allen möglichen Fehlerschleifen (sechs Fehlerschleifen) überwacht und gerechnet werden. [8]

Die Fehlerimpedanz der drei grundlegenden Fehlerfälle kann nach [9] bestimmt werden:

- Dreipoliger Kurzschluss ohne Erdberührung:

$$\underline{Z}_F = \frac{U_{12}}{I_{12}} \quad (2.32)$$

- Zweipoliger Kurzschluss ohne Erdberührung (L2-L3):

$$\underline{Z}_F = \frac{U_{23}}{I_{23}} = \frac{U_{23}}{2 \cdot I_2} \quad (2.33)$$

- Einpoliger Erdschluss (L1-E):

$$\underline{Z}_F = \frac{U_{1E}}{I_1 + k_0 \cdot I_Y} \quad (2.34)$$

Der Erdfehlerfaktor k_0 kann durch Mit- und Nullimpedanz der Leitung berechnet werden. I_Y ist der Summenstrom, der gleich dem dreifachen Nullstrom ist.

$$k_0 = \frac{1}{3} \cdot \left(\frac{Z_L^0}{Z_L^1} - 1 \right) \quad (2.35)$$

$$I_Y = I_1 + I_2 + I_3 = 3 \cdot I^0 \quad (2.36)$$

3 Methodik

Dieses Kapitel beschäftigt sich mit der grundlegenden Konzeption der Projektdurchführung. Erstens wird der Aufbau des Prüfstandes betrachtet, dann werden die unterschiedlichen Phasen des Projektes kurz zusammengefasst.

3.1 Aufbau des Messsystems

Das Messsystem besteht aus drei Hauptteilen (siehe Abbildung 3.1).

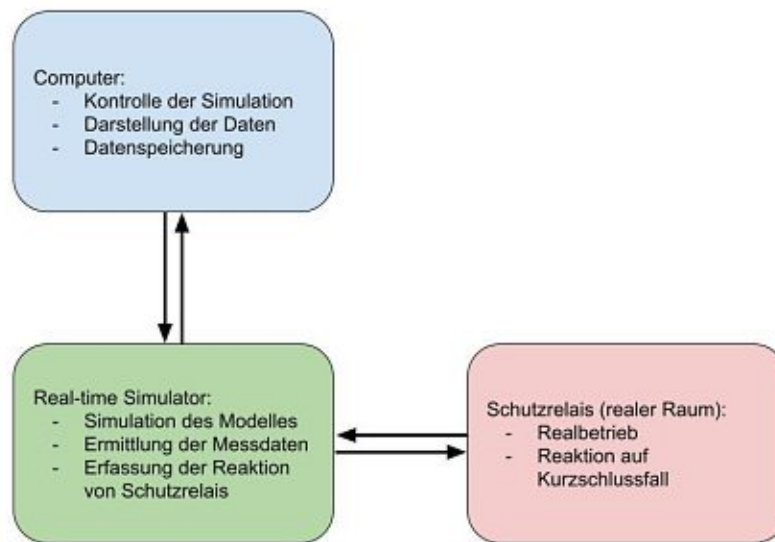


Abbildung 3.1: Grundlegender Messaufbau

Die Simulation des in Kapitel 1 vorgestellten Modells läuft auf dem Real-Time-Simulator OP5600. Die zwei Messblöcke AAF32 und AAF34 stellen die für die Prüfung relevanten Messdaten zur Verfügung. Diese Spannungs- und Strommesswerte sind äquivalent mit den Messwerten des Distanzschutzes im realen Netz. Deswegen werden sie dem Schutzrelais in realen Raum übertragen. Die andere Aufgabe des Simulators ist die Erfassung der Reaktion von Schutzrelais. Das Verhalten des Schutzrelais wird in dieser Weise in die Simulation implementiert. Das heißt, die Simulation und das Schutzrelais interagieren miteinander. Für diesen Zweck wird das Modell mit Leistungsschaltern ergänzt (siehe Abschnitt 5.3.4). Am Computer wird das Modell vorbereitet. Während der Simulation werden hier die Kontrolle der Simulation bzw. die Darstellung und Speicherung der Messdaten durchgeführt. Mit den verwendeten Geräten und Programmen beschäftigt sich das Kapitel 4 detaillierter. Der ausführliche Aufbau des Messsystems wird im Kapitel 5 betrachtet.

3.2 Projektphasen

Das Projekt besteht aus fünf Hauptphasen. Die erste war die Planungsphase. Hier wurden die Lösungsmöglichkeiten der Aufgabe überlegt und wurde die Ausführung des Prüfstandes geplant.

In der zweiten Phase wurden die verwendeten Programme und Geräte vorbereitet. Da der an der Universität zur Verfügung stehende Simulator nur über eine alte Lizenz verfügt, war es ein kritischer Punkt, die entsprechende Version der Programme zu besorgen und zu installieren. Andernfalls konnten die Programme miteinander nicht zusammenarbeiten. Außerdem wurde die Kommunikation zwischen den Geräten sowie die IP-Adresse eingestellt.

Im nächsten Schritt wurde der Simulator getestet. Durch einfachere Modelle und Simulationen wurde das Gerät untersucht und seine Funktionen richtig eingestellt. Diese Periode war essenziell, um die späteren Simulationen problemlos durchführen zu können.

Nachdem das richtige Verhalten des Simulators bei einfachen Modellen sichergestellt wurde, erfolgte der Aufbau des Messsystems. Zu dieser Phase gehörte einerseits der Umbau des Windparkmodelles in eine, für das Programm RT-LAB passende Form (siehe Abschnitt 5.1), andererseits der hardwareseitige Aufbau des Prüfstandes.

Als das System sowohl softwareseitig als auch hardwareseitig fertiggestellt wurde, konnte es mit den Tests, weiters mit den Messungen angefangen werden. Das Ziel war hier, dass das Schutzrelais auf die simulierten Fehlerfälle richtig reagiert. Es musste die Fehlerart und -distanz zurecht erkennen und nach seinen Einstellungen mit oder ohne Verzögerung auslösen. Danach mussten diese Ergebnisse mit Messungen von einer anderen authentifizierten Testmethode verglichen werden. Dazu wurde das Testgerät CMC 356 von Firma Omicron verwendet. Dieses Gerät wurde explizit fürs Testen der Schutzgeräte entwickelt und seit langem ist es sowohl in der Praxis als auch im Forschungsbereich genutzt. Deswegen diente es als ein idealer Referenzpunkt für die Messungen.

In der letzten Phase wurden die Messergebnisse ausgewertet bzw. wurde die Diplomarbeit fertiggestellt. Die Zusammenfassung und Auswertung der Messergebnisse befinden sich in dem Kapitel 6.

4 Verwendete Programme und Geräte

4.1 MATLAB und Simulink

MATLAB und Simulink müssen im technischen Bereich besonders auf dem Gebiet von Elektrotechnik niemandem vorgestellt werden, jedoch ist es wichtig ein wenig über diese Programme und ihre Rolle in diesem Projekt zu schreiben.

MATLAB und sein Zusatzprogramm Simulink sind die Produkte des US-amerikanischen Unternehmens MathWorks. Cleve Moler hat die erste Version von MATLAB am Ende der 1970er Jahren an der Universität New Mexiko entwickelt. Später hat er mit Hilfe von Jack Little das Unternehmen MathWorks gegründet. [10]

Das Grundziel dieses Programmes ist die Erleichterung der Lösung von komplexen mathematischen Rechnungen und der grafischen Darstellung der Ergebnisse ohne Kenntnisse hochwertiger Programmiersprachen. Nach der Erscheinung weiterer Toolboxen hat sich das Programm nicht nur in mathematischen, sondern auch in technischen Bereichen wie Kommunikations-, Luftfahrt- und Elektrotechnik, usw. verbreitet. Mit Hilfe von Simulink kann man nichtlineare Systeme modellieren und Simulationen durchführen. [10] [11]

In diesem Projekt wurde MATLAB Version 8.1 (2013a) 32-bit benutzt. Obwohl diese bei der Entwicklung des Messsystems nicht die neuste Version von MATLAB war, wurde wegen der Kompatibilität mit dem Programm RT-LAB Version:10.7.4.436 verwendet. Hier muss es betont werden, dass RT-LAB nur mit 32-bit MATLAB problemlos zusammenarbeiten kann, da sie ebenso eine 32-bit Software ist. Darüber wird es im nächsten Abschnitt diskutiert.

Die folgenden Toolboxen waren unbedingt notwendig bei diesem Projekt:

- Simscape
- SimElectronics
- SimPowerSystems
- MATLAB Coder
- Simulink Coder
- Symbolic Math Toolbox

Diese Pakete können mit MATLAB und Simulink heruntergeladen und installiert werden. Neben ihr muss noch die folgenden Pakete nach der Installation von RT-LAB hinzugefügt werden:

- RT-LAB
- RT-EVENTS Blockset
- ARTEMIS Blockset

Zur Installation müssen die passenden Installationsfiles in alle drei Fälle mit Befehl *run* (in MATLAB Command Window) und Eingabe ihres Dateipfades geöffnet werden.

4.1.1 SimPowerSystems

SimPowerSystem (SPS) ist eine der wichtigsten Simulink-Toolboxen bezüglich dieser Arbeit, da das Modell des Windparks, der als Gegenstand der Untersuchung dieses Projektes dient, durch SPS-Blöcke aufgebaut wurde. [12]

Die Toolbox wurde von Hydro-Québec Research Institute (IREQ) entwickelt. Das Ziel dieser Entwicklung ist es, die Simulation und Analyse elektrotechnischer Netze und Systeme in der Umgebung von Simulink zu ermöglichen. Die Simulation des Stromkreises kann ebenfalls die Wechselwirkungen mit mechanischen, thermischen, und anderen Disziplinen umfassen. [12]

Das System von SimPowerSystem wird durch spezielle physische Modellierungsblöcke (Physical Modeling Block) und ihre Verbindungen (Connection Line Interface) aufgebaut. Die Bibliotheken von SimPowerSystem beinhalten Modelle typischer energietechnischer Einrichtungen wie Transformatoren, Leitungen, Maschinen und Leistungselektronik. Diese Modelle stammen aus Lehrbüchern und ihre Gültigkeit basiert auf den Erfahrungen des Test- und Simulationslabors für Stromversorgungssysteme von Hydro-Québec. [12]

4.2 OPAL-RT und RT-LAB

OPAL-RT wurde von Jean Bélanger und Lise Laforce im Jahr 1997 in Kanada gegründet. Ihre Zielsetzung war die Entwicklung eines Echtzeitsimulators (real-time simulator), der sich auf dem Schreibtisch jedes Systemingenieurs befinden kann. [13]

Aber was ist ein *Real-Time-Simulator* und was bedeutet *Real-Time-Simulation*? Zwei Arten der Simulationen können unterschieden werden. Die erste ist die sogenannte *non real-time Simulation*, in der variable Zeitschritte (variable time-steps) verwendet werden. Diese Technologie ist sehr gut geeignet für die Lösung von nichtlinearen Systemen mit Hochfrequenz, aber kann keine Echtzeitsimulation durchführen.

Die zweite ist die *Real-Time-Simulation*. Die Grundvoraussetzung ist die Nutzung von festen Zeitschritten (fixed time-step/ discrete-time). Aber allein die Nutzung von *fixed time-step* ist nicht genügend. Bei einer *discrete-time* Simulation werden die mathematischen Funktionen und Gleichungen nacheinander gelöst. Die notwendige Berechnungszeit dieser mathematischen Funktionen und Gleichungen kann länger oder kürzer sein als die Dauer des Simulationszeitschritts. Wenn die Berechnungszeit kürzer als die Dauer von einem *fixed time-step* ist (siehe Abbildung 4.1 (a)), erfolgen die Simulationsvorgängen schneller als bei einem realen System. Wenn es länger ist (siehe Abbildung 4.1 (b)), ist das Verfahren langsamer. Diese beiden Situationen werden als Offline-Simulation bezeichnet. [14]

Damit eine Echtzeitsimulation durchgeführt werden kann, müssen die internen Variablen und Ausgaben der Simulation innerhalb der gleichen Zeitspanne erzeugt werden, wie ihr physikalisches Gegenstück. Dazu muss natürlich das Zeitintervall der Berechnung kleiner sein als die Dauer eines Zeitschrittes, aber die berechneten Variablen und Systemen werden erst in einem vorgeschriebenen Zeitpunkt ausgegeben (siehe Abbildung 4.1 (c)). Wenn die Kalkulation innerhalb dem erforderlichen festen Zeitschritt nicht durchgeführt werden kann, verursacht es einen Fehler, den man *overrun* nennt. [14]

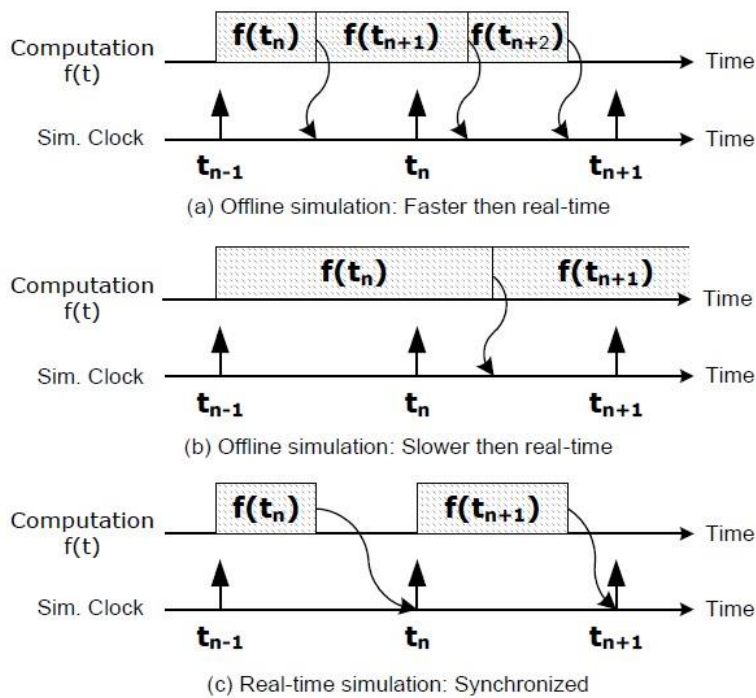


Abbildung 4.1: Art von fixed time-step Simulationen [14]

Zum besseren Verständnis des Systems eines Real-Time-Simulators wird die im Projekt verwendete Technologie von OPAL-RT zunächst kurz vorgestellt. Sie basiert auf zwei Grundkomponenten:

- die Softwareplattform RT-LAB (und ihre zusätzlichen Toolboxes)
- die Hardwareplattform (Real-Time-Simulator OP5600)

Die nächste zwei Kapitel geben einen kurzen Überblick über diese Komponenten.

4.2.1 RT-LAB

Das Programm RT-LAB ermöglicht die Interaktion zwischen Simulink-Modelle und der realen Welt in Echtzeit. Das Programm ist sehr vielfältig in Bezug auf Simulationsmethode, Kommunikationsprotokoll, Beobachtungsmöglichkeit von Variablen, usw. Jedoch fokussiert dieser Abschnitt nur auf die wichtigsten Anwendungsbereiche und Einstellungen des Programmes, die eng mit dieser Arbeit verbunden sind. [15]

Während des Projektes wurde die Version: v10.7.4.436 von RT-LAB benutzt, deshalb gelten diese Beschreibung nur für diese Version, bei späteren Ausgaben können Abweichungen entstehen.

Ein entsprechender Aufbau von Simulink-Modell ist essenziell für eine reibungslose Zusammenarbeit zwischen beiden Programmen. Ein Modell muss in Subsysteme aufgeteilt werden. Die zwei Arten der Subsysteme sind *GUI* und *Computation* Subsystem.

Das vorherige ist für die Steuerung des Systems und die Darstellung der Daten verantwortlich. Es kann Blöcken wie *scopes*, *displays*, *manual switches*, *constants* beinhalten. Dieses Subsystem läuft asynchron von dem *Computation* Subsystem auf dem Host-PC und darf keine Signalerzeugung, keine mathematischen Operationen und kein physikalisches Modell erfassen. In einem Modell darf nur ein *GUI* Subsystem sein. [15]

Computation Subsystem beinhaltet alle Berechnungselemente des Modells, mathematische Operationen, I/O-Blöcke, Signalgeneratoren, physikalische Modelle. Mehrere *Computation* Subsysteme können sich gleichzeitig in einem Modell betätigen. Jedes Subsystem wird auf einem CPU-Kern des Real-Time-Gerätes ausgeführt. Sie kommunizieren miteinander synchron. Die Simulationen funktionieren ohne *GUI* Subsystem, aber sie müssen mindestens ein *Computation* Subsystem beinhalten. [15]

Die Benennung der Subsysteme ist ebenfalls sehr wichtig, da das Programm RT-LAB durch die richtige Benennung zwischen ihnen unterscheiden kann. Diese Regeln werden in der nächsten Tabelle dargestellt. [15]

Tabelle 4.1: Benennungsvorschriften der Subsysteme

Funtion	Benennung
GUI Subsystem	SC_anyName
Computation Subsystem (main)	SM_anyName
Computation Subsystem (additional)	SS_anyName

Außer der Benennung der Subsysteme muss noch auf zwei Simulationseinstellungen geachtet werden. Es ist erforderlich sowohl als Samplezeit feste Zeitschritte (*fixed-step*) einzustellen als auch als Simulationsdauer unendlich (*inf*) einzustellen.

Vor der Öffnung von RT-LAB muss der Server *MetaController* aktiviert werden, der die Interaktion zwischen RT-LAB und Simulator ermöglicht. Im Programm kann im Fenster *Project Explorer* ein neues Projekt erstellt werden, in welches das vorbereitete Modell importiert werden kann. Danach muss das Simulink-Modell der gewählten Plattform kompiliert werden (*build*). Zum Laufen der Simulation muss erst das Modell auf den Echtzeitsimulator hochgeladen werden (*load*), dann ist ein Programmstart möglich (*execution*). [15]

Bei dem Simulationsstart wird ein Modell im Simulink geöffnet und gestartet, der durch RT-LAB aus dem *GUI*-Subsystem automatisch generiert wird. Durch dieses Fenster kann die Variablen geändert und die unterschiedlichen Vorgänge beobachtet werden. Es ist möglich die Werte im Fenster *Variable Viewer* oder durch ein Python-Skript automatisch zu modifizieren. Mit der Automatisierung der Simulationen mit Python-Skript beschäftigt sich der Abschnitt 5.3.5 detaillierter.

4.2.2 OP5600 Real-Time-Simulator

An der Technischen Universität Wien ist ein Real-Time-Simulator von OPAL-RT aus der Serie OP5600 vorhanden, der während dieser Arbeit verwendet wurde. Die wichtigsten Eigenschaften und der Aufbau des Gerätes werden in diesem Kapitel betrachtet und in Tabelle 4.2 zusammengefasst. [16]

Tabelle 4.2: Liste die Eigenschaften von OP5600 [16]

Elemente	Menge	Beschreibung
Betriebssystem	1	Redhat v2.6.29.6-opalrt-6.1
Gehäusotyp	1	OP5600 Chassis
CPU	2	Intel Xeon Six-Core 3.46GHz 12M Cache
Total Core #	12	
Speicher	2	4 GB
Motherboard		X8DTL-I-O Supermicro Motherboard, Intel® Xeon® processor 5600/5500 series, with QPI up to 6.4 GT/s
AC Input		100-240V, 60-50Hz

Hier ist es wesentlich die Zahl des Prozessorkernes zu erwähnen. In diesem Fall gibt es 12 Kerne, das bedeutet, dass es möglich ist das Modell in 12 *Computation* Subsysteme aufzuteilen.

Das vorhandene Gerät verfügt über 16 analoge und 32 digitale Kanäle. Analog kann es im Spannungsbereich maximal zwischen -16 V und +16 V arbeiten. Die digitalen Inputs haben flexible Eingangsspannung von 4 bis 50 V, es ist bei digital Outputs 5-30 V. Wenn die rückseitigen DB37F-Anschlüsse (siehe Abbildung 4.3 A) als digitaler Ausgang genutzt werden, ist eine externe Spannungsversorgung notwendig, die mit Hilfe des 5/12 V-Anschlusses (siehe Abbildung 4.3 D) und eines *loopback* Bordes gesichert werden kann. Der Aufbau der Schnittstellen ist in Abbildung 4.2 und Abbildung 4.3 sichtbar. [16]

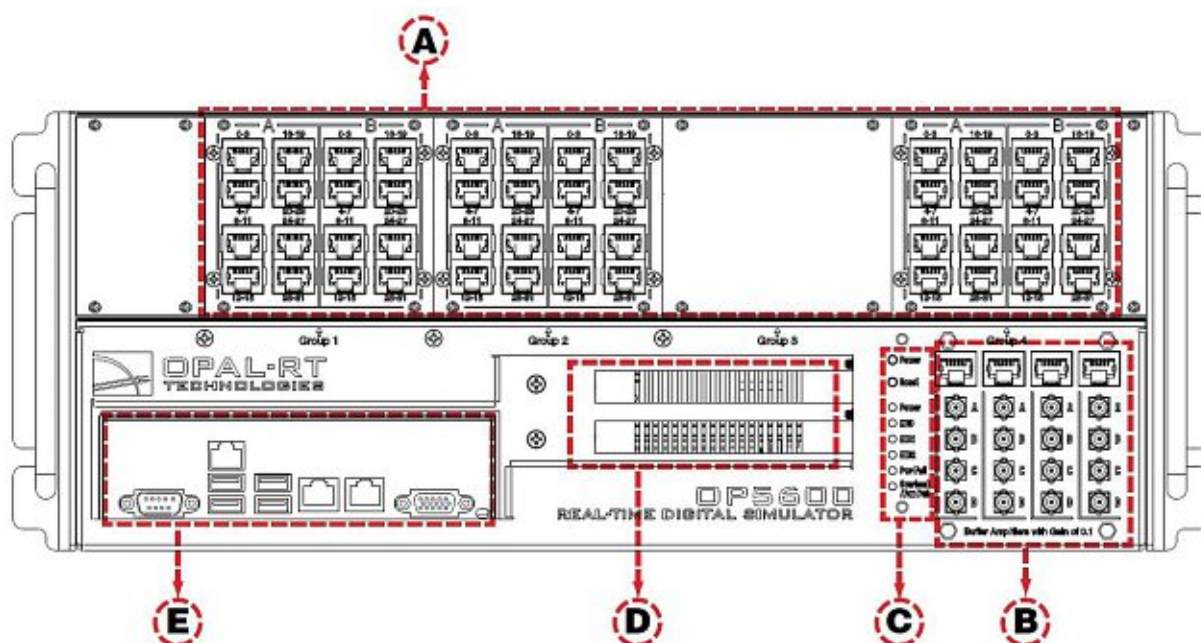


Abbildung 4.2: OP5600 Frontschnittstelle [16]

Die Frontseite des Simulators besteht aus den folgenden Bauteilen:

- A. 3 Panel mit RJ45-Anschlüssen. Jeder Anschluss ist mit den vorderen und hinteren Mezzaninen auf der Trägerplatine verbunden. Jedes Panel ist in zwei Slots (Slot A und Slot B) aufgeteilt. In einem Slot gehört ein Mezzanin zu einer Spalte, die aus vier RJ45-Anschlüssen besteht. Diese RJ45-Buchsen sind zu zweit zu gruppieren und jede beinhaltet vier analoge oder digitale Kanäle.

- B. RJ45-Anschlüssen mit Mini-BNC-Terminals. Mit Hilfe dieses Panels ist es möglich einen Prozess mit einem äußeren Überwachungsgerät wie ein Oszilloskop zu überwachen.
- C. Überwachungsschnittstelle des Zielcomputers mit zwei Drucktasten und 6 LED-Anzeigen. Die obere Taste ist für Einschalten des Apparates (POWER), die zweite ist für Neustarten (RESET).
- D. Optionale PCI- oder PCIe-Anschlusssteckplätze.
- E. Standard-Computeranschlüsse wie Maus und Tastatur, USB-Anschlüsse, Monitor, Netzwerkanschlüsse.

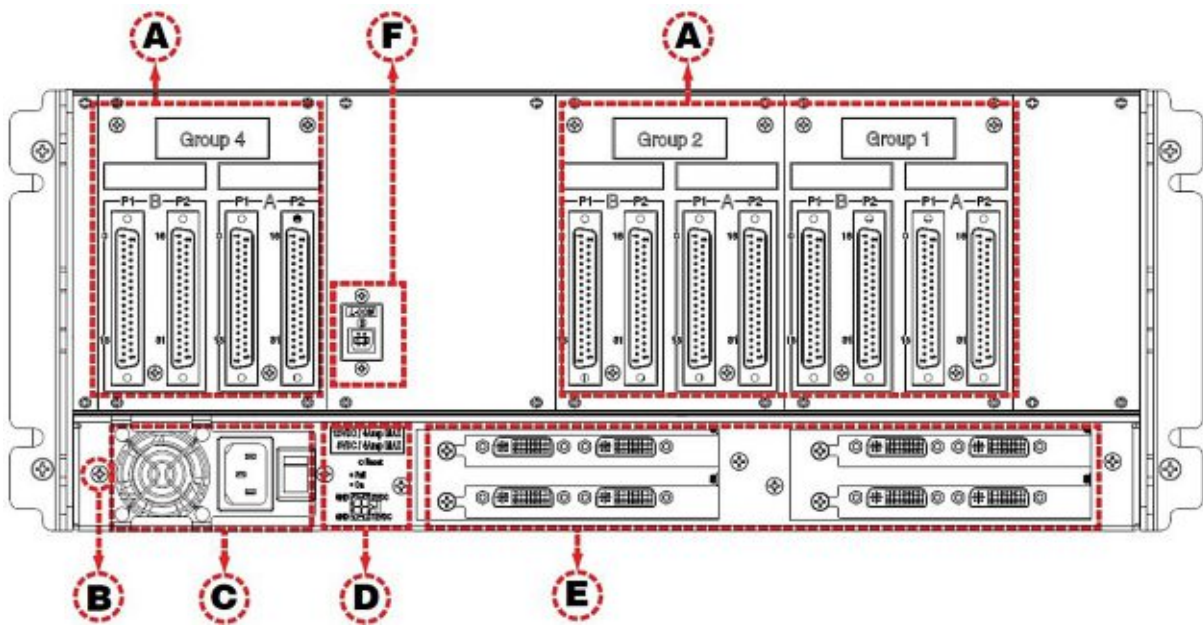


Abbildung 4.3: OP5600 rückseitige Schnittstelle [16]

Die Rückseite des Simulators besteht aus den folgenden Bauteilen:

- A. DB37F I/O-Anschlüsse mit Verbindung zu den vorderen und hinteren Mezzaninen der Trägerplatine. Die Aufteilung der Paneele ist sehr ähnlich wie im Fall der Frontpaneelle. Der einzige Unterschied liegt darin, dass ein DB37F-Anschluss alle Kanäle eines Mezzanins (16 Kanäle) beinhaltet.
- B. Erdungspunkt: Der Simulator kann einer elektromagnetischen Störung (EMI) unterworfen sein, wenn er in der Nähe zu anderen Geräten platziert ist. In diesem Fall ist es empfohlen die Masse des OP5600 mit dem Regal zu verbinden, um Schäden am Simulator durch EMI zu vermeiden.
- C. Der Stromanschluss und der Ein- / Ausschalter
- D. Reset-Taste und 5/12 V-Anschluss
- E. Optionaler PCI- oder PCIe-Anschluss
- F. Der USB-Anschluss für die JTAG-Programmierung

Die während der Simulation verwendeten Kanälen werden im Anhang C.: zusammengefasst.

4.3 Doble F6350e

Da der Simulator OP5600 nur analoge Signale zwischen maximal -16...16 V abgeben kann, muss ein Verstärker zwischen dem Real-Time-Simulator und dem Schutzrelais eingebaut werden. Für diese Aufgabe wurde der externe Verstärker F6350e von Firma Doble verwendet, der über 6 Spannungs- und 6 verstärkte Stromquellen verfügt. Der Apparat ist auch zur Batteriesimulation fähig. [17]

Die Konfiguration des Gerätes erfolgt durch den Software *F6 Multiple Amplifier Configurator* von Doble. Die Versionsnummer in diesem Fall ist V 4.1. Erstens musste die IP-Adresse des Verstärkers durch die serielle Schnittstelle eingestellt werden, dann konnte die Konfiguration des Gerätes durch Ethernet-Verbindung durchgeführt werden. Die wichtigsten Konfigurationseinstellungen werden in Tabelle 4.3 zusammengefasst. [17]

Tabelle 4.3: Einstellungen des Verstärkers

Communications	Given Name	F6Type	ER	Sources	Transient	Select Sources
128.131.75.109	TUW_doble1	F6350	<input checked="" type="checkbox"/>	6V & 6I	False	<input checked="" type="checkbox"/>

Für den Aufbau der Simulation müssen alle 6 Spannungs- und 6 Stromquellen benutzt werden, so der Leistungsbereich auf 75 VA begrenzt wird. Der eingestellte Spannungsbereich aller Spannungsausgängen ist 150 V_{eff}, der Strombereich ist 8,75 A_{eff}. Die Position und die Bezeichnung der Spannungs- und Stromquellen werden in Abbildung 4.4 dargestellt.

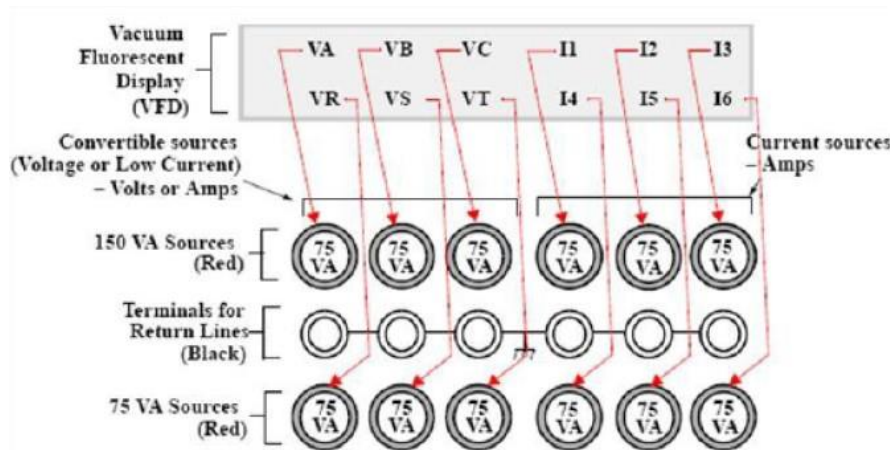


Abbildung 4.4: Quellenbezeichnungen für die 12 75VA-Quellen [22]

Für Eingangssignale steht ein DB-15-Anschluss zur Verfügung. Der Zusammenhang zwischen Ein- und Ausgangswerten ergibt sich in folgender Weise: Bei Spannungsquellen gehört eingangsseitig 6,698 V_{eff} pro Quelle zur Erzeugung des vollen Bereichs (150 V_{eff}), bei Stromquellen gehört die Eingangsspannung 3,5 V_{eff} zum Ausgangswert 8,75 A_{eff}.

4.4 Siemens SIPROTEC 7SL87

Der SIPROTEC 7SL87 ist ein Schutzrelais von Siemens, der die Funktion des Leitungsdifferential- und Distanzschutzes in einem Apparat kombiniert. Dieses Gerät ist fähig zu 1- und 3-poliger Auslösung unter minimal 9 ms Auslösezeit. [18]

Während dieses Projektes wurden zwei solche Geräte verwendet, die durch die Firma Vetö in einem gemeinsamen Gehäuse installiert wurden. Die Ein- und Ausgänge beider Schutzrelais wurden mit Bananenbuchsen an der Wand des Gehäuses ausgeführt. Der Schaltplan des Apparates wird im Anhang A.: dargestellt. Die für die Messung verwendeten Ein- und Ausgänge werden im Anhang C.: zusammengefasst. [18]

4.5 Omicron CMC 356

Das CMC 356 ist das universelle Relaisprüfgerät der österreichischen Firma Omicron. Das Gerät ermöglicht vielfältige Simulationen und Testen für Schutzrelais, Messwandler, Stromzähler und Stromversorgungsanalysatoren. Das Test-Set besteht hardwareseitig aus dem Prüfgerät (siehe Abbildung 4.5) und den Verkabelungen, softwareseitig aus dem Programm Test-Universe. [19]



Abbildung 4.5: CMC 356 Frontseite [19]

Liste der Anschlüsse:

- Spannungsausgänge (voltage output):
4 × 300 V oder 2 × 600 V
- Stromausgänge (current output A/B):
6 × 32 A / 6 × 430 VA oder
3 × 64 A / 3 × 860 VA oder
1 × 128 A / 1 × 1000 VA
- Generator-Gesamtbuchse
3 × 300 V und 3 × 32 A
- DC-Stromversorgung:
Zwischen 0...264 V
- 4 Binärausgänge (binary output)
- DC-Messeingänge (anlog DC input)
(0 ... 10 V und 0 ... 20 mA)
- 10 Multifunktionseingänge (binary/ analog input)

Obwohl das Prüfgerät über zahlreiche analoge und binäre Ein- und Ausgänge verfügt, wie es in der obigen Liste sichtbar ist, wird nur die Generator-Gesamtbuchse (siehe Abbildung 4.5 A) und ein binärer Eingang während der Messungen als Anschluss verwendet.

5 Messaufbau

In diesem Kapitel wird der Vorbereitungs- und Aufbauvorgang des Messsystems dargestellt. Es fokussiert darauf, wie das ursprüngliche Modell des Windparks umgebaut wurde und wie die verwendeten Geräte miteinander verbunden sind.

5.1 Umstellung des Modelles

Wie es schon im Kapitel 4.2.1 beleuchtet wurde, muss ein Simulink-Modell umgestellt werden, damit es für eine Echtzeitsimulation mit RT-LAB geeignet ist. Hier wird nur die erste, einfachste Testversion des Modelles betrachtet. Erstens müssen die Bausteine in zwei Subsysteme wie in Abbildung 5.1 geordnet werden.

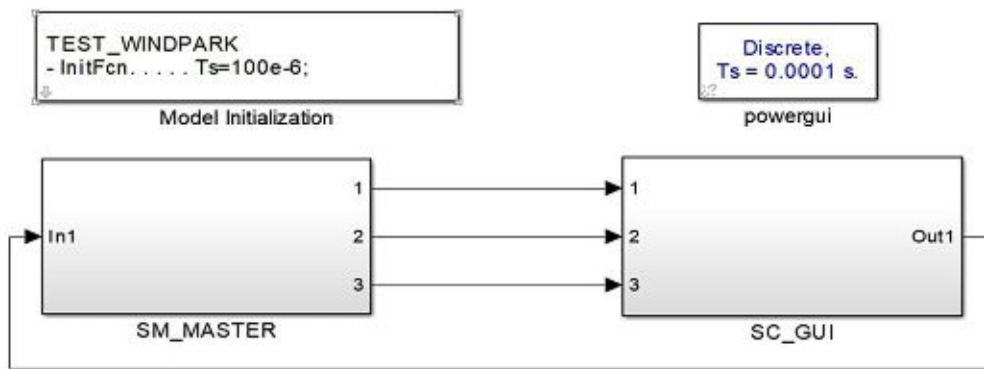


Abbildung 5.1: Subsystemen im Windparkmodell

Neben den zwei Subsystemen sind noch zwei Blöcke in der Abbildung sichtbar. Mit Hilfe des ersten Blocks (Model Initialization) wird die Samplezeit der Simulation festgestellt, die in diesem Fall $T_s = 100 \mu s$ ist. Der andere ist der Block *powergui*, der unbedingt notwendig ist für die Funktion der *SimPowerSystems* Bausteine. Aber dieser Block darf sich nicht in *Computation Subsystem* (in diesem Fall SM_MASTER) befinden, weil es ein Problem in RT-LAB bei Kompilieren des Programms verursacht.

Zur Kommunikation der Subsysteme während der Real-Time-Simulation ist es nötig den Funktionsblock *OpComm* in sie einzubauen. Diesen Block beinhaltet die RT-LAB Bibliothek. Er muss in beiden Subsystemen unmittelbar nach dem Eingangsport eingesetzt werden.

SM_MASTER umfasst neben der Konstruktion des Windparks die folgenden Blöcke (siehe Anhang F.):

- **OpComm Block:** übermittelt das Startsignal des einpoligen Fehlers von Subsystem *SC_GUI*.
- **OpCtrl:** Für jede verwendete FPGA-Karte (in diesem Fall eine Karte), die physisch in dem System vorhanden ist, muss ein Block in das Modell eingefügt werden. Er definiert den Namen der Controller, Board ID, Art der Synchronisierung, Type der FPGA-Karte. Unter inneren Parametern muss noch der Name des sogenannten *bitstream* Files angegeben werden. Er und der *configuration* File wird von OPAL-RT hergestellt und stellt I/O-Konfiguration dar. (Sie müssen sich im gleichen Verzeichnis wie der File des Modelles befinden.)
- **OpCtrl AnalogOut:** definiert die Verbindung zwischen dem Modell und realem Analog-Port. Als innerer Parameter kann der Controller-Name, die Portnummer bzw. die Zahl und der Spannungsbereich der benutzten Kanäle angegeben werden.

Die Messsignale der Messpunkte (Three-Phase Measurement Block) werden sowohl in analoge Ausgänge als auch ins GUI-Subsystem überführt. Außerdem gibt es eine kleine Änderung im Block *Three-Phase Fault*. Das Startsignal des Fehlers wird als äußeres Signal bestimmt.

SC_GUI wird in der Abbildung 5.2 dargestellt. Es besteht aus zwei Teilen: Die Input-Signale werden unmittelbar in den *OpComm*-Block und dann zur Visualisierung in einen *Scope*-Block eingeführt. Die Signale sind nach Eingangsportnummer:

1. Startsignal des Fehlers
2. *Vabc_AAF32*, *Iabc_AAF32* (insgesamt sechs Signale)
3. *Vabc_AAF34*, *Iabc_AAF34* (insgesamt sechs Signale)

Der andere Teil des Subsystems ist für die Angabe des Startsignals verantwortlich. Durch einen Schalter mit manueller Betätigung kann der Wert des Signals zwischen 0 und 1 (also logisch wahr und falsch) gewechselt werden.

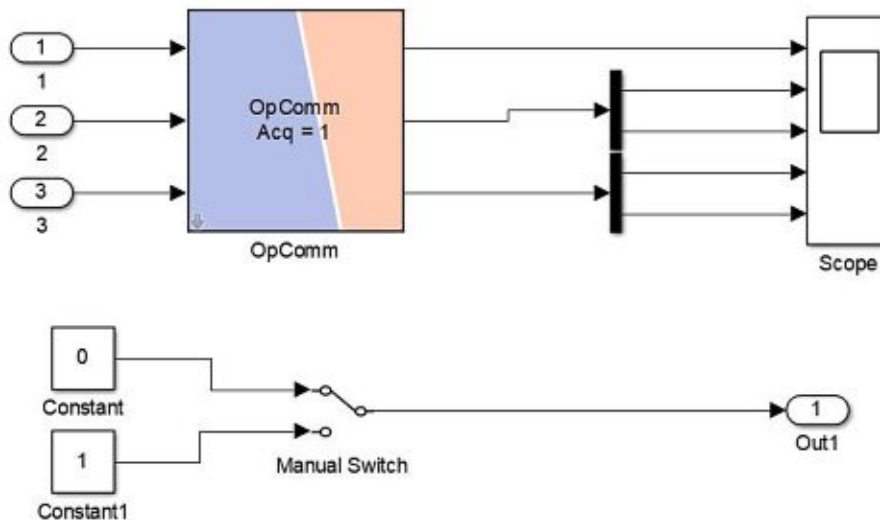


Abbildung 5.2: Subsystem *SC_GUI*

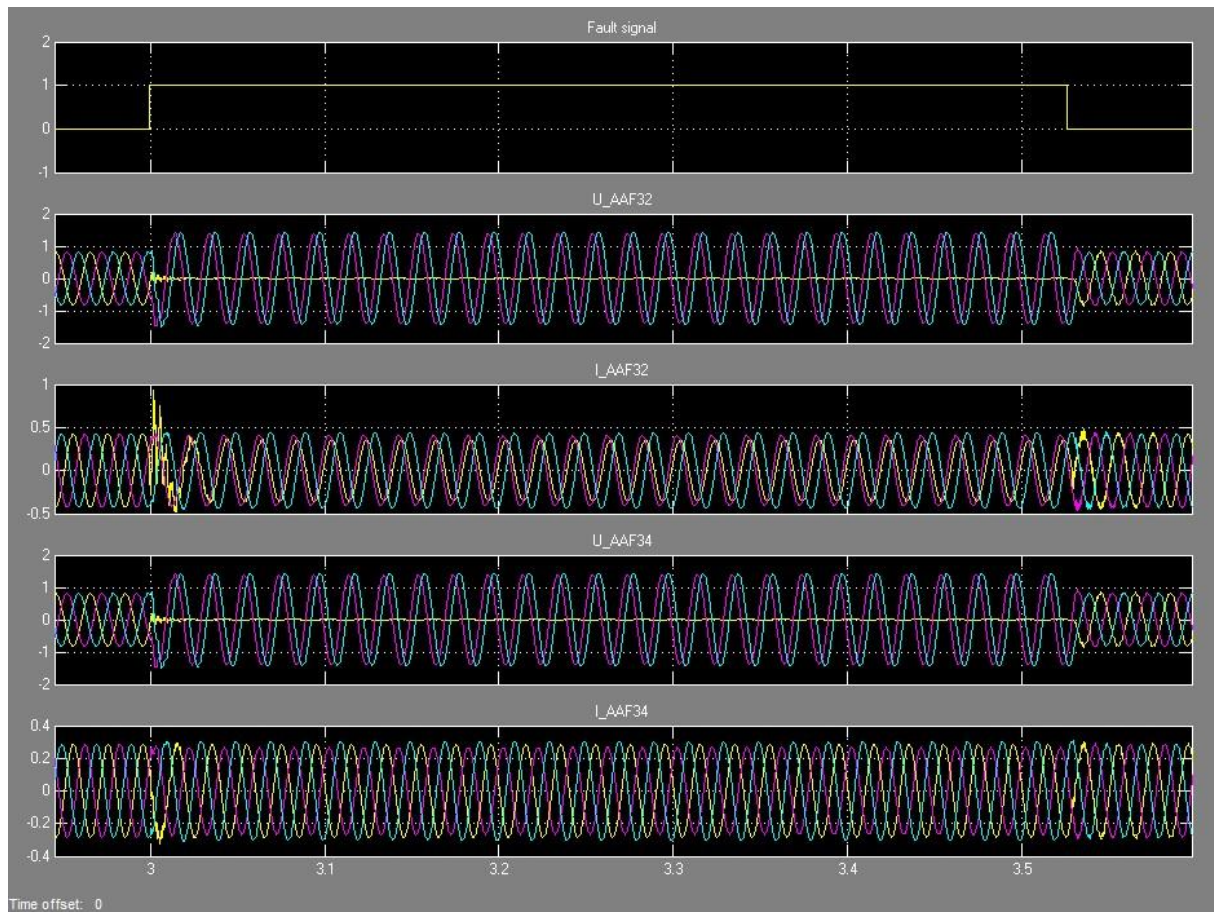


Abbildung 5.3: Fehlersimulation

Die Darstellung der Signale während einer Simulation als Scope-Ansicht ist in Abbildung 5.3 sichtbar. Wie in dieser Abbildung erkennbar ist, erfolgt eine einpolige Fehlersimulation von Zeitpunkt 3 s, als das Aktivierungssignal des Fehlers (Fault signal) auf 1 geschaltet wird, bis 3,54 s, als es auf 0 zurückgeschaltet. Diese Signale werden auch auf die analogen Ausgangskanäle ausgeführt. Sie kann durch ein Oszilloskop mit Hilfe der Mini-BNC-Terminals beobachtet werden (siehe Abschnitt 4.2.2). Im nächsten Kapitel wird die Nummerierung der Kanäle genauer betrachtet.

Zu einer erfolgreichen Real-Time-Simulation müssen noch einige Simulationsparameter unter Menüpunkt *Model Configuration Parameters* eingestellt werden. Neben Simulations- und Samplezeit empfiehlt die Firma OPAL-RT die Ausschaltung von Funktionen *Block reduction* und *Signal storage reuse* wegen der Vermeidung des Datenverlustes.

5.2 Aufbau des Messsystems

Ein wesentlicher Punkt des Projektes ist der hardwareseitige Aufbau des Messsystems. Die verwendeten Geräte und Programme wurden schon in dem Kapitel 4 vorgestellt. Hier wird aber auf ihre Verbindung näher eingegangen. Die Abbildung 5.4 gibt einen Überblick über diese Verbindungen. Die Nummern und die Beziehungen der Kanäle sind im Anhang C.: zusammengefasst.

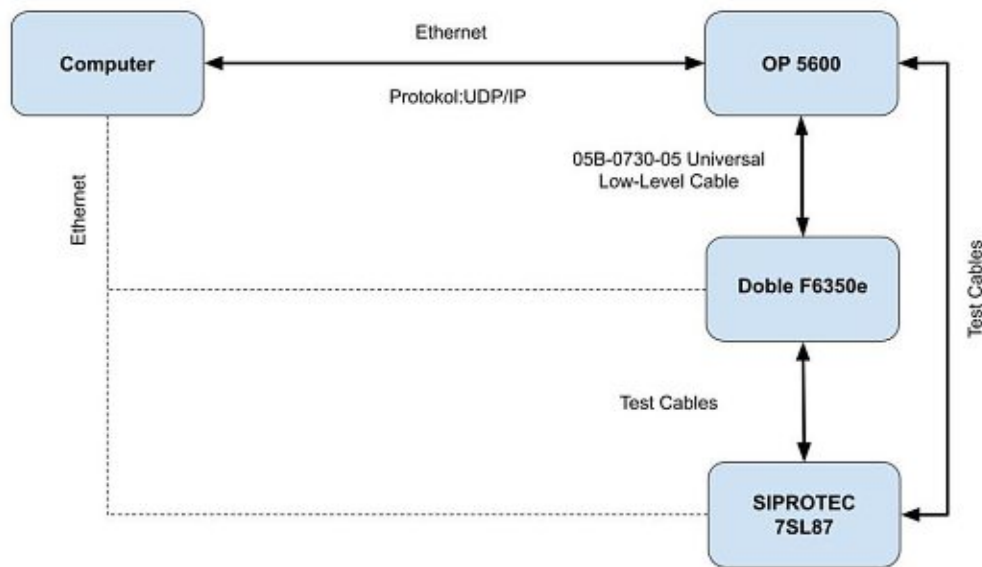


Abbildung 5.4: Schematische Darstellung von Messsystem

Die Kommunikation zwischen dem Computer und den anderen Geräten erfolgt durch ein Ethernet-Netz. Aber solange eine ständige Interaktion zwischen dem Host-PC und dem Echtzeitsimulator während der Simulation entsteht, wird der Computer vor der Simulation nur für die Einstellung oder Konfigurierung der anderen zwei verwendet.

Der Simulator OP5600 schickt die analogen Signale durch ein Low-Level-Kabel zu dem Verstärker. Eines der beiden Enden des verwendeten Kabels wird für den Verstärker mit einem DB-15-Anschluss ausgestattet. Am anderen Ende werden die Fasern separat ausgeführt und mit Aderendhülle geschlossen. Damit diese Fasern an dem Simulator angeschlossen werden können, wird eine zusätzliche Platine benutzt. Auf dieser Platine werden ein DB-37-Anschluss integriert, dessen Pins mit Anschlussklemmen verbunden sind. In dieser Art und Weise ist es möglich die entsprechenden Ein- und Ausgänge miteinander zu verbinden.

Damit die Reaktion der Schutzrelais gemessen werden kann, wird ein digitaler Ausgang beider Schutzrelais verwendet. Deswegen wird eine Leitung von der 5 V- Spannungsquelle des Simulators in der Buchse B13 beider Apparate und eine andere von Buchse 14 in zwei digitalen Eingängen von OP5600 geführt. Die verstärkte Spannung und Strom wird durch 12 einzelne Drähte zwischen dem Verstärker und dem Schutzrelais geleitet. Noch eines ist für die Erdungspunkt verantwortlich.

Dieses Messsystem wurde an der Technischen Universität Wien im Schutzlabor zusammengestellt.

5.3 Lösung der Simulationsautomatisierung

Das Hauptziel dieser Arbeit ist vorzustellen, wie die Vorteile und Potentiale der Real-Time-Technologie ausgeschöpft werden können. Als ein solcher Vorteil gilt es, dass die Reaktion eines realen Schutzrelais in Echtzeit gemessen werden kann. Außerdem können Simulationen wesentlich schneller durchgeführt werden als bei einer Offline-Simulation mit langer Berechnungszeit.

Aber dieser zweite Vorteil kann nicht ausgenutzt werden, wenn man Simulationen mit unterschiedlichen Parametern umsetzen möchte. In diesem Fall muss man die Simulation stoppen, die bestimmten Werten im Simulink umstellen, das Modell erneut kompilieren lassen, in Simulator hochladen und die Simulation wieder starten. Besonders die Kompilierung und die Hochladung des Modells sind zeitaufwendig. Sie verursachen ca. 2-3 min Totzeit per Simulation, die bei einer Iterationszahl mit hunderten Simulationen als sehr viel gilt.

Damit diese Totzeit in diesem Projekt vermieden werden kann, mussten im Windparkmodell einige Änderungen vorgenommen werden. Sie werden in den nächsten Kapiteln ausgeführt. Die Grundidee, mit der dieses Problem gelöst werden kann, liegt darin, dass mehrere Komponenten (z.B.: Three-Phase Fault Block) mit unterschiedlichen Parametern parallel im Modell eingebaut werden. So können sie gleichzeitig kompiliert und hochgeladen aber voneinander separiert aktiviert werden.

In diesem Projekt wird untersucht, wie der Lichtbogenwiderstand, der Phasenwinkel im Zeitpunkt des Kurzschlusses und die Entfernung von Einspeisung den Ablauf des Fehles beeinflusst. In jedem Fall wird ein einpoliger Kurzschluss simuliert.

5.3.1 Lichtbogenwiderstad

Der Widerstand des Lichtbogens ist einer der Parameter, der den Kurzschlussstrom am deutlichsten beeinflusst. Wie schon im Kapitel 2.2.3 erklärt wurde, ändert sich der Wert des Lichtbogenwiderstandes während des Kurzschlussfalls nicht-linear. Er beträgt in Mittelspannungsnetz einige Ohm.

Für Lichtbogenwiderstand werden fünf unterschiedliche Werte zwischen 0 und 4 Ohm eingestellt. Dieser Wert kann unter inneren Parametern von *Three-Phase Fault* Block umgestellt werden. Deswegen müssen fünf solche Blöcke parallel im Modell eingebaut, und in einem Subsystem (Name: Fault) eingeordnet werden, wie es in der Abbildung 5.6 sichtbar ist. Alle Blöcke haben ein externes Aktivierungssignal, mit dem immer nur der vorgeschriebene Fehler eingesetzt wird. Diese Signale werden durch Eingangsport 2 in das Subsystem eingeführt. Der Schalter (switch) wechselt nach Schaltsignal (input 1) zwischen den verknüpften Aktivierungssignale und einem Konstant mit Wert 0. Im ersten Fall muss mindestens ein Signal gleich 1 sein, damit ein Fehlerfall erfolgt. Im anderen Fall wird zu allen Ports Null übermittelt und alle *Three-Phase Fault* Blöcke bleiben inaktiv.

Der Anschluss der drei Phasen mit den Leitungen erfolgt durch drei *Physical Modeling Connection* Ports.

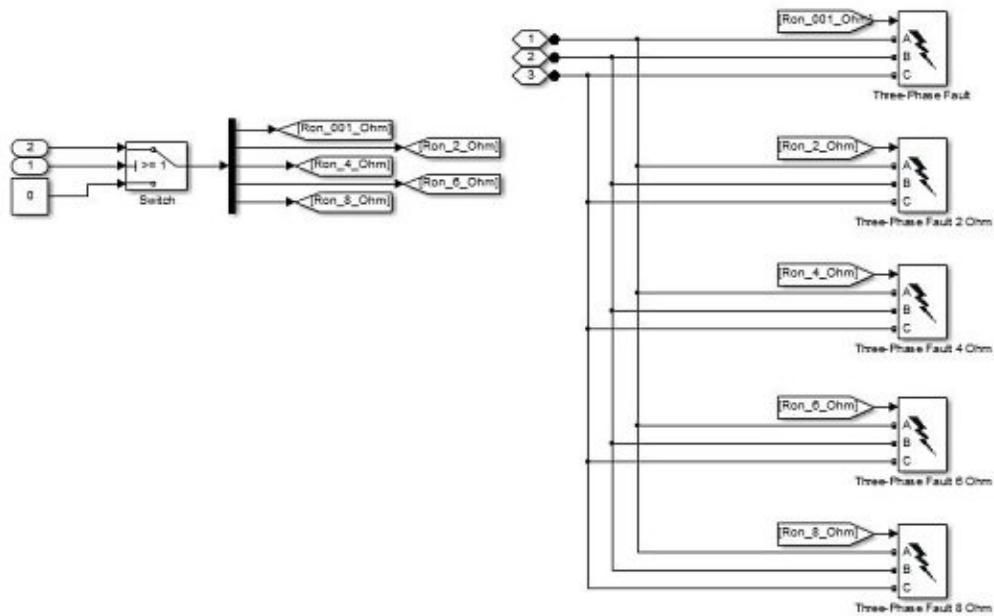


Abbildung 5.6: Fault-Subsystem

Ein Fehler wird in folgender Weise modelliert (siehe Abbildung 5.5):

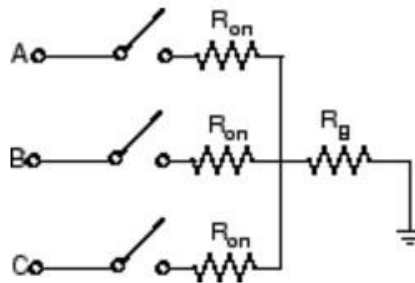


Abbildung 5.5: Fehlermodell

Neben den drei Schaltern, die die Phasen kurzschließen, befinden sich zwei Widerstände. R_g und R_{on} repräsentiert den Boden- und den Lichtbogenwiderstand. Die während der Simulation eingestellten Parameter sind:

- Fehlerart: Phase A - Ground Fault
- R_{on} = Variable (0,001 Ω – 8 Ω)
- R_g = 0,001 Ω
- Snubbers Widerstand – R_p = 1 M Ω
- Snubbers Kapazität – C_p = inf

Zur Auswahl und Aktivierung des entsprechenden *Three-Phase Fault* Blocks werden fünf manuelle Schalter (Manual Switch Block) genutzt, die sich zwischen 0 und 1 wechseln. Sie müssen sich im Subsystem *SC_GUI* befinden, um während der Simulation betätigen zu können.

5.3.2 Phasenwinkel

In der Abbildung 5.7 sind der Spannungs- (oben) und Stromverlauf (unten) von drei Fehlerfällen bemerkbar. Alle drei Fehler sind einpolige Erdschlüsse in gleichem Netz. Der Unterschied liegt nur in dem Phasenwinkel des Stromes in Phase L1 im Zeitpunkt des Erdschlusses in folgender Weise: (a): $\alpha = 0^\circ$; (b): $\alpha = 30^\circ$; (c): $\alpha = 90^\circ$.

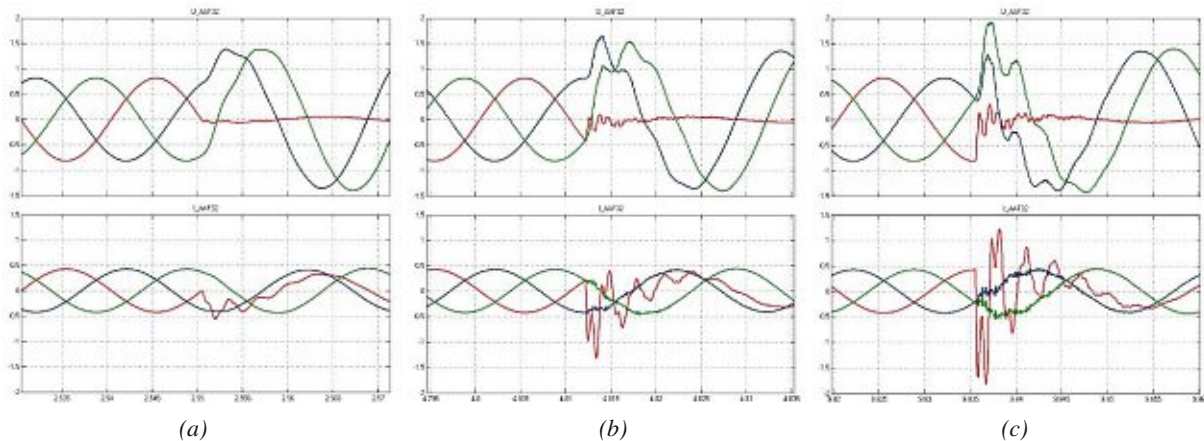


Abbildung 5.7: Kurzschlussverläufe mit unterschiedlichen Phasenwinkeln

Wie es in den Diagrammen gut sichtbar ist, je größer der Winkel zwischen 0° und 90° ist desto größer ist der Anfangskurzschlussstrom. Diese Stromerhöhung kann einen Einfluss auf das Verhalten des Schutzrelais haben, deshalb ist es wichtig diesen Parameter während der Messung ebenfalls zu untersuchen. Damit der Fehler während der Simulation nur bei dem bestimmten Bestand aktiviert wird, muss eine neue Funktion in das Modell eingebaut werden, mit der der momentane Winkel des Zeigers des Stromes (oder der Spannung) stets beobachtet und bestimmt werden kann.

Zu diesem Ziel steht der sogenannte *Complex to Magnitude and Angle* Block zur Verfügung. Wie der Name schon sagt, zerlegt er einen komplexen Wert in zwei Skalare: einerseits den Betrag, andererseits den mit der reellen Achse eingeschlossenen Winkel des Vektors. Jedoch kann man beobachten, dass diese Funktion verbunden mit Spannungs- oder Strommessung eines AC-Netzes bei *discrete step* Lösung nicht korrekt funktioniert.

Als Anhang D.: sind zwei Abbildungen sichtbar. Die rechte Seite der Abbildung A stellt dar, wie der Signalverlauf eines rotierenden Zeigers aussehen sollte: ein konstanter Betrag (Magnitude) und zwischen $-\pi \dots \pi$ geänderter Winkel (Angle). Mit Hilfe von dem Block *Complex to Real- Imag* wird der reelle und imaginäre Teil dieses rotierenden Zeigers ebenfalls dargestellt. In der Abbildung B wird ein Modell eines einfachen Systems mit einer AC-Stromquelle, einer Last und einer Spannungsmesseinheit dargestellt. Wie es in der rechten Seite der Abbildung erkennbar ist, wird der Vektor der momentanen Spannung nicht in erwarteter Weise gebildet. Um dieses Problem vermeiden zu können, muss er aus einer anderen Richtung annähert werden.

Eine komplexe Zahl (oder Zeiger) kann durch ihren Betrag und ihren Phasenwinkel oder auch durch ihren Real- und Imaginärteil dargestellt werden. [1]

$$\underline{Z} = Z[\cos\varphi + j\sin\varphi] = \operatorname{Re}\{\underline{Z}\} + j\operatorname{Im}\{\underline{Z}\} \quad (5.1)$$

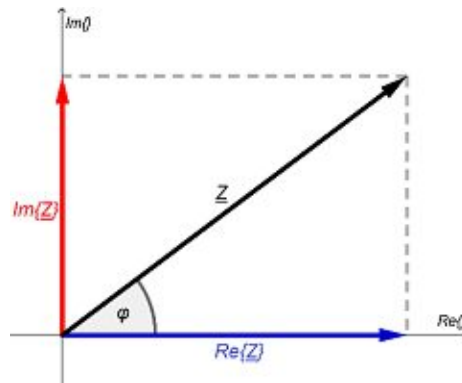


Abbildung 5.8: Darstellung des Zeigermodelles

Mit Hilfe dieser Beschreibung können sinusförmige Wechselgrößen (Spannung, Strom) ebenfalls beschrieben werden. Der momentane Wert dieser Wechselgrößen ist gleich mit dem reellen Teil des Zeigers, der während der Simulation gemessen werden kann. Um diesen Vektor richtig bestimmen zu können, muss sein auf dem reellen Teil senkrecht imaginärer Teil auch bestimmt werden.

In einem dreiphasigen System sind die Zeiger der Wechselgrößen im normalen Zustand um jeweils 120° versetzt angeordnet. Nach diesem Satz kann der Betrag von Phase b und c genutzt werden, um einen mit dem imaginären Teil von Phase a identischen Wert zu bestimmen. Da während der Simulation nur Skalare gemessen werden können, kann nur der reale Teil dieses Vektors verwendet werden. Wenn die Magnitude gleich Eins angenommen wird, können die gemessenen Werte in folgender Weise aufgeschrieben werden:

- $V_a = \operatorname{Re}\{\underline{V}_a\} = \cos(\alpha)$
- $V_b = \operatorname{Re}\{\underline{V}_b\} = \cos(\alpha - 120^\circ)$
- $V_c = \operatorname{Re}\{\underline{V}_c\} = \cos(\alpha - 240^\circ)$

$$\cos(x) - \cos(y) = -2 \cdot \sin\left(\frac{x+y}{2}\right) \cdot \sin\left(\frac{x-y}{2}\right) \quad (5.2)$$

Mit Verwendung der trigonometrischen Formel (5.2) kann der Unterschied $V_b - V_c$ bestimmt werden:

$$\cos(\alpha - 120^\circ) - \cos(\alpha - 240^\circ) = -\sqrt{3} \cdot \sin(\alpha - 180^\circ) \quad (5.3)$$

$$\sin(x \pm y) = \sin(x) \cos(y) \pm \sin(y) \cos(x) \quad (5.4)$$

$$\sin(\alpha - 180^\circ) = \sin(\alpha) \cos(180^\circ) - \sin(180^\circ) \cos(\alpha) = -\sin(\alpha) \quad (5.5)$$

$$V_b - V_a = \sqrt{3} \cdot \sin(\alpha) \quad (5.6)$$

Nach der Herleitung der Gleichung (5.3) gemäß Formel (5.4) ist das Ergebnis in Gleichung (5.6) genau Wurzel von 3 – mal größer als der imaginäre Teil von Phase a (siehe Formel (5.1)).

Gemäß diesen theoretischen Grundlagen wurde die folgende Funktionalität hergestellt, die sich im Subsystem *Angle_Measurement* befindet:

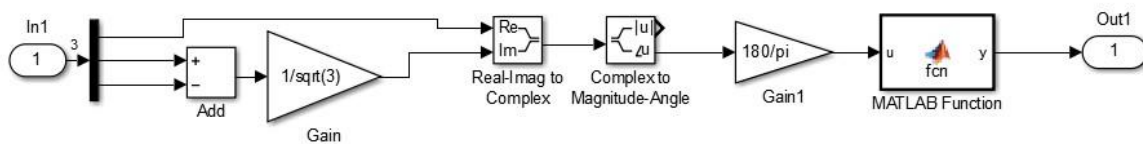


Abbildung 5.9: Subsystem *Angle_Measurement*

Als Inputsignal gelten die gemessenen Spannungswerte (oder Stromwerte) von AAF32. Wie in der Abbildung 5.9 sichtbar ist, wird eine komplexe Zahl mit Hilfe des Blockes *Real-Imag to Complex* gebildet, deren Betrag und Winkel mit dem nächsten Block (*Complex to Magnitude-Angle*) bestimmt werden. Da der Block *Complex to Magnitude-Angle* den Winkel zwischen $-\pi \dots \pi$ als Radiant definiert, wird dieser Wert mit einer Gain-Funktion in Grad umgeformt. Zu einer besseren Nutzbarkeit wird dieser Bereich mit der MATLAB-Funktion von $-180^\circ \dots 180^\circ$ in $0 \dots 360^\circ$ verschoben.

Als nächster Schritt muss der aktuelle Phasenwinkel mit einem eingestellten Wert vereinbart werden. Diese Aufgabe wird im Subsystem *Switch* durchgeführt, der in der Abbildung 5.10 dargestellt wird.

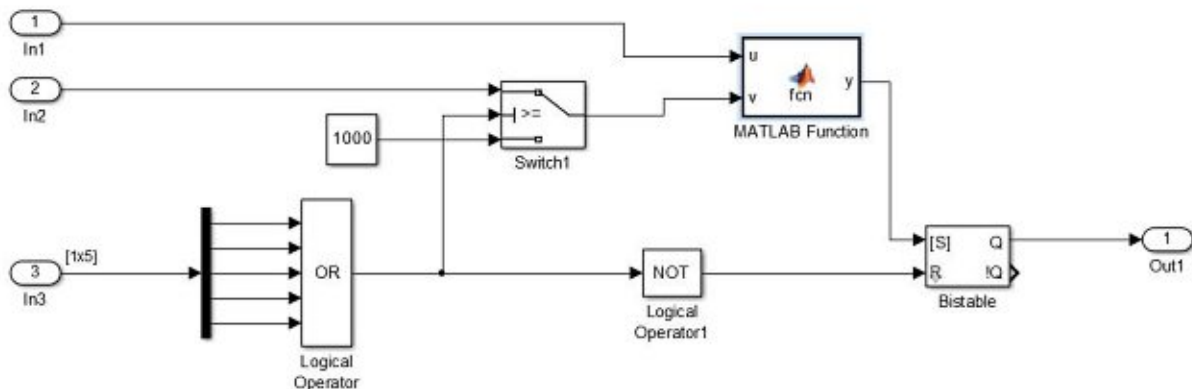


Abbildung 5.10: Subsystem *Switch*

Das Schlüsselement dieser Funktionalität ist der MATLAB-Funktionsblock, der die rechtzeitige Einschaltung des Blockes *Three-Phase Fault* ermöglicht. Sie hat zwei Eingangsvariablen und einen Ausgang. Durch den ersten Eingang (u) wird der gemessene Phasenwinkel definiert. Durch den zweiten (v) wird ein eingestellter Wert eingelesen, der als der gewählte Winkel zum Fehler gilt. Die MATLAB-Funktion vergleicht diese zwei Werte und bei einer Übereinstimmung stellt ihren Ausgangswert von 0 auf 1 um. Es ist das Startsignal des Fehlers. Um den Block *Three-Phase Fault* nicht nur für einen Moment einzuschalten, muss ein bistabiler Block ins System integriert werden.

Wegen der diskreten Lösungsmethode der Simulation können die zwei Variablen niemals identisch sein. Deswegen muss ein Toleranzband eingestellt werden. Zur Bestimmung des Umfanges dieses Toleranzbandes ist es notwendig das Zeitintervall eines Grades bei einer 50 Hz sinusförmigen Spannungsverlauf zu bestimmen:

$$\tau = \frac{1}{360f} = 55,556 \mu s \quad (5.7)$$

Es ist sichtbar, dass dieses Zeitintervall kleiner ist als die eingestellte Samplezeit $T_s = 100 \mu s$. Deshalb muss eine 2° große Zeitspanne eingestellt werden, um mindestens eine Stichprobe während dieser Zeit ziehen zu können.

Durch Eingang 3 werden die Aktivierungssignale der Fehler-Blöcke eingeführt. Wenn mindestens ein Signalwert gleich Eins ist, schaltet *Switch1* von der Konstante auf Eingangssignal 2 um, das den ausgewählten Phasenwinkel übermittelt. Als Konstante kann ein Betrag eingestellt werden, der außerhalb des Bereiches $0^\circ \dots 360^\circ$ liegt. In diesem Fall ist dieser Wert gleich 1000.

Wenn alle Eingangssignale des logischen Operator *OR* gleich Null sind, wird das Ausgangssignal ebenfalls gleich Null. Mit dem logischen Operator *NOT* wird dieser logische Wert von FALSCH (0) auf WAHR (1) umgestellt, der das Reset-Signal dem bistabilen Block gibt. Also die Fehlerfunktion wird ausgeschaltet.

5.3.3 Fehlerort

Da der Simulink keine Möglichkeit gibt einen Fehler auf einen bestimmten Punkt einer Leitungssektion aufzusetzen, kann ein Fehler nur zwischen Elementen integriert werden. Zur Änderung der Position des Erdschlusses sind zwei Möglichkeiten vorhanden.

Erstens können mehrere Leitungsstücke parallel im Modell so eingebaut werden, dass eine Leitungssektion (Three-Phase PI Section Line) auf zwei Stücke mit gleichen Parametern (r, l, c) aber mit unterschiedlichen Längen aufgeteilt wird. Mit diesen Längen kann eingestellt werden, wo die Position des Fehlers sein muss. Damit die parallelen Leitungen nicht gleichzeitig bestromt werden, müssen Leistungsschalter vor und nach ihnen eingesetzt werden. Mit der Öffnung oder dem Schließen dieses Schalters kann immer nur die Leitung mit dem entsprechenden Verhältnis aktiviert werden. Der schematische Aufbau dieser Methode wird in der Abbildung 5.11 (a) dargestellt.

In der zweiten Möglichkeit wird die Leitung auf 10 Stücke aufgeteilt und sie werden nacheinander in Reihe geschaltet. Jedes Stück repräsentiert 10% der gesamten Strecke und zwischen sie kann ein Fehlerblock durch einen Schalter hinzugefügt werden. Die grobe Darstellung dieser Methode ist in der Abbildung 5.11 (b) sichtbar.

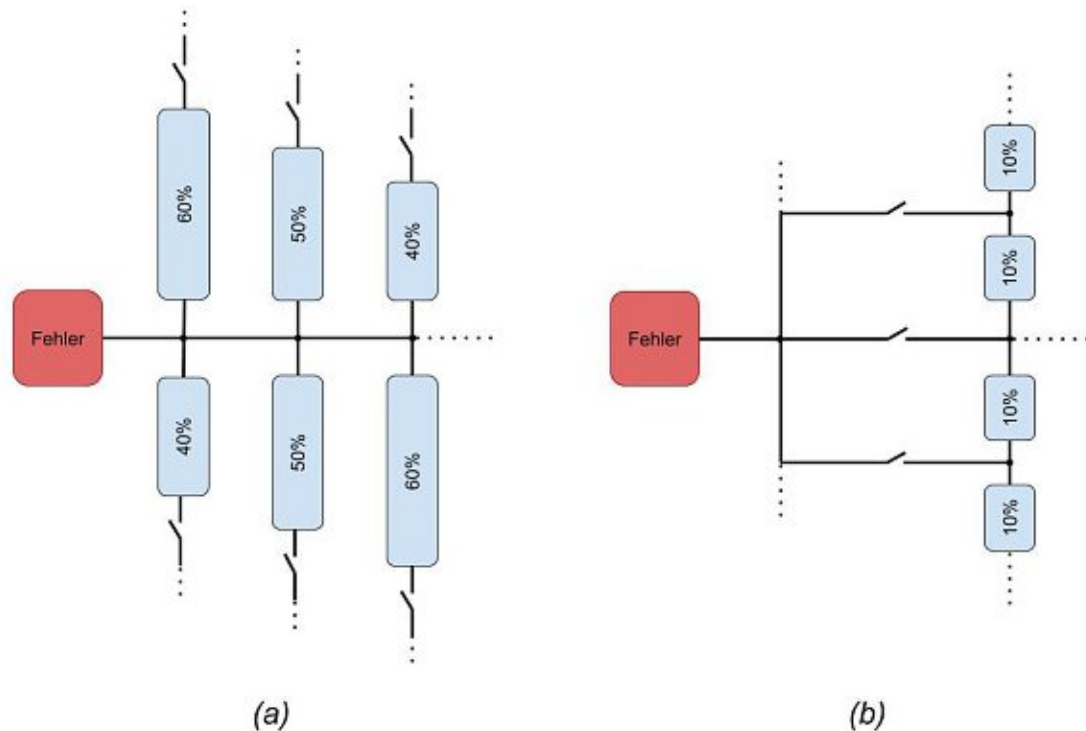


Abbildung 5.11: Möglichkeiten der Fehlerpositionierung

In diesem Projekt wird von den beiden die zweite Lösung gewählt, da sie einerseits eine deutlich einfachere Struktur hat. Zweites, der Schalter zwischen der Stromquelle und der Leitung kann zu einem Simulationsproblem im Simulink führen. Ähnliches Problem taucht während der Implementierung des Schutzrelais ins Modell auf (siehe Abschnitt 5.3.4).

Die zweite Herausforderung mit dieser Funktion ist die Kontrolle der insgesamt 11 Schalter, mit denen immer die richtige Position des Erdschlusses ausgewählt wird. Dafür gibt es zwei Möglichkeiten. Die erste ist die gleiche Lösung wie bei der Aktivierung der Fehlerblöcke. Zu jedem Block gehört genau ein manueller Schalter, der zwischen 0 und 1 schaltet und sich im Subsystem *SC_GUI* befindet.

Die zweite Möglichkeit, die am Ende verwendet wird, ist die Erstellung eines MATLAB-Funktionsblocks. Das Eingangssignal ergibt sich aus der Multiplikation eines Konstantes 1 mit einem durch einen *Gain* Block eingestellten Wert. Im MATLAB-Funktionsblock wird ein Vektor mit elf Elementen hergestellt. Am Programanfang werden alle Elemente auf 0 gestellt, dann wird der Wert in bestimmter Position nach dem Eingangssignal von 0 auf 1 ausgetauscht. Dieser Ausgangsvektor wird vor den Schaltern mit Hilfe einem *Demux*-Block auf elf Einzelsignale aufgeteilt, die zu den entsprechenden Schaltern zugeführt werden. Die Blöcke *Constant*, *Gain* und *MATLAB Function* werden wegen der besseren Übersichtbarkeit in einem

Subsystem eingesetzt. Es wird mit einer Maske ausgestattet, die die Auswahl des Multiplikators des *Gain* Blocks vereinfacht. Das einzige Problem mit dieser Methode ist, dass der *OpComm*-Block den Vektor als Signaltyp nicht unterstützt und eine Fehlermeldung bei dem Simulationsstart auftritt. Deshalb muss der MATLAB-Funktionsblock im Subsystem *SM_Master* umgesetzt werden.

5.3.4 Reaktion des Schutzrelais

Um das Verhalten des Windparks genau simulieren und untersuchen zu können, müssen Leistungsschalter dem System hinzugefügt werden, die sich wie die realen Geräte verhalten. Mit Hilfe von OP5600 kann die Reaktion des Schutzrelais von Siemens gemessen werden. Im Modell werden die zwei Leistungsschalter implementiert, die für die Problemstellung des Projekts relevant sind (siehe Abschnitt 1.1).

Die Schalter (Three-Phase Breaker) werden also in die ersten zwei Linien zwischen Abzweigung und Messpunkt (Three-Phase VI Measurement) installiert. Die wichtigsten eingestellten Parameter sind folgende:

- Schalterwiderstand (Breakers resistance): $R_{on} = 0.01 \Omega$
- Dämpfungswiderstand (Snubbers resistance): $R_p = \infty$
- Dämpfungskapazität (Snubbers capacitance) $C_p = \infty$

Außerdem wird der Zeitpunkt des Umschaltens durch ein äußeres Signal (External control of switching times) definiert. Bei 0 ist der Schalter offen, bei 1 geschlossen. Das Schaltsignal des Schutzrelais wird durch zwei binäre Eingänge detektiert. Deshalb ist es noch nötig einen weiteren Block *OpCtrl DigitalIn* (Port Nr.:3) in das Modell einzubauen. Da der Schalter des verwendeten digitalen Ausganges ein Schließer ist, ist das Eingangssignal in dem normalen Betriebszustand null. Deswegen muss das Signal nach dem Block *OpCtrl DigitalIn* durch einen *NOT Logical Operator* durchgeführt werden, (weil der Leistungsschalter bei einem Eingangssignal 1 geschlossen ist.)

Das Schutzrelais wird so eingestellt, dass seine digitalen Ausgänge bei einer Anregung impulsartig umschalten. Damit die Leistungsschalter im Modell bei einem Fehlerfall bis einen bestimmten Zeitpunkt offen bleiben, muss ein bistabiler Funktionsblock hinzugefügt werden. Die Zurücksetzung dieses Funktionsbockes erfolgt durch einen manuellen Schalter.

In diesem Zustand des Modelles konnte das folgende Phänomen beobachtet werden: Wenn der Schalter geöffnet wird, ist das Spannungsniveau hochgesprungen. Wegen dieser Überspannung folgte der Wiedereinschaltung des Zweiges ein Stoßstrom, der 20-fach größer als der Nennstrom war. Der Grund für dieses Phänomen ist, dass die dreiphasige Einspeisung der Windkraftanlagen durch drei Stromquellen ersetzt wird. Diese Quellen versuchen nach der Abbrechung der Linie ins System weiterhin zu speisen. Dieses Spannungs- und Stromniveau war dem Verstärker zu hoch, der sich deswegen während der Simulation abgeschaltet hat.

Um diese Spannungs- und Stromerhöhung zu vermeiden, müssen die Stromquellen gleichzeitig mit der Öffnung des Leistungsschalters abgeschaltet werden. Dazu gibt es zwei Möglichkeiten. Die erste Option ist die Ersetzung eines weiteren Schalters zwischen den Quellen und dem restlichen Teil der Leitung. Diese Methode macht das System aber deutlich komplizierter, da Simulink in diesem Fall die Erdung des Sternpunktes fordert. Und das abgeschaltete Leitungsstück müsste auch geerdet werden.

Die andere Möglichkeit ist der Austausch der einfachen Stromquellen durch eine kontrollierte Stromquelle (Controlled Current Source). Dieser Block wandelt das Simulink-Eingangssignal in eine äquivalente Stromquelle um. Dazu braucht man noch einen Block, der das erforderliche Sinussignal (Sine wave) generiert und einen Schalter, der abhängig vom Eingangssignal des Schutzrelais zwischen der Sinussignal und null umschaltet. Die alte (a) und die neue (b) Version des Subsystems Array 32 (gleiche wie Array 34), das die Stromquellen beinhaltet, wird in der Abbildung 5.12 dargestellt.

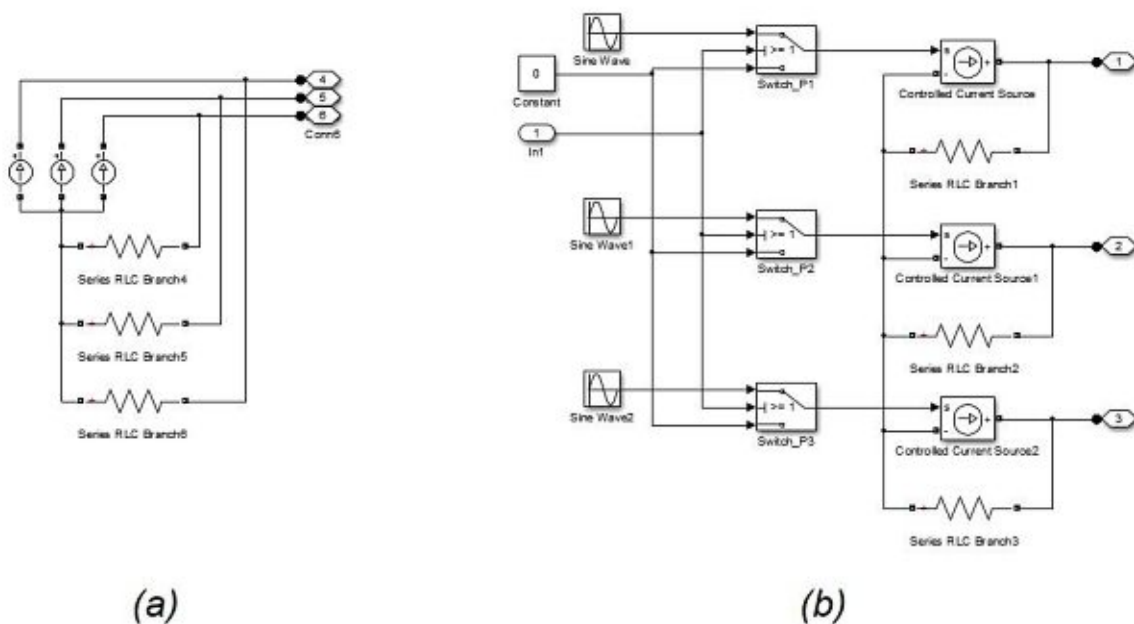


Abbildung 5.12: Subsystem Array 32

Das Modell nach den Änderungen, die in den Abschnitten 5.3.1-5.3.4 beschrieben sind, ist in Anhang F.: sichtbar. Mit diesen Umsetzungen können Simulationen nacheinander ohne erneute Kompilierung durchgeführt werden. Aber in diesem Fall müssen die Parameter manuell umgestellt werden. Um die manuelle Steuerung mit einer automatisierten Simulationsdurchführung auszutauschen, kann ein Python-Skript verwendet werden. Der nächste Abschnitt beschäftigt sich mit dieser Möglichkeit.

5.3.5 Automatisierung mit Python-Skript

Wie es schon im Kapitel 4.2.1 erwähnt wurde, kann eine Simulation nicht nur durch den Subsystem *SC_GUI* manuell, sondern auch durch eine Programmierschnittstelle (API - application programming interface) automatisch kontrolliert werden. Zu diesem Zweck beinhaltet das Programm RT-LAB einen eingebauten Python-Editor. Mit Hilfe des Python-Skriptes können bestimmte MATLAB Variablen und Signalen während der Simulation geändert werden. Diese Variablen können im Fenster *Project Explorer* ausgesucht und ebenfalls mit *Variable Viewer* überschrieben werden.

Man kann im Fenster *Project Explorer* ein neues File seinem Projekt hinzufügen. Man muss bei der Eingabe des Filenames darauf achten, die Dateinamenserweiterung (.py) ebenfalls einzugeben. Nach dieser Vorbereitung kann das Skript im Fenster *Editor* geöffnet und bearbeitet werden. Das Ablaufdiagramm von dem verwendeten Python-Skript wird in der Abbildung 5.13 dargestellt. Der vollständige Code befindet sich im Anhang G.:

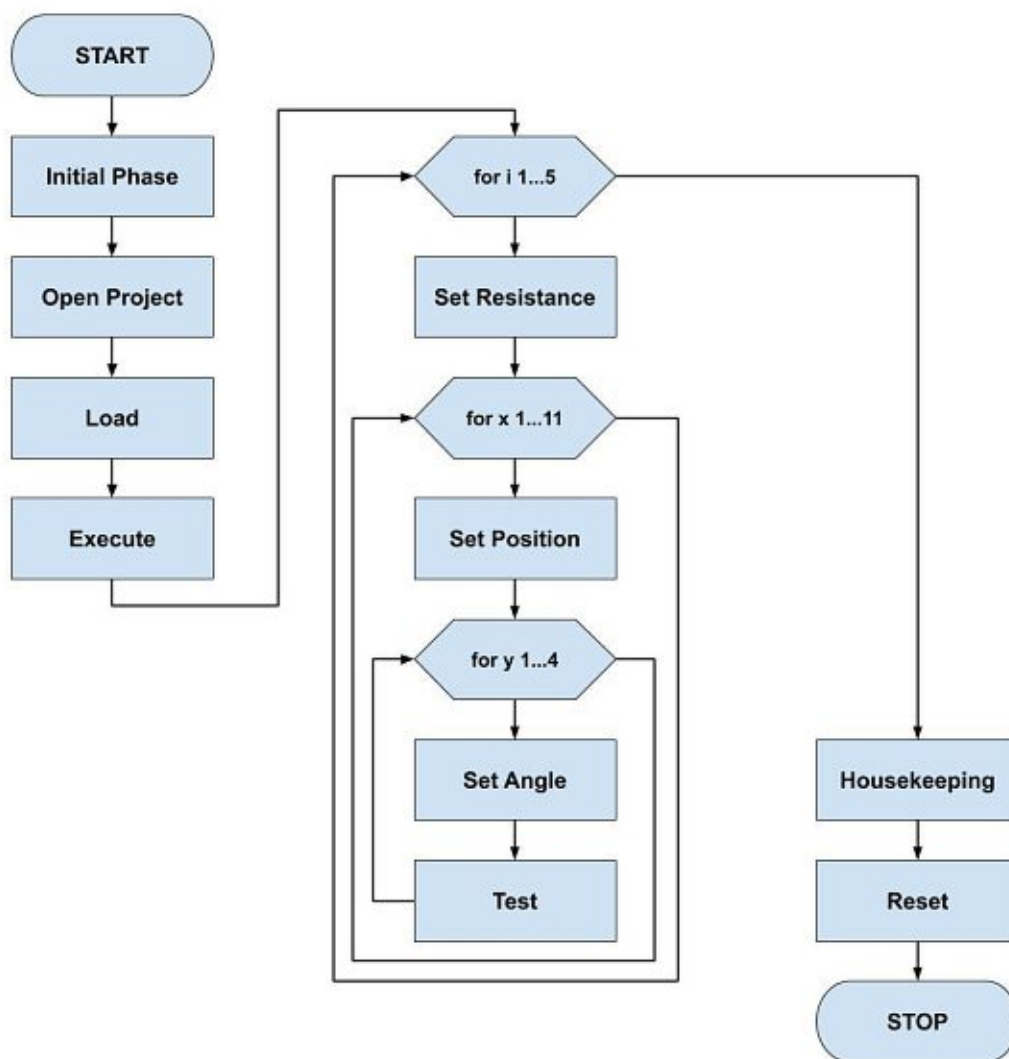


Abbildung 5.13: Ablaufdiagramm des Python-Skripts

Der Kern des Skripts sind drei ineinander eingebettete Zählschleifen (for loop). In jedem Zyklus wird eins von den drei Variablen neu definiert. Wenn man für bestimmte Parameter die Fehlersimulation mehrmals wiederholen möchte, ist es möglich durch das Hinzufügen einer weiteren Zählschleife.

Am Anfang des Skripts werden vier Bibliotheken eingefügt. Die erste Bibliothek **RtlabApi** ist die wichtigste von den vier. Sie beinhaltet die Funktionsaufrufe von RT-LAB. Die detaillierte Beschreibung dieser Funktionen befindet sich im Hilfeinhalt des Programmes unter Menüpunkt *RT-LAB Python API Reference Guide*. In diesem Abschnitt werden nur die im Projekt verwendeten Funktionen betrachtet.

- **RtlabApi.OpenProject(projectName):** Mit dieser Funktion kann ein vordefiniertes Projekt geöffnet werden. Als *projectName* muss man den Pfad des Files mit Dateinamenserweiterung (.llp) in dem Ordner des Projektes als Variable *string* eingeben.
- **RtlabApi.Load(realTimeMode, timeFactor):** Es ladet die Simulation. Der Eingangswert *realTimeMode* definiert die Art der Simulationsdurchführung. Es kann durch Hardware synchronisiert (RtlabApi.HARD_SYNC_MODE), durch Software synchronisiert (RtlabApi.SOFT_SIM_MODE) oder so schnell wie möglich (RtlabApi.SIM_MODE) durchgeführt werden. Der *timeFactor* bestimmt die Geschwindigkeit der Simulation. Bei einer Real-Time-Simulation ist dieser Wert gleich 1.
- **RtlabApi.Execute(timeFactor):** Es lässt die Simulation durchführen. Der *timeFactor* ist der gleiche Wert wie bei der Load-Funktion.
- **RtlabApi.Reset():** Es stoppt die aktuelle Simulation und setzt sie zurück.
- **RtlabApi.CloseProject():** Es schließt das Projekt.

Die obigen Funktionen werden zum Start und Stopp einer Simulation benutzt. Um während der Simulation bestimmte Parameter ändern zu können, muss die Parametersteuerung erlaubt werden. Dazu können die folgenden Funktionen verwendet werden:

- **RtlabApi.GetParameterControl(Control):** zur Steuerung innerer Parameter bestimmter Elemente wie z.B.: innerer Wert eines Gain-Blocks.
- **RtlabApi.GetSignalControl(Control):** zur Bestimmung des Wertes von Kontrollsignalen.
- **RtlabApi.GetAcquisitionControl(Control):** Diese Funktion ist notwendig, wenn man die Messdaten speichern möchte.

Der Parameter *Control* kann wahr (1) oder falsch (0) sein. Vor der Umstellung eines Parameters (oder Signals) muss die Steuerung also angefragt werden (*Control* – 1) und am Ende der Steuerung muss sie freigegeben werden (*Control* – 0). Die Umstellung des Wertes erfolgt mit Hilfe der folgenden Funktionen:

- **RtlabApi.SetParametersByName (parameterNames, parameterValues)**
- **RtlabApi.SetSignalsByName(signalNames, signalValues)**

Die Eingangswerte *parameterNames* und *signalNames* sind durch das Programm RT-LAB generierte Pfade, die aus dem PARAM-File (.param) und SIGNAL-File (.signal) geschöpft werden können. Zur Verbesserung der Übersicht werden Variablen statt dieser Pfade verwendet, die in Initialphase definiert wurden. Als *parameterValues* und *signalValues* können die gewählten neuen Werte eingestellt werden. Mit dieser Methode können die Parameter der Fehler (Position, Winkel, Lichtbogenwiderstand) umgestellt und die Steuerung der Leistungsschalter realisiert werden.

Die letzte Funktion dieser Bibliothek ist

- **RtlibApi.SetAcqWriteFile(acqGroup, fileName, varName, enabled),**

womit die Signalwerte einer Acquisition-Gruppe gespeichert werden können. Sie sind grundsätzlich die Werte der Signale, die durch das Block *OpComm* von Subsystem *SC_GUI* durchgeführt werden. Die Eingangswerte sind sinngemäß:

- *acqGroup* – Nummer der Acquisition-Gruppe
- *fileName* – Name des neuen Files
- *varName* – Variablenname
- *enabled* – Es kann Wahr (1) → Start der Datenerfassung oder Falsch (0) → Stopp der Datenerfassung und Speicherung des Files, sein.

Als zweite wird die Bibliothek *time* definiert. Die einzige aber essentielle Funktion, die in dem Skript verwendet wird, ist die Funktion *time.sleep(secs)*. Mit dieser Funktion können Wartezeiten während der Simulation eingestellt werden. Auf dieser Art kann z.B. die Länge eines Fehlerfalles bestimmt werden.

Die Bibliotheken *os* und *shutil* ermöglichen die Operationen mit Files. Da das Programm RT-LAB alle Dateien in den Ordner des Projektes unter die anderen Systemfiles speichert, lohnt es sich, sie in einen anderen Ordner zu verschieben. Die Erstellung des neuen Ordners erfolgt sich in Initialphase mit der Funktion

- **os.mkdir(path)**

Die Verschiebung der Dateien erfolgt sich in Phase *Housekeeping*. Die hier verwendete Funktion ist:

- **shutil.move(oldpath, newpath),**

deren Eingangswerte sind:

- *oldpath* – der Pfad des Files
- *newpath* – der neue Pfad des Files

Beide Pfade müssen den Namen der Dateien mit Dateinamenserweiterung beinhalten.

6 Ergebnisse

Um ein richtiges Verhalten des Modells und der verwendeten Geräte gewährleisten zu können, wurde der zusammengestellte Prüfstand Schritt für Schritt getestet. Dieser Vorgang wird in diesem Kapitel vorgestellt.

6.1 Test des Schutzrelais

In der ersten Phase wurde untersucht, ob das Schutzrelais mit bestimmten Einstellungen den Fehlerabstand richtig misst. Es ist aus der Sicht der Zonengrenzen besonders wichtig. Als erster Schritt wurde das Modell des Windparks stark vereinfacht. Bis auf den ersten Zweig (AAF32) wurden alle andere entfernt und die Leitungsstrecke hat nicht mehr aus zwei Leitungstypen, sondern nur aus einem bestanden. Die gesamte Kabellänge l ist weiterhin 7,4 km lang.

Die auf die Leitungslänge bezogenen Parameter der Leitung sind:

- $R'_1 = 0,06 \frac{\Omega}{km}$
- $L'_1 = 0,318 \frac{mH}{km}$
- $C'_1 = 0,368 \frac{\mu F}{km}$
- $R'_0 = 0,136 \frac{\Omega}{km}$
- $L'_0 = 0,388 \frac{mH}{km}$
- $C'_0 = 0,368 \frac{\mu F}{km}$

Die Leitungsreaktanz des Null- und Mitsystems kann durch die folgende Formel bestimmt werden: [1]

$$X'_{L1} = 2\pi\omega L'_1 = 0,1 \frac{\Omega}{km} \quad X'_{L0} = 2\pi\omega L'_0 = 0,122 \frac{\Omega}{km} \quad (6.1)$$

Die Leitungsimpedanz ergibt sich durch:

$$\underline{Z}_1 = l \cdot (R'_1 + jX'_{L1}) = 0,444 + j0,739 \Omega = 0,862 \cdot e^{j59^\circ} \quad (6.2)$$

Der Sicherheitsabstand von Zone 1 wurde auf 20 % eingestellt. Die Impedanz ist bei dieser Länge:

$$\underline{Z}_{1,80\%} = 0,355 + j0,591 \Omega = 0,69 \cdot e^{j59^\circ} \quad (6.3)$$

Wenn man beim Einstellen des Schutzrelais Sekundärgrößen benutzt, müssen die Übersetzungsverhältnisse des Wandlers richtig definiert werden, sonst entstehen Ungenauigkeiten in der Abstandbestimmung. In diesem Fall wurden die folgenden Übersetzungsverhältnisse verwendet:

- $\ddot{U}_U = 340 \text{ V}$
- $\ddot{U}_I = 1000 \text{ A}$

Nach diesen zwei Werten kann das Übersetzungsverhältnis der Impedanz bestimmt werden:

$$\ddot{U}_Z = \frac{\ddot{U}_U}{\ddot{U}_I} = \frac{340}{1000} = 0,34 \quad (6.4)$$

Dem Schutzrelais können die Sekundärgrößen der Leitung nach der Formel (6.5) angegeben werden:

$$Z_{sek} = \frac{Z_{prim}}{\ddot{U}_Z} \quad (6.5)$$

Die umgerechnete Sekundärgrößen sind:

- $R'_{1,sek} = 0,176 \frac{\Omega}{km}$
- $X'_{L1,sek} = 0,294 \frac{mH}{km}$
- $C'_{1,sek} = 1,082 \frac{\mu F}{km}$
- $\underline{Z}_{1,sek} = 1,306 + j2,1735 \Omega$
- $R'_{0,sek} = 0,4 \frac{\Omega}{km}$
- $X'_{L0,sek} = 0,359 \frac{mH}{km}$
- $C'_{0,sek} = 1,082 \frac{\mu F}{km}$
- $\underline{Z}_{1,80\%,sek} = 1,045 + j1,739 \Omega$

Obwohl er nur bei einpoligem Fehler relevant ist, muss der Erdfehlerfaktor k_0 bestimmt werden. Nach dem Handbuch des Schutzrelais [20] kann dieser Faktor durch zwei Komponenten berechnet werden:

$$K_r = \frac{1}{3} \left(\frac{R_0}{R_1} - 1 \right) = 0,42 \quad K_x = \frac{1}{3} \left(\frac{X_0}{X_1} - 1 \right) = 0,07 \quad (6.6)$$

Wegen der Division können diese Faktoren sowohl durch primäre als auch durch sekundäre Werten berechnet werden. Die Lichtbogenreserve wird auf 3 Ω eingestellt, damit der Kurzschluss mit einem Lichtbogenwiderstand 1 Ω ($R_{on} = 0,5 \Omega$) noch in Zone 1, aber mit 2 Ω ($R_{on} = 1 \Omega$) schon in Zone 2 erkannt wird. Nach diesen Rechnungen können die wichtigsten Parameter des Schutzrelais eingestellt werden.

Nach den entsprechenden Einstellungen werden Fehlersimulationen mit veränderlicher Position des Fehlers und veränderlichem Winkel durchgeführt. Um die Einwirkung des Lichtbogenwiderstandes auf der Leitungsgerade vermeiden zu können, wurde sein Wert in allen Fällen auf ~ 0 Ohm eingestellt. Außerdem wurde nur zweipoliger Kurzschluss ohne Erdberührung simuliert, da so die Erdungsart des Sternpunktes den Kurzschlussstrom nicht beeinflusst, nicht wie bei einpoliger Erdberührung.

Während der ersten Teste wurde die Entfernung des Fehlers in zehnpromzentigen Schritten untersucht. In erwarteter Weise hat das Schutzrelais bei 80 % der Leitungslänge ohne Verzögerung (in ungefähr 30 ms), bei 90 % mit Verzögerung (ca. 300 ms) abgeschaltet. Um die Zonengrenze genauer bestimmen zu können, wurden die Schritte, um die die Fehlerabstand immer erhöht wird, von 10 % auf 0,5 % minimiert. Der Abstand wurde so zwischen 80 % und 85 % geändert und die folgenden Ergebnisse wurden erhalten:

Das Schutzrelais schaltet in jedem Fall unverzögert ab, wenn der Kurzschluss sich bei 80 % der Leitung befindet. Im Gegensatz dazu erfolgte eine verzögerte Abschaltung bei 81 % in jedem Fall unabhängig vom Phasenwinkel des Fehlers. In diesem Punkt kann behauptet werden, dass das Schutzrelais ideal reagiert, da solange die 80 % Entfernung noch zu der Zone 1 gehört, liegt ein Fehler bei 81 % in Zone 2.

Allerdings ist eine interessante Erscheinung bei der 80,5 % Distanz aufgetreten. In diesem Fall konnte das Schutzrelais nicht immer entscheiden, ob der Fehler in Zone 1 oder in Zone 2 liegt. Deswegen sind unterschiedliche Ergebnisse vorgekommen. Die Entscheidung des Relais wird stark durch den Phasenwinkel beeinflusst. Um diesen Zusammenhang bestimmen zu können, wurden zahlreiche Prüfungen in dieser Position mit unterschiedlichen Winkeln durchgeführt.

Das Ziel in diesem Fall war nicht die Erhaltung der Strom- und Spannungsverläufe, sondern nur die Bestimmung für den Fehlerabstand 80,5%, wie oft das Schutzrelais in Zone 1 und Zone 2 den Kurzschluss erkennt. Zu diesem Zweck musste das Python-Skript leicht modifiziert werden. Die Funktion der Datenspeicherung wurde aus dem Code entfernt. Da in diesem Fall sich nur der Phasenwinkel ändert, wurde der Code mit zwei Loop ebenfalls vereinfacht. Jedoch wurde eine Zählerfunktion in das Programm eingebaut. Eine FOR-Loop wiederholt den Fehlerfall 50-mal bei einem bestimmten Phasenwinkel. Nach der Aktivierung des Fault-Blockes wartet das Programm 100 ms, dann überprüft es, ob das Schutzrelais schon reagiert oder nicht. Da die Staffelzeit des Distanzschutzes für Zone 1 ungefähr 30 ms und für Zone 2 300 ms ist, kann so bestimmt werden, in welcher Zone das Relais den Kurzschluss detektiert. Dazu wurde die Funktion *GetSignalsByName(signalName)* von Bibliothek *RtlabApi* verwendet. Dieser Befehl gibt den aktuellen Betrag eines Signals (in diesem Fall boolean 0/1) zurück. Das überprüfte Signal ist in diesem Fall das Aktivierungssignal des Leistungsschalters im Modell. Diese Version von Skript befindet sich als Anhang H.: am Ende des Dokumentes. Die Messergebnisse werden im nächsten Diagramm dargestellt.

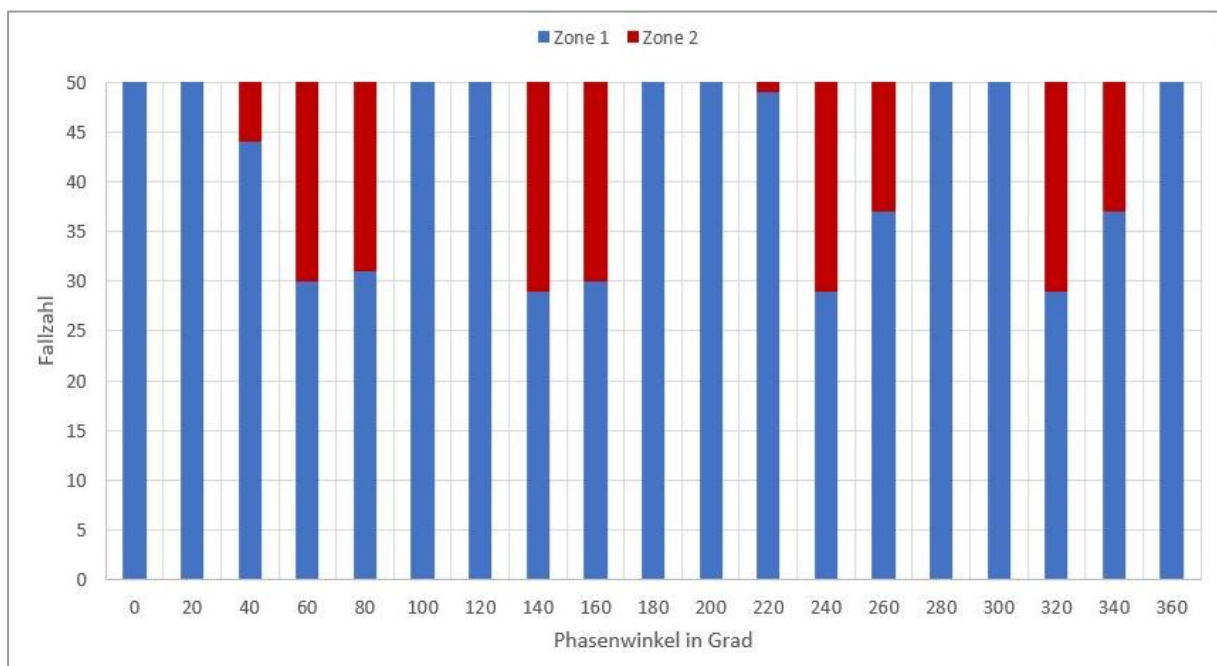


Abbildung 6.1: Messergebnisse

In der Abbildung 6.1 ist es sichtbar, dass der Kurzschlusspunkt meistens in Zone 1 aber bei bestimmten Phasenwinkeln mehrmals in Zone 2 detektiert wurde. Ebenfalls kann man erkennen, dass dieses Phänomen zwischen 0° und 360° eine Periodizität hat. Verglichen mit Abbildung 6.2 kann ein Zusammenhang zwischen der verketteten Spannung U_{L12} und der Zugehörigkeit des Fehlers beobachtet werden. Das Schutzrelais hat den Fehler bei Maximum und bei Minimum der verketteten Spannung mehrmals in Zone 2 detektiert, in anderen Fällen fast immer in Zone 1.

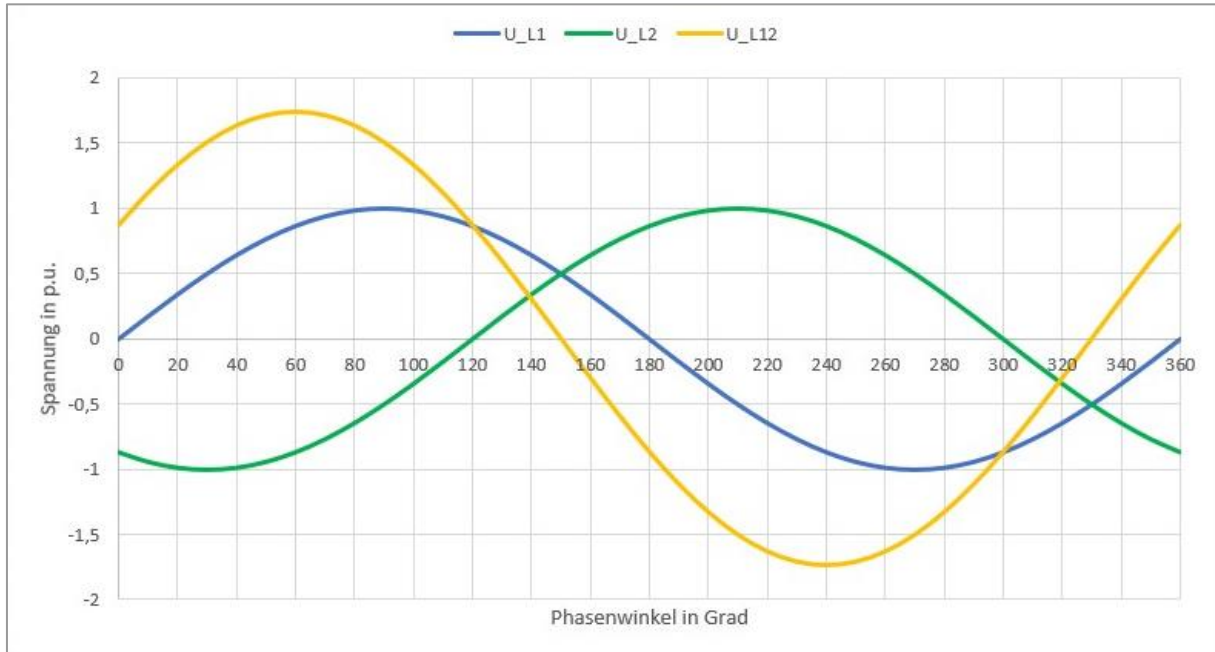


Abbildung 6.2: Phasenverlauf der Spannungen

Der Grund dieses Phänomens ist wahrscheinlich die Programmierung des Schutzrelais, jedoch gilt seine detaillierte Betrachtung nicht als Gegenstand dieser Diplomarbeit.

6.2 Testen mit Omicron CMC 356

Um die Authentie der Messergebnisse mit OP5600 überprüfen und einen möglichen Messfehler wegen der ungenauen Relaisinstellungen ausschließen zu können, wurde das System auch mit dem Omicron CMC 356 getestet. Um diese Tests durchführen zu können, wurden als erster Schritt die untersuchenden Fehlerfälle mit den entsprechenden Parametern im Simulink simuliert und die Messergebnisse des Blocks Three-Phase Measurement AAF32 wurden durch einen Scope-Block gespeichert. Nach Einordnung der Daten in bestimmten Strukturen kann ein Comtrade-File durch ein Python-Skript, das an der TU Wien entwickelt wurde, aus diesen Messdaten hergestellt werden. Dieses Comtrade-File kann mit dem Programm Test-Universe eingelesen werden (siehe Abbildung 6.3), und eine Simulation durchgeführt werden.

Zur Übertragung der Ausgangssignale vom Prüfgerät zum Schutzrelais wurde die Generator-Kombinationsbuchse (Generator Combination Socket) mit dem dazugehörigen Kabel verwendet. Zur Detektion der Reaktion des Schutzrelais wurde das Panel Binary / Analog Input genutzt. Von der Seite des Relais wurden gleiche Anschlüsse und Kanäle verwendet wie bei der Simulation mit OP5600.

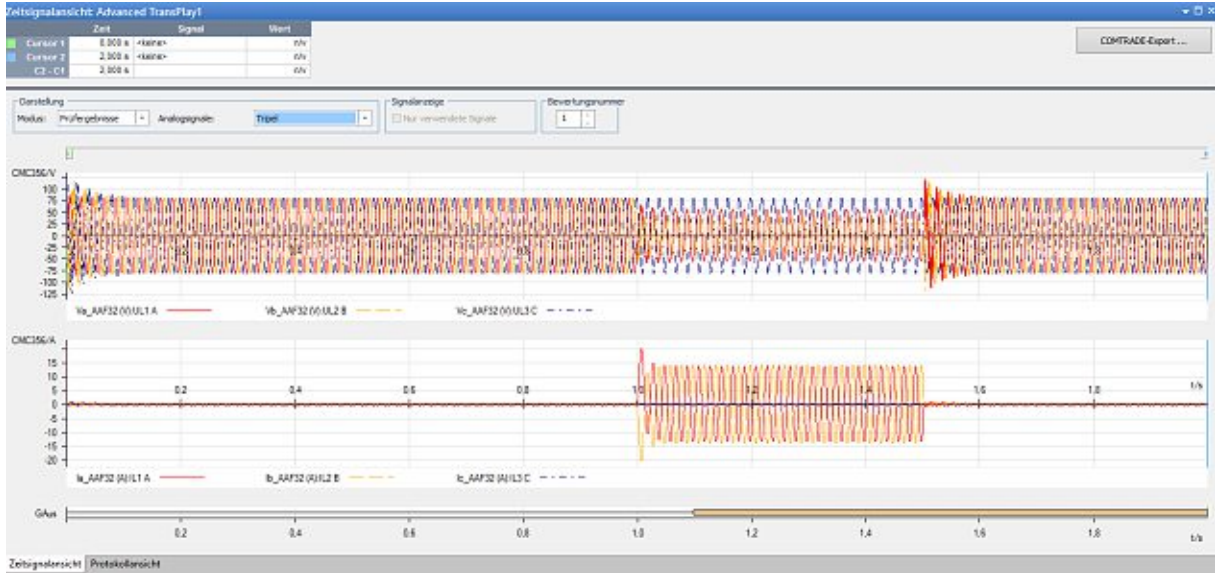


Abbildung 6.3: Simulationsansicht von Test-Univers

6.2.1 Zweipoliger Kurzschluss

Die Prüfungen haben sich auf die Distanzbestimmung im Bereich der Zonengrenze fokussiert. Um den Einfluss der Erdungsart und des Erdfehlerfaktors vermeiden zu können, wird weiterhin ein zweipoliger Kurzschluss ohne Erdberührung und ohne Fehlerwiderstand ($R_{on} = 0,001 \Omega$) simuliert. Die Messpunkte und -Ergebnisse werden in der Tabelle 6.1 zusammengefasst. Für jeden Messpunkt wurden fünf Tests durchgeführt, deren Durchschnitt dargestellt wird.

Tabelle 6.1: Messergebnisse

Distanz		OP5600			CMC 356		
[%]	[km]	Zone	Zeit [ms]	gemessener Abstand [km]	Zone	Zeit [ms]	gemessener Abstand [km]
0	0	Z1	1	-	Z1	1	0
78	5,772	Z1	24	5,7	Z1	24	5,8
78,5	5,809	Z1	24	5,7	Z1	24	5,8
79	5,846	Z1	24	5,75	Z1	24	5,8
79,5	5,883	Z1	24	5,8	Z1	29	5,9
80	5,92	Z1	24	5,8	Z1	28	5,9
80,5	5,957	Z1	24	5,9	Z2	298	5,9
81	5,994	Z2	299	5,9	Z2	298	6,0
81,5	6,031	Z2	299	6,0	Z2	298	6,0
82	6,068	Z2	299	6,0	Z2	298	6,1
100	7,4	Z2	299	7,4	Z2	297	7,4

Die Grenze der Zone 1 liegt weiterhin bei 80 % der Leitung. Man kann nach der Tabelle 6.1 bemerken, dass die Distanzbestimmung im Fall der Messung mit CMC 356 um 0,5 % genauer ist als bei der Simulation mit OP5600. Ein möglicher Grund dafür ist, dass bei OP5600 ein Verstärker (Doble) benötigt wird, da der Real-Time-Simulator nur über analoge Kanäle mit maximalen ± 16 V Spannungsniveau verfügt. Wegen des Verstärkers befindet sich ein weiteres Übersetzungsverhältnis im System (siehe Abschnitt 4.3), dessen Ungenauigkeit diese leichte Abweichung zwischen den beiden Ergebnissen verursachen kann. Dieses Problem kann aber durch die Justierung der Gain-Werte im Modell gelöst werden.

Zu dem besseren Vergleich können die durch das Schutzrelais gespeicherten Messdaten genutzt werden. Die Störverläufe können aus dem Gerät heruntergeladen und als Comtrade-File geöffnet werden. Auf diese Weise kann man bemerken, welche Spannungen und Ströme das Schutzrelais während des Fehlerfalles gemessen hat. Zur Darstellung dieser Files wird das Programm TransViewer verwendet (siehe Abbildung 6.4 und Abbildung 6.5). In beiden Fällen wird ein zweipoliger Kurzschluss mit einem Fehlerabstand von 70% simuliert. In beiden Abbildungen werden die Messdaten als Sekundärwerte dargestellt.

Der wesentliche Unterschied zwischen den beiden Störfällen ist die Dauer des Kurzschlusses. Bei der Prüfung mit OP5600 wird der Kurzschluss mit der Auslösung des Schutzrelais abgeschaltet werden. Da vorsimulierte Messdaten im Test mit CMC 356 verwendet werden, besteht diese Möglichkeit in diesem Fall nicht. Es ist also ein wesentlicher Vorteil des Real-Time-Simulators OP5600, dass diese Methode eine Simulation ermöglicht, die näher an der Realität steht als im Fall von CMC 356. Das Schutzrelais hat aber bei beiden Methoden gleich reagiert, wie es in Tabelle 6.1 ebenfalls sichtbar ist, obwohl das Schutzrelais bei Prüfung mit CMC 356 die Distanz nicht immer berechnen konnte. Es reagiert entsprechend dem Fehlerort aber in der Zusammenfassung der Störfallmeldung hat sich oft eine Fehlermeldung statt des berechneten Abstandes befunden. Dieses Phänomen wurde während des Projektes aus mehreren Aspekten untersucht aber keine Antwort wurde darauf gefunden.

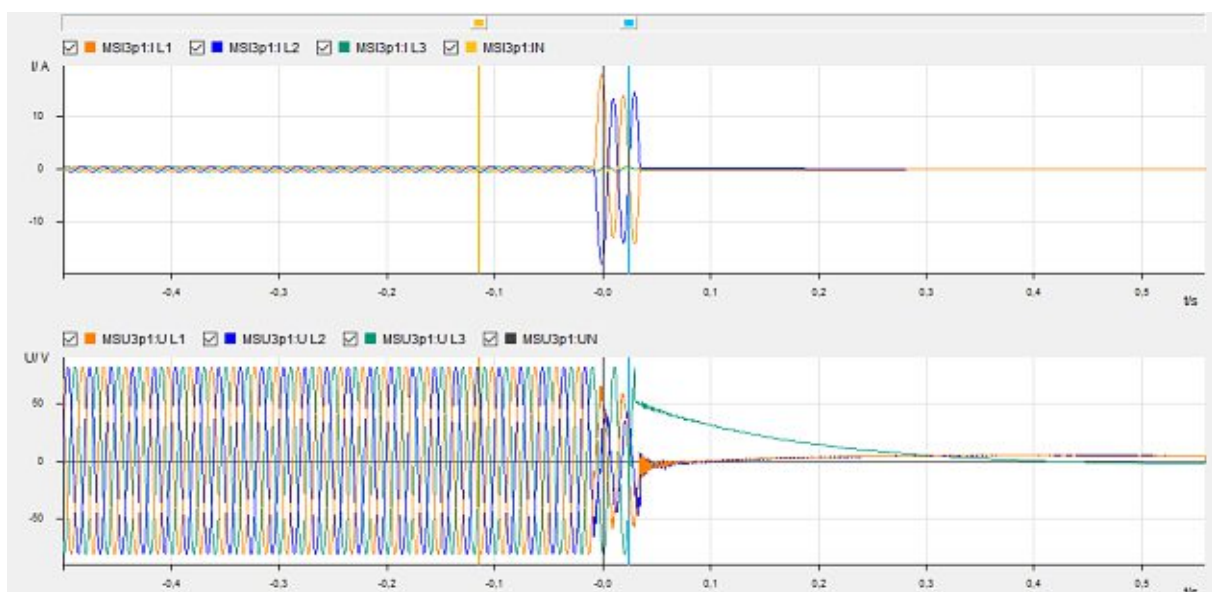


Abbildung 6.4: Kurzschlussverlauf des zweipoligen Kurzschlusses (OP5600-Simulation)

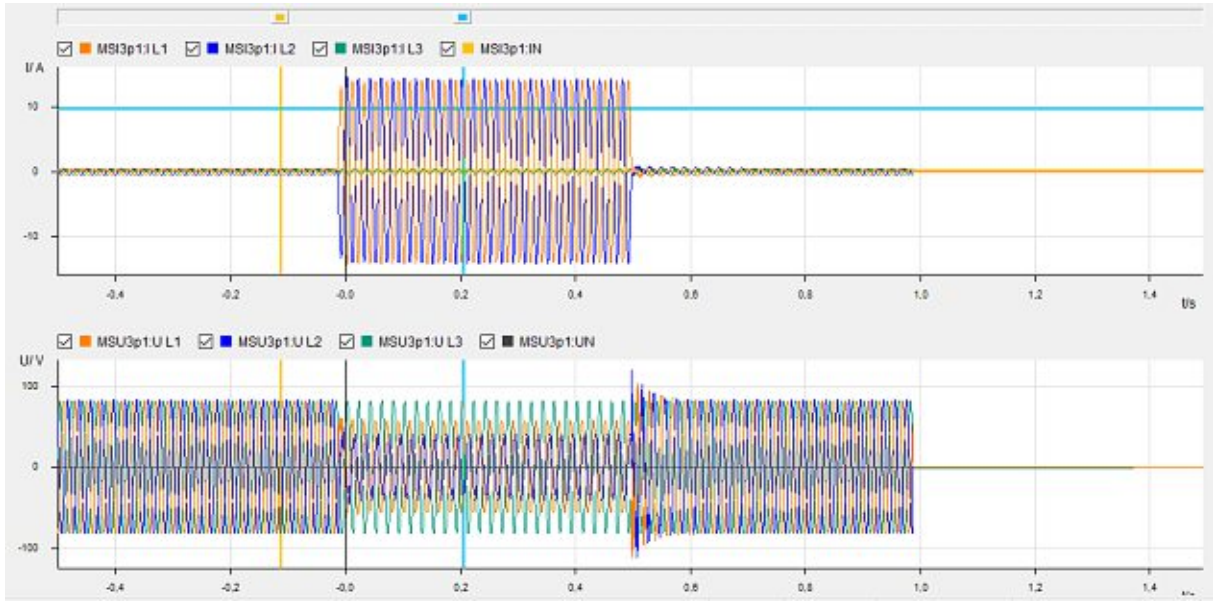


Abbildung 6.5: Kurzschlussverlauf des zweipoligen Kurzschlusses (CMC 356-Test)

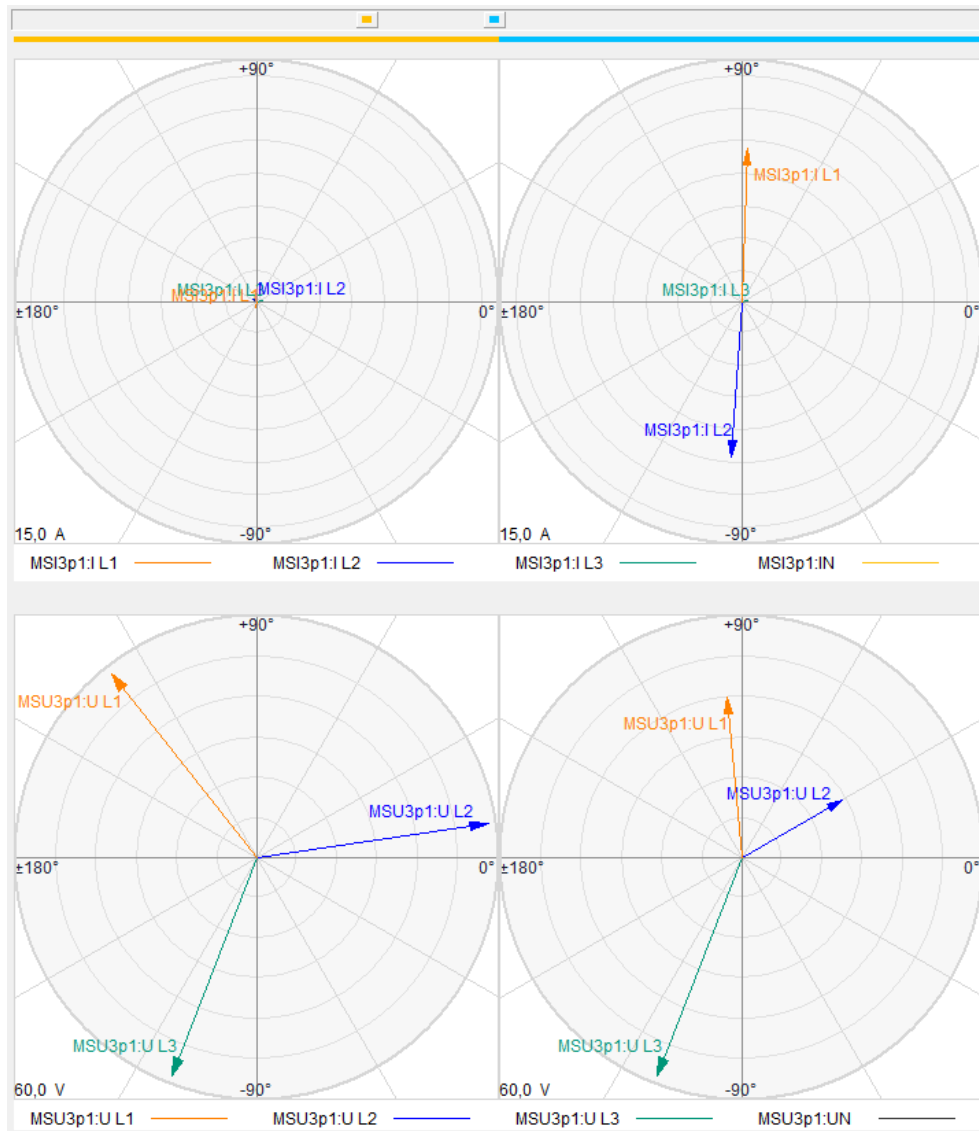


Abbildung 6.6: Phasor-Diagramm des zweipoligen Kurzschlusses (OP5600-Simulation)

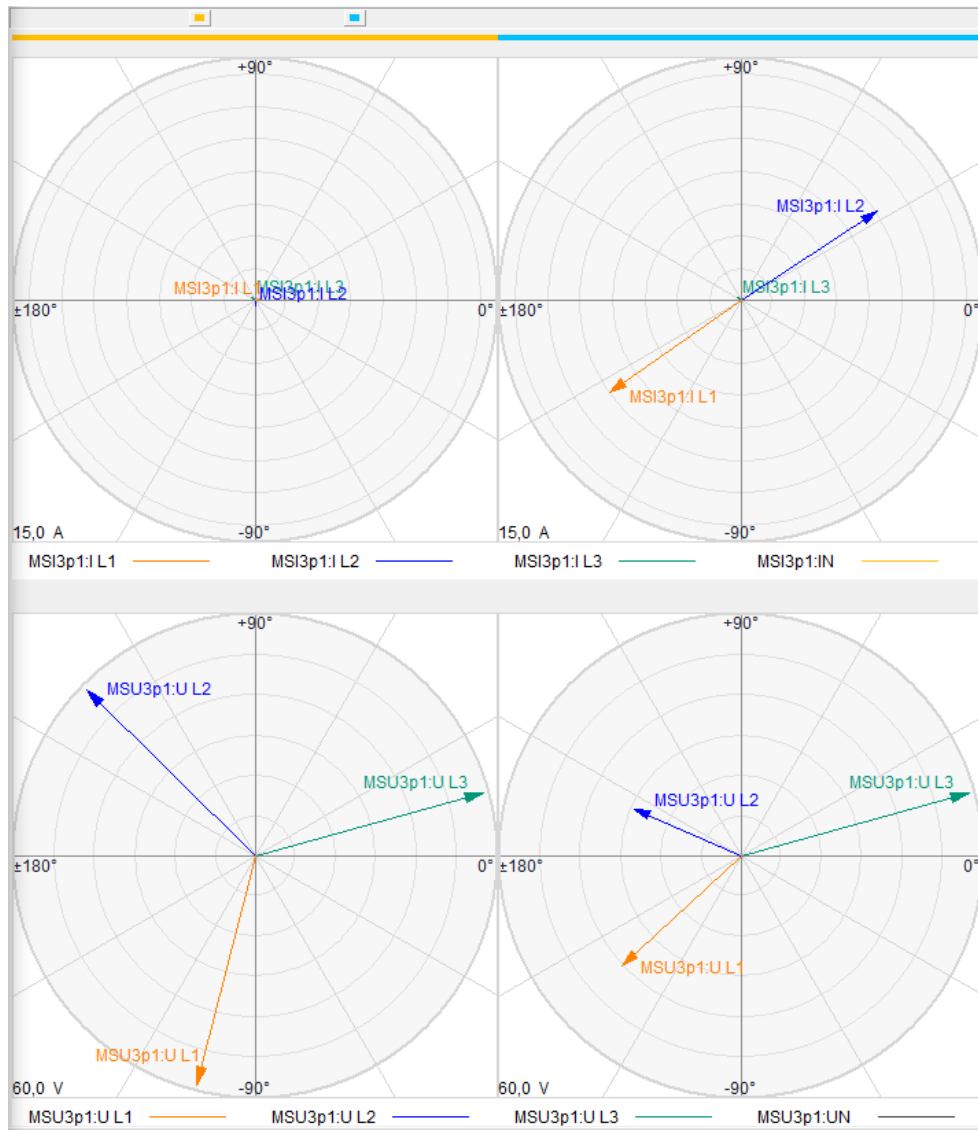


Abbildung 6.7: Phasor-Diagramm des zweipoligen Kurzschlusses (CMC 356-Test)

Das Programm TransView ermöglicht eine vektorielle Darstellung der Spannungs- und Stromwerte für bestimmte Zeitpunkte. In den Abbildung 6.6 – Abbildung 6.7 werden die obigen zwei Störfälle dargestellt. An der linken Seite der Abbildungen werden die Spannungs- (unten) und Stromvektoren (oben) im normalen Zustand vor dem Kurzschluss (gelbe Markierung) dargestellt, während sie auf der rechten Seite während des Fehlerfalles (blaue Markierung) gezeichnet werden.

Aus diesen Abbildungen ist es ebenfalls sichtbar, dass die Spannungs- und Stromvektoren sich in beiden Fällen in gleicher Weise ändern. Also diese Tests bestätigen die Authentie der Simulationen des Real-Time-Simulators OP5600.

6.2.2 Einpoliger Fehler

Wie es schon früher erwähnt wurde, ist die Einstellung eines Distanzschutzrelais für einen einpoligen Erdschluss wegen der Berücksichtigung der Erdungsart und der Parameter von Nullsystem schwieriger als für einen zweipoligen Kurzschluss. Während dieses Projektes waren die Eigenschaften des modellierten Netzes weitere erschwerende Faktoren.

Im Modell sorgt ein Erdungstransformator (BGT31) für die Sternpunktbehandlung in der Mittelspannungsseite. Er wird durch einen Widerstand (RA001) mit dem Erdpotential verbunden, der 25Ω beträgt. In diesem Fall ist das System also hochohmig geerdet. Damit dieser große Widerstand das System nicht beeinflusst, wurde seine Betrag auf $0,001 \Omega$ umgestellt. Aber trotz der Umstellung des Widerstandes hat sich das System weiterhin fast wie ein isoliertes Netz verhalten.

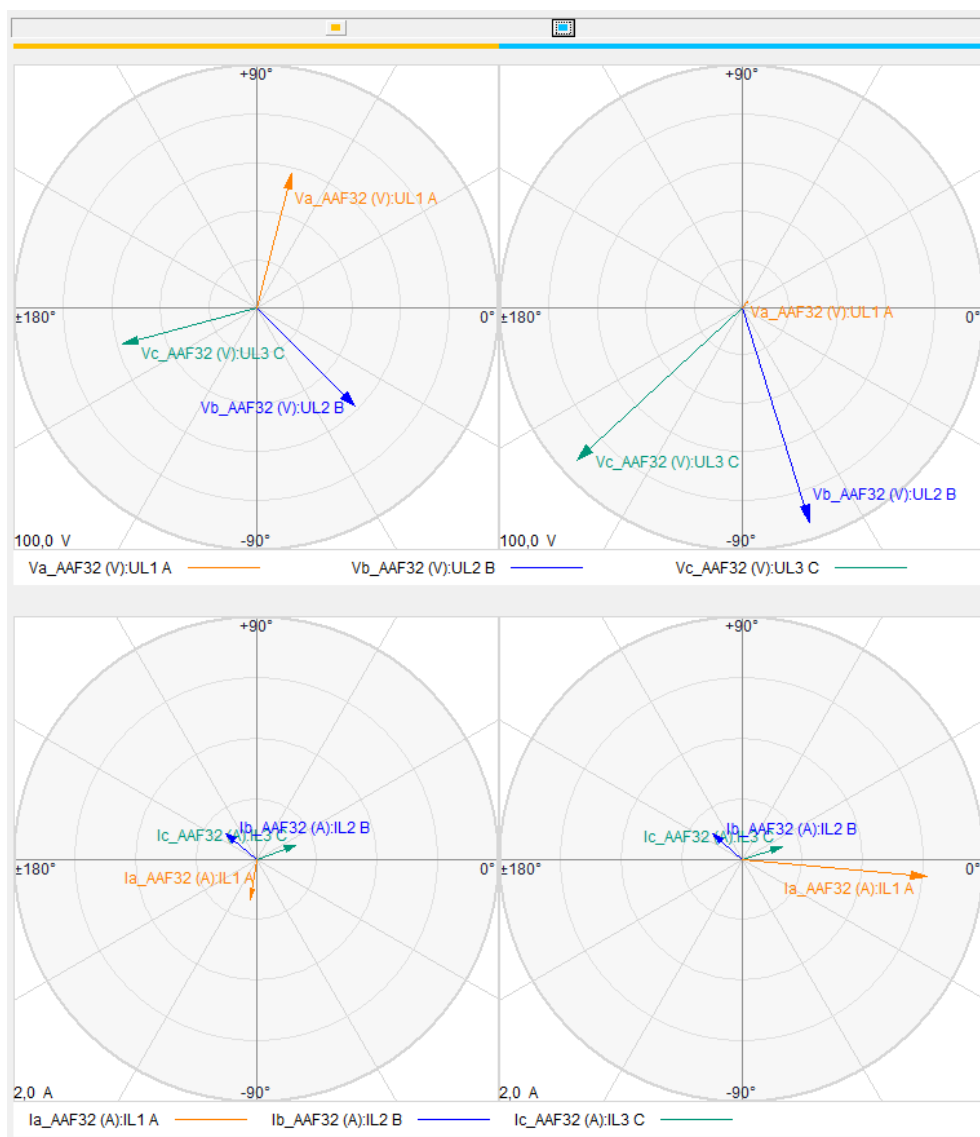


Abbildung 6.8: Phasor-Diagramm des einpoligen Kurzschlusses im Fall mit Erdungstransformator

In der Abbildung 6.8 kann dieses Verhalten beobachtet werden. Der Kurzschlussstrom ist deutlich kleiner als bei zweipoligem Kurzschluss und die Spannungsverlagerung ist ebenfalls gut sichtbar. Der Grund für dieses Phänomen ist die Einstellungsparameter des Erdungstransformators, nämlich die hohe Nullimpedanz $\underline{Z}_{0,ET} = 11,56 + j30,056 \Omega$. Diese hohe Impedanz hat eine große Abweichung bei der Bestimmung der Fehlerdistanz verursacht.

Um den Einfluss der Nullimpedanz des Erdungstransformators auf das System vermeiden zu können, wurde die Erdung durch den Drehstromtransformator statt des Erdungstransformators gewährleistet. Dazu wurde die Schaltgruppe aus Ynd auf Dyn umgestellt und der Erdungstransformator wurde aus dem System entfernt.

Obwohl das Schutzrelais nach dieser Umstellung den Abstand etwas genauer bestimmen konnte, bestand weiterhin ein Unterschied von ca. 10 % zwischen der bestimmten und der eingestellten Distanz. Ein möglicher Grund dafür kann die zweiseitige Einspeisung sein, aber da die präzise Einstellung des Schutzrelais kein Gegenstand dieser Diplomarbeit ist, wurde dieses Problem nicht weiter untersucht. Jedoch, da die Distanzbestimmung des Schutzrelais für einpoligen Fehler mit den beiden Prüfgeräten (OP5600, CMC 356) die gleiche Abweichung produziert, kann die Authentie beider Messmethoden festgelegt werden. Die Störverläufe eines einpoligen Erdkurzschlusses mit Fehlerdistanz 70 % werden in Abbildung 6.8 – Abbildung 6.12 dargestellt.

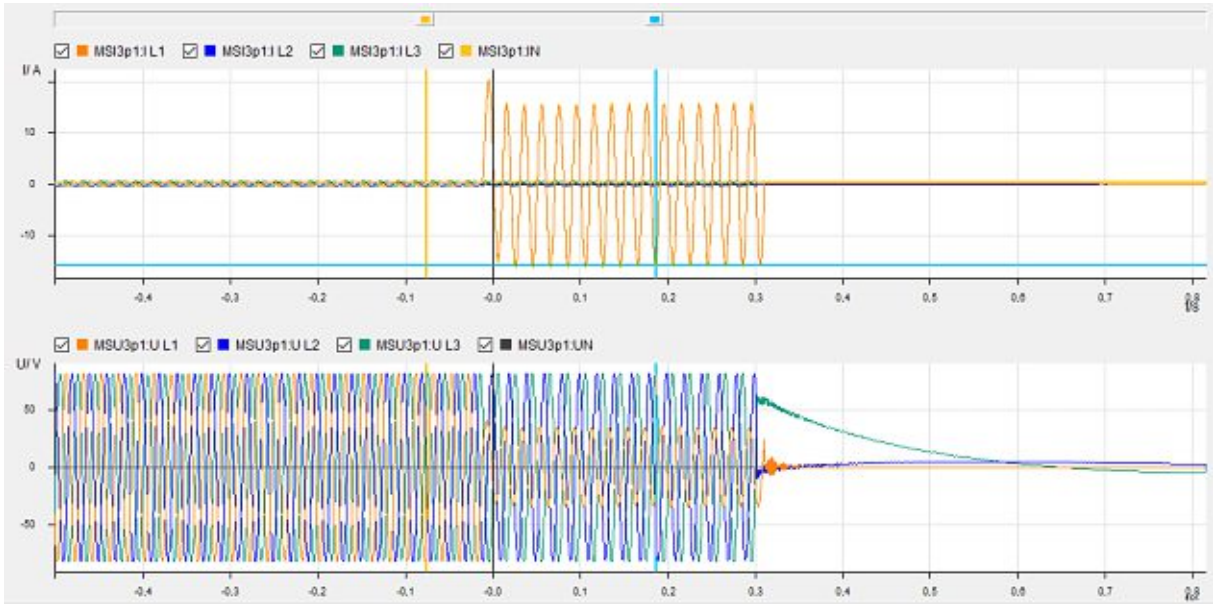


Abbildung 6.9: Kurzschlussverlauf des einpoligen Kurzschlusses (OP5600-Simulation)

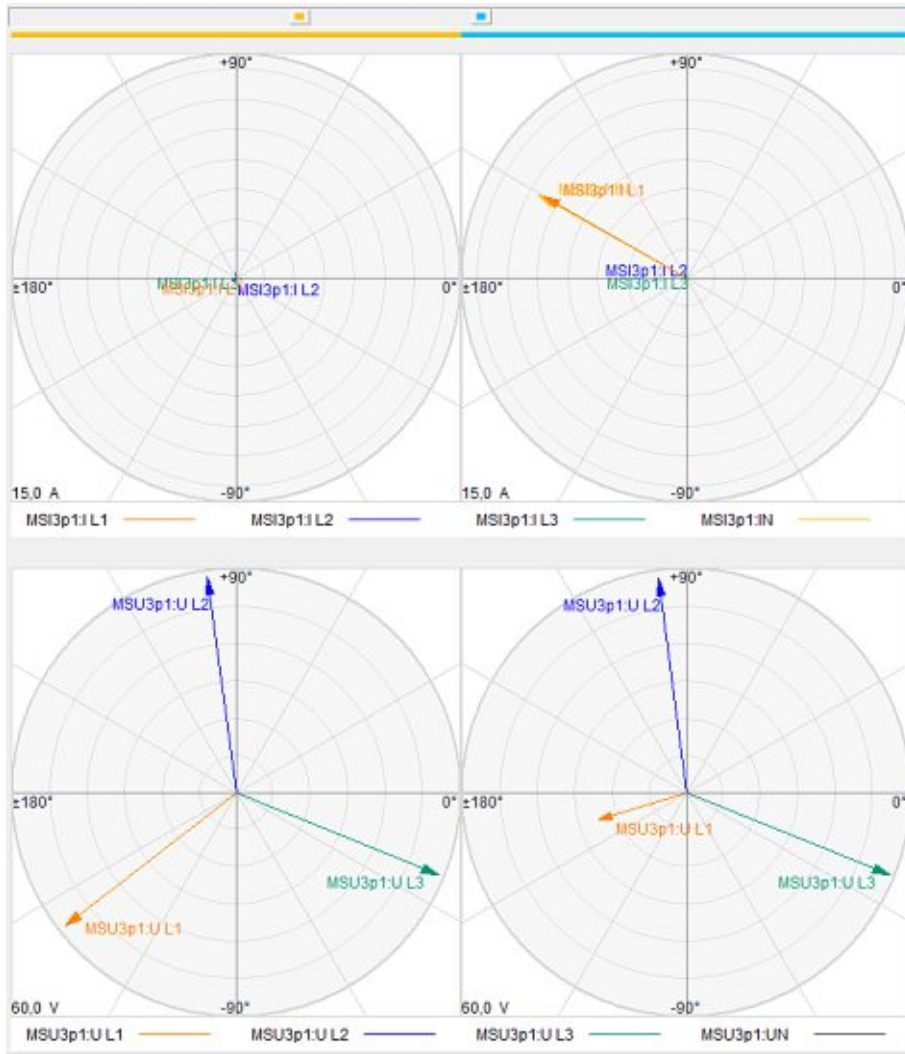


Abbildung 6.10: Phasor-Diagramm des einpoligen Kurzschlusses (OP5600-Simulation)

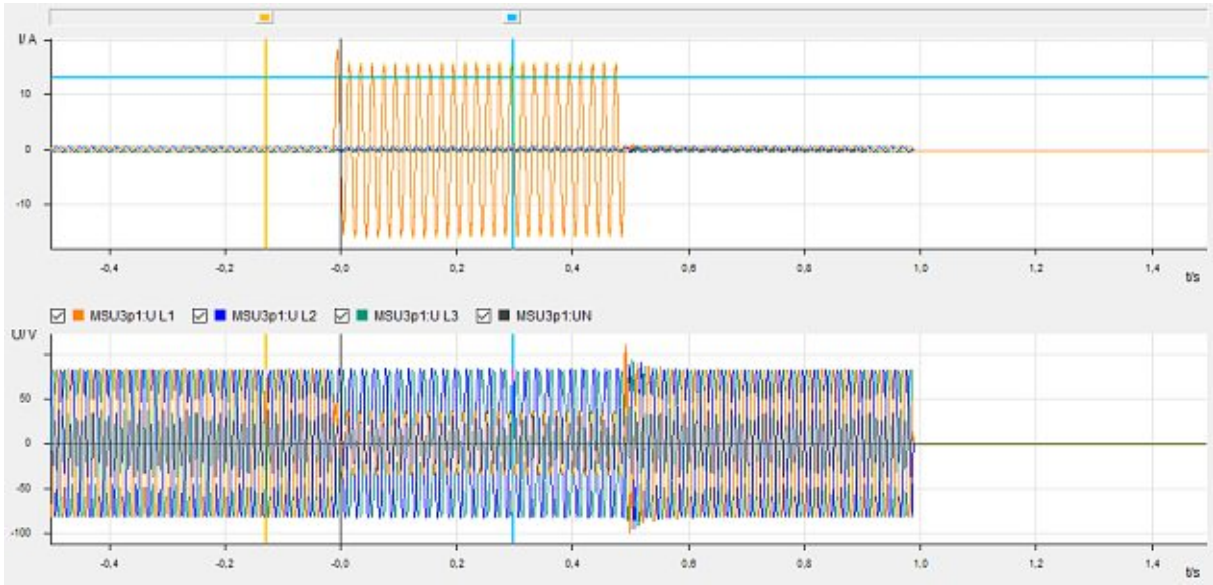


Abbildung 6.11: Kurzschlussverlauf des einpoligen Kurzschlusses (CMC 356-Test)

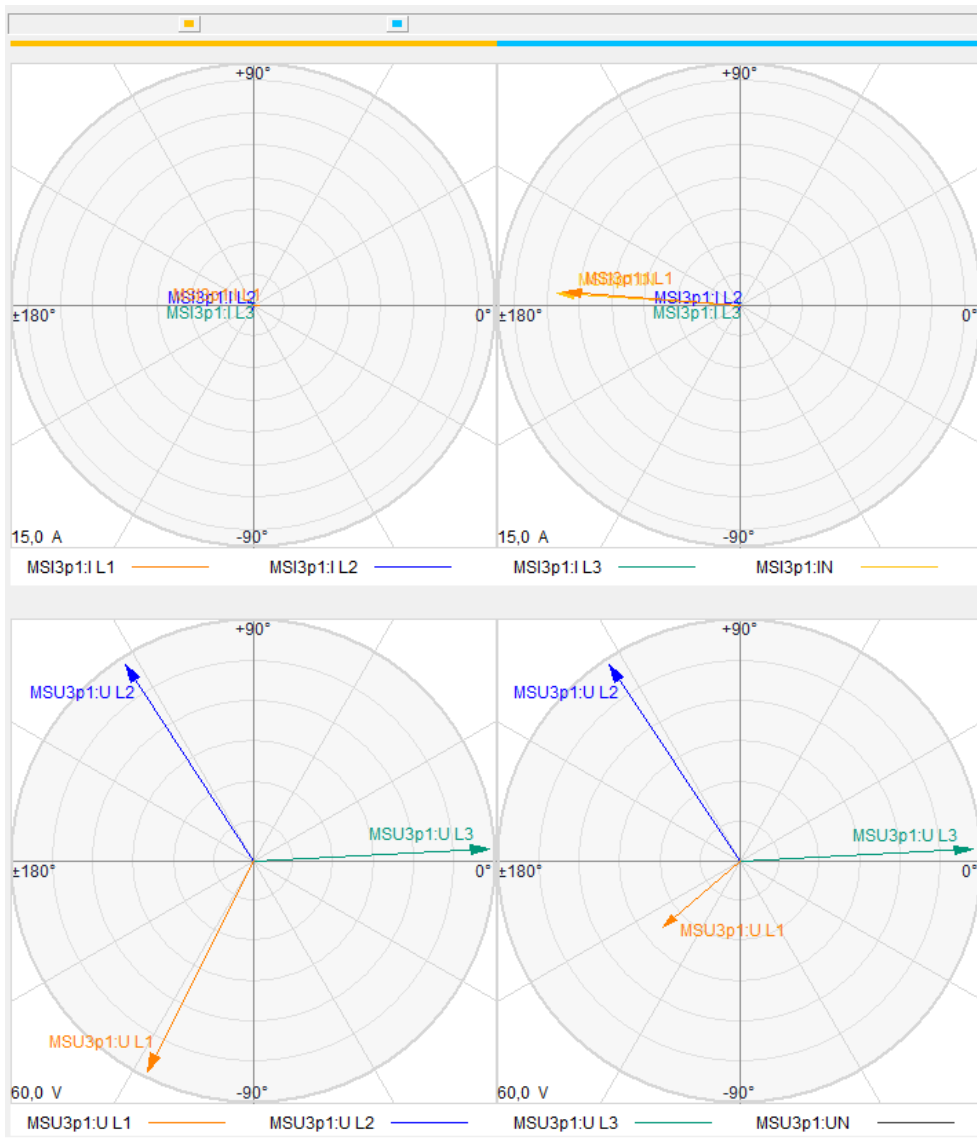


Abbildung 6.12: Phasor-Diagramm des einpoligen Kurzschlusses (CMC 356-Test)

7 Ausblick

Dieses Kapitel beschäftigt sich mit den Grenzen der untersuchten Real-Time-Technologie fokussierend auf die Leistung und die Rechnerkapazität des Simulators bzw. mit den Weiterentwicklungsmöglichkeiten des Messsystems.

In der Real-Time-Technologie, wenn der Simulator die nötigen Berechnungen in einer Periode nur über eine längere Zeitdauer durchführen kann als der feste Zeitschritt, bezeichnet man als Überlauf (overrun). Es bedeutet, dass die Simulation für diese Zeitperiode ihren real-time Charakter verliert. Dieses Phänomen und seine mögliche Vermeidung werden in diesem Abschnitt bezogen auf den Simulator von OPAL-RT enger betrachtet.

Bei einem Simulink-Modell, das aus SimPowerSystems (SPS) Blöcken besteht, hängt seine Komplexität von der Anzahl der Schalter im System ab. Wenn ein Modell n Schalter beinhaltet, müssen 2^n unterschiedliche Netztopologien berücksichtigt werden, die die Lösung von 2^n Systemgleichungen bedeuten. Der Solver von SPS verwendet die Zustandsraumdarstellung (State-space representation). Nach dieser Methode wird das System durch zwei Gleichungen dargestellt. Die erste Gleichung bestimmt den Zustand, die zweite legt den Ausgang des Systems fest. [21]

$$x'(t) = A(t)x(t) + B(t)u(t) \quad (7.1)$$

$$y(t) = C(t)x(t) + D(t)u(t) \quad (7.2)$$

wo

- x – Zustandsvektors
- u – Eingangsvektor
- y – Ausgangsvektor
- A – Systemmatrix
- B – Eingangsmatrix
- C – Ausgangsmatrix
- D – Durchgangsmatrix

In der Abbildung 7.1 wird der Monitoring-Panel von RT-LAB während der Simulation des vereinfachten Windparks (nur ein Zweig, siehe Abschnitt 6.1) dargestellt. Trotz der niedrigen Ausnutzung des Prozessorkernes (37,83 %) sind während der Simulation mehrere Überläufe passiert. Man konnte während des Simulationsvorganges beobachten, dass die Überläufe (overruns) im Allgemeinen aufgetreten sind, wenn das System nach einem Fehlerfall in den Normalzustand zurückgeschaltet wird. Das heißt, in diesem Moment werden die Leistungsschalter in allen drei Phasen gleichzeitig zurückgeschaltet und die Stromquellen werden wiederaktiviert.

Model: Breaker_Test Ts= 1.0E-4[s] T=400,5706[s] Number of overruns=254

Probes	Info				Data	
	Usage [%]	Min	Max	Mean	1	2
Breaker_Test Ts= 1.0E-4[s]	37,83%					
SM_MASTER T_9997474E-5[s]	37,83%	dt= 37,43 [us]	dt= 38,37 [us]	dt= 37,83 [us]	dt= 38,00 [us]	dt= 37,94 [us]
New data acquisition	0,12%	dt= 0,09 [us]	dt= 0,13 [us]	dt= 0,12 [us]	dt= 0,11 [us]	dt= 0,13 [us]
Major computation time	14,68%	dt= 14,42 [us]	dt= 15,13 [us]	dt= 14,68 [us]	dt= 14,68 [us]	dt= 14,47 [us]
Minor computation time	21,12%	dt= 21,04 [us]	dt= 21,22 [us]	dt= 21,12 [us]	dt= 21,10 [us]	dt= 21,18 [us]
Execution cycle	37,83%	dt= 37,43 [us]	dt= 38,37 [us]	dt= 37,83 [us]	dt= 38,00 [us]	dt= 37,94 [us]
Total step size	100,0%	dt= 99,77 [us]	dt= 100,28 [us]	dt= 100,02 [us]	dt= 100,03 [us]	dt= 100,11 [us]
Total idle	60,47%	dt= 0,11 [us]	dt= 61,48 [us]	dt= 60,47 [us]	dt= 0,11 [us]	dt= 61,19 [us]
Op7160ex1_wait_din_2_opctrl	0,1%	dt= 0,10 [us]	dt= 0,11 [us]	dt= 0,10 [us]	dt= 0,10 [us]	dt= 0,10 [us]

Abbildung 7.1: Monitoring-Panel

Während der Simulation des ganzen Windparks (Ausnutzung: ca. 85%) treten diese Überlaufen deutlich regelmäßiger auf, nicht nur bei den oben erwähnten Fällen. Es gibt mehrere Methoden, mit den die Belastung des Simulators reduziert werden können. Jedoch, wie es später eindeutig wird, kann keine von diesen Methoden für dieses Modell gut implementiert werden.

7.1 Verwendung mehrerer Prozessorkerne

Eine sehr naheliegende Lösung ist die Ausnutzung mehrerer Kerne der Simulator. Wie in der Tabelle 4.2 sichtbar ist, hat der Simulator OP5600 12 Prozessorkerne (Total Core), woraus während der Simulation nur einer verwendet wird. Um mehrere Kerne verwenden zu können, müssen die Elemente der Subsystem *Computation* in mehrere Subsysteme eingeordnet werden (siehe Abschnitt 4.2.2). Hier ist es ebenfalls essentiell, auf die Benennung der Subsysteme zu beachten.

Die Schwachstelle dieser Methode ist die Kommunikation zwischen den Subsystemen. Nämlich kann die Kommunikation durch die *OpComm*-Blöcke erfolgen und sie können nur mit standardmäßigen Simulink-Interfaces verbunden werden. Es ist in diesem Fall problematisch, da das Modell größtenteils aus SPS-Blöcken und ihren Interfaces besteht. Mit dieser Methode könnte nur ein kleines Teil des Modells separiert werden, der z.B. für die Winkelrechnung oder Einlesen des Digitaleinganges verantwortlich ist. So wäre einer von beiden Kernen weiterhin überlastet. Damit man SPS-Elemente voneinander separieren kann, muss die ARTEMiS-Toolbox von OPAL-RT verwendet werden.

7.2 ARTEMiS

ARTEMiS (Advanced Real-Time Electro-Magnetic Solvers) ist eine Toolbox von RT-LAB, mit dem man die Simulation eines Stromversorgungssystems oder Leistungselektronikmodelles beschleunigen kann. Die Toolbox enthält eine Bibliothek mit gleichen oder leicht modifizierten Blöcken wie SimPowerSystems (z.B.: Three-Phase Fault Block, Asynchronmaschine), aber ihr Kernelement ist der ARTEMiS-Solver. [21]

Dieser Solver verwendet in ähnlicher Weise wie SPS eine Zustandsraumdarstellung, jedoch vorberechnet und diskretisiert er alle State-Space-Matrizen für alle Kombinationen der Switch-Topologien vor der Echtzeit- (oder Offline-) Simulation und speichert sie im Cache-Speicher. Deswegen können die festen Zeitschritte besser eingehalten werden, da die benötigte Zeit der Übertragung der State-Space-Matrizen aus dem Zwischenspeicher deutlich weniger ist als die Wiederberechnungszeit der Matrizen. Jedoch benötigt diese Methode eine große Speicherkapazität und hat deswegen eine Begrenzung bezüglich der maximalen Zahl der Schalter, die 28 in einem Modell beträgt. [21]

Um die Cache-Speicher nicht zu überfüllen, muss das große elektrische System durch Entkopplung (decoupling) in mehrere kleineren, elektrischen Systeme aufgeteilt werden. So erhält man statt eine große State-Space-Matrix mit vielen Elementen kleinere Matrizen mit deutlich weniger Elementen (siehe Abbildung 7.2)

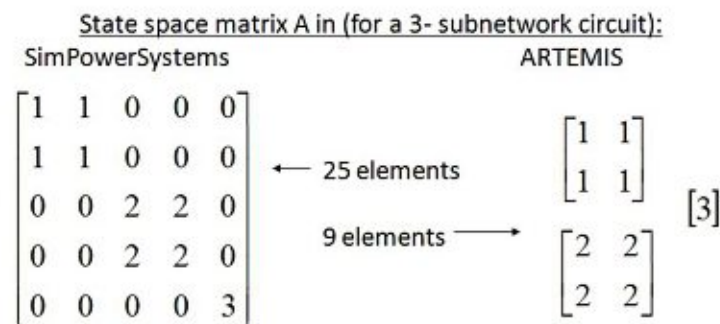


Abbildung 7.2: Matrixreduktion durch ARTEMiS [21]

Zur Entkopplung bietet ARTEMiS zwei Möglichkeiten. Die erste ist das Ersetzen der SPS-Leitungen durch ARTEMiS-Leitungen. Damit diese Technik funktionieren kann, muss zwischen den zwei Systemteilen mindestens eine Zeitschrittverzögerung (one time-step delay) eingeführt werden. Es gibt zwei Hauptarten von ARTEMiS-Linien: [21]

- **Distributed Parameters Lines (DPL)**

Bei einer in elektrotechnischer Hinsicht langen Leitung entsteht eine Phasenverschiebung der Spannung zwischen beiden Enden der Leitung. Diese natürliche Verzögerung, die in einer langen Übertragungsleitung entsteht, kann mit dieser Linie modelliert werden. In diesem Fall ist die Verzögerung größer oder gleich als ein Zeitschritt der Simulation. Die folgende Ungleichung muss sich erfüllen:

$$l_{line}[km] \geq T_s[s] \cdot 300\,000[km/s] \quad (7.3)$$

- **Stub Line**

Wenn sich die obige Ungleichung nicht erfüllt, d.h. eine elektrotechnisch kurze Leitung vorhanden ist, kann die Linie *Stub Line* verwendet werden. Diese Linie führt eine Zeitschrittverzögerung ein.

Beide Linien können innerhalb eines Subsystems genutzt werden, aber sie sind für die Kopplung der Subsysteme geeignet. Wenn das Modell keine Übertragungsleitung beinhaltet, kann das Modell durch das Block *State Space Nodal* (SSN) entkoppelt werden. SSN führt keine Zeitverzögerung in das System, also es ist ideal für die Modellierung von Microgrids. Dieser Interface-Block kann multiple Ports haben, die als Spannungsquelle (V-type port) für induktive Elementen oder als Stromquelle (I-type port) für kapazitive Elementen definiert werden.

Leider kann eine Entkopplung in dem untersuchten Modell nicht oder mit vielen Einschränkungen durchgeführt werden. Da das System keine lange Übertragungsleitung beinhaltet, kann keine *DPL* verwendet werden. Mit *Stub Line* oder *SSN* könnten bestimmte Teile des Modells entkoppelt werden, aber das Hauptproblem liegt bei der Positionsfestlegung des Fehlers. Wie es im Abschnitt 5.3.3 erklärt wurde, kann ein *Three-Phase Fault* Block nur zwischen zwei Leitungssektionen (*Three-Phase PI Section Line*) hinzugefügt werden. Deshalb wurde die 7,4 km lange Hauptleitung in 10 gleich große, in Reihe geschaltete Stücke aufgeteilt. Es macht die Verwendung von *Stub Lines* unmöglich, da es eine zehnfache Zeitverzögerung bedeuten würde. SSN-Blöcke können ebenfalls nicht verwendet werden, da sie mit den Leitungssektionen nicht in Reihe geschaltet werden können. Der Grund dafür ist, dass Simulink die Reihenschaltung einer Stromquelle mit einer Induktivität bzw. die Parallelschaltung der Spannungsquelle mit einer Kapazität nicht ermöglicht. Wegen des Aufbaues der π -Leitungsmodell führt diese Verbindung zur Fehlermeldung.

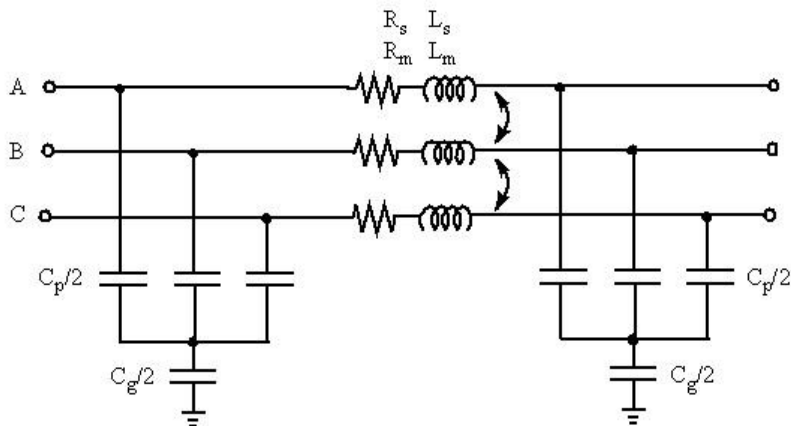


Abbildung 7.3: Ersatzschaltbild von *Three-Phase PI Section Line* [11]

Das Block *Three-Phase Fault* wird durch das Schließen des entsprechenden Leistungsschalters mit dem bestimmten Punkt verbunden. Diese Methode erfordert viele (in diesem Fall 11) Leistungsschalter. Obwohl 28 Schalter in einer Gruppe freigegeben sind, zählt ein dreiphasiger Leistungsschalter für 3 Schalter. Das bedeutet 33 Schalter in dieser Gruppe, deren Entkopplung, wie es schon erwähnt wurde, unmöglich ist. Die einzige Möglichkeit in diesem Fall ist die Reduktion der möglichen Fehlerorte.

Wie es sichtbar ist, kann ARTEMiS für die Optimierung dieses Modells nicht oder nur mit den oben erwähnten Einschränkungen verwendet werden. Jedoch ist nach der Reduktion der Schalter eine problemlose Simulationsdurchführung nicht gewährleistet.

8 Zusammenfassung

Man kann feststellen, dass das während dieser Arbeit aufgebaute und vorgestellte Messsystem, das auf dem Real-Time-Simulator OP5600 basiert, ist für das Testen eines Schutzrelais geeignet. Im Abschnitt 6.2 wurde es vorgestellt, wie die Ergebnisse der Testmethode durch ein anderes Testverfahren beglaubigt wurden.

Der wesentlichste Vorteil dieser Methode ist ihre Geschwindigkeit. Da die Simulationen in Echtzeit durchgeführt werden können, kann eine riesige Menge von Fehlerfällen schnell nacheinander simuliert werden. Im Fall einer Offline-Simulation im Simulink dauern diese Simulationen in einem allgemeinen PC deutlich länger. Obwohl das Prüfgerät CMC 356 ebenfalls die schnelle Wiederholung der Tests ermöglicht, können nur gleiche Störverläufe wiederholt werden. Mit dem entsprechenden Simulink-Modellumbau wurde die Änderungsmöglichkeit der Parameter des Fehlers während der Simulation ermöglicht (siehe Abschnitt 5.3). Diese Parameter sind der Fehlerwiderstand, die Fehlerdistanz und der Phasenwinkel der Spannung im Zeitpunkt des Kurzschlusses. Mit dieser Lösung wurde die größte Schwäche des Real-Time-Simulators OP5600, nämlich die hohe Kompilierungszeit beseitigt. Wenn man einen Parameter während der Simulation manuell ohne diese Lösung ändern möchte, muss er den Simulationsvorgang stoppen, die bestimmten Parameter im Modell ändern und das Modell neukompilieren, welche Prozesse bei einer großzähligen Untersuchung sehr viel Zeit verbrauchen.

Nach dem Umbau des Modells können während der Simulation zwischen fünf Fehlerwiderständen und elf Fehlerorte gewählt werden bzw. der Phasenwinkel kann stufenlos zwischen 0° und 359° umgestellt werden. Wenn nur vier Phasenwinkeleinstellungen genommen werden, können 220 unterschiedliche Fehlersimulation mit einer Kompilierung durchgeführt werden. Wenn man mit einer allgemeinen Kompilierungs- und Hochladenzeit je 2 Minuten und 1,5 Sekunde pro Fehlersimulation berechnet, kann die folgende Gesamtzeit mit manueller Parameteränderung erhalten werden:

$$T_{\Sigma,mauelle} = 220 \cdot (T_{Komp.} + T_{Lade.} + T_{Sim.}) = 53\,130\,s = 14,76\,h \quad (8.1)$$

Diese Zeit ist bei einem automatisierten Simulationsvorgang mit Parameteränderung während der Simulation:

$$T_{\Sigma,auto} = T_{Komp.} + T_{Lade.} + T_{Warte} + 220 \cdot T_{Sim.} = 575\,s = 9,58\,min \quad (8.2)$$

Es ist sichtbar, dass die Messungen fast 100-fach schneller durchgeführt werden können, dank den Änderungen. Diese Geschwindigkeit kann schwieriger mit der Technologie von CMC 356 verglichen werden, da die Zeitdauer der Offline-Simulation im zweiten Fall stark von den Eigenschaften des verwendeten PCs abhängt. Aber man kann feststellen, dass die Gesamtdauer in diesem Fall mindestens zweimal länger ist als bei der Simulation mit OP5600, da der Störverlauf erstens im Simulink vorsimuliert werden muss und erst dann kann die Relaisprüfung durchgeführt werden. Hier wird es sogar mit der Dauer der Datenspeicherung und Konvertierung nicht gerechnet.

Die vorgestellte Technologie hat aber ebenfalls Nachteile. Je komplexer das Modell ist, desto höher ist die Wahrscheinlichkeit des Vorkommens von overruns während der Simulation. Im kritischen Fall können diese overruns zu der Unterbrechung der Simulation führen. Um diesen Fall vermeiden zu können, muss die Samplezeit vergrößert oder die Simulation optimiert werden (siehe Kapitel 7). Aber die unterschiedlichen Optimierungsmethoden können in ein schon existierendes Modell, wie es schon erwähnt wurde, nicht immer implementiert werden.

Ein weiterer Nachteil dieser Technologie ist ihre Empfindlichkeit. Ein Simulink-Modell muss nach den Vorschriften von OPAL-RT sehr präzise vorbereitet werden, sonst kann es das Programm RT-LAB nicht kompilieren. Leider erhält man bei einem Kompilierungsfehler oft keine eindeutige Beschreibung über die Ursache des Problems.

Mit der Kenntnis der Vor- und Nachteile dieser Real-Time-Technologie kann man konstatieren, dass es sich lohnt, diesen Real-Time-Simulator weiter zu untersuchen und den Prüfstand für die Relaiseinstellung oder die Untersuchung der Netzerscheinungen zu verwenden.

Literaturverzeichnis

- [1] G. Wolfgang, *Energieversorgung - Skriptum zur Vorlesung*, Wien: Technische Universität Wien, 2017.
- [2] L. Hofmann, *Elektrische Energieversorgung Band 3*, Berlin: Walter de Gruyter GmbH, 2019.
- [3] E. Schmidt, „Skriptum zur Vorlesung Elektrische Maschinen,“ TU Wien, Wien, 2020.
- [4] D. Oeding und B. R. Oswald, *Elektrische Kraftwerke und Netze*, Deutschland: Springer, 2011.
- [5] G. Balzer, *Kurzschlussströme in Drehstromnetzen*, Deutschland: Springer Vieweg, 2020.
- [6] G. Wolfgang und G. Brauner, *Energieübertragung und Hochspannungstechnik Univ.-Skriptum*, Wien, Österreich, 2016.
- [7] R. Marenbach, D. Nelles und C. Tuttas, „Netzschutz,“ in *Elektrische Energietechnik*, Deutschland, Springer Vieweg, 2013, pp. 418-432.
- [8] G. Ziegler, *Digitaler Distanzschutz: Grundlagen und Anwendungen*, Deutschland: Publicis Corporate Publishing, Erlangen, 2008.
- [9] M. Wurm, *Skriptum zur LU 370.024: "Schutztechnik"*, Wien: Technische Universität Wien, 2018.
- [10] C. Moler und J. Little, „A History of MATLAB,“ *Proceedings of the ACM on Programming Languages*, Bd. 4, p. Article 81, June 2020.
- [11] „MathWorks.com,“ MathWorks, [Online]. Available: https://www.mathworks.com/?s_tid=gn_logo. [Zugriff am 15.12.2020].
- [12] Hydro-Québec Trans Énergie Technologies, *SimPowerSystems User's Guide V.3.*, The MathWorks, Inc., 2003.
- [13] J. Bélanger und L. Laforce, „opal-rt.com,“ 2019. [Online]. Available: https://blob.opal-rt.com/medias/L00161_0481.pdf. [Zugriff am 03.01.2021].

- [14] J. Bélanger, P. Venne und J.-N. Paquin, „The What, Where and Why of Real-Time Simulation,“ *Planet RT*, pp. 37 - 49, 01 01 2010.
- [15] Opal-RT, RT-LAB Version 10.1 User Guide, Canada: Opal-RT Technologies Inc., 2005.
- [16] „wiki.opal-rt.com,“ OPAL-RT, [Online]. Available: <https://wiki.opal-rt.com/display/HDGD/OP5600V2+Hardware+Interface>. [Letzter Zugriff am 21.01.2021].
- [17] Doble Engineering Company, „DOBLE PROTECTION TESTING F6350e External Amplifier,“ Doble Engineering Company, Watertown, Massachusetts, USA, 2014.
- [18] Siemens, „siemens.com,“ Siemens, [Online]. Available: <https://new.siemens.com/global/de/produkte/energie/energieautomatisierung-und-smart-grid/schutztechnik/siprotec-5/leitungsdifferential-und-distanzschutz/leitungsdifferential-und-distanzschutz-siprotec7sl87.html>. [Letzter Zugriff am 21.01.2021].
- [19] OMICRON electronics, „CMC 356 Unser Manual,“ 2003.
- [20] Siemens, Handbuch SIPROTEC 5 V8.01, 2019.
- [21] OPAL-RT, *Application-Oriented Course*, 2019.
- [22] Company, Doble Engineering, „F6 Multiple Amplifier Configurator Version 4 User Guide,“ Doble Engineering Company, Watertown, Massachusetts, USA, 2014.

Abbildungsverzeichnis

Abbildung 1.1: Original Modell des Windparks	2
Abbildung 2.1: Modell eines Kurzschlussvorganges [2]	5
Abbildung 2.2: Stromverlauf bei einem generatornahen Kurzschluss [4]	7
Abbildung 2.3: Stromverlauf bei einem generatorfernen Kurzschluss [4]	7
Abbildung 2.4: Einpoliger Kurzschluss im starr geerdeten Netz [1]	9
Abbildung 2.5: Einpoliger Kurzschluss im isolierten Netz [1]	11
Abbildung 2.6: Spannungsverlagerung bei einpoligem Erdschluss [1]	12
Abbildung 2.7: Zweipoliger Kurzschluss ohne Erdberührung [1]	13
Abbildung 2.8: Netz mit Distanzschutz [8]	15
Abbildung 2.9: Darstellung der Distanzgrößen in der komplexen Impedanzebene [9]	16
Abbildung 2.10: Einfluss des Lichtbogenwiderstandes [9]	17
Abbildung 2.11: Polygon-Charakteristik [9]	17
Abbildung 2.12: U/I – Kennlinie [8]	18
Abbildung 3.1: Grundlegender Messaufbau	20
Abbildung 4.1: Art von fixed time-step Simulationen [14]	24
Abbildung 4.2: OP5600 Frontschnittstelle [16]	26
Abbildung 4.3: OP5600 rückseitige Schnittstelle [16]	27
Abbildung 4.4: Quellenbezeichnungen für die 12 75VA-Quellen [22]	28
Abbildung 4.5: CMC 356 Frontseite [19]	29
Abbildung 5.1: Subsystemen im Windparkmodell	31
Abbildung 5.2: Subsystem SC_GUI	32
Abbildung 5.3: Fehlersimulation	33
Abbildung 5.4: Schematische Darstellung von Messsystem	34
Abbildung 5.5: Fehlermodell	36
Abbildung 5.6: Fault-Subsystem	36
Abbildung 5.7: Kurzschlussverlaufen mit unterschiedlichen Phasenwinkeln	37
Abbildung 5.8: Darstellung des Zeigermodelles	38
Abbildung 5.9: Subsystem Angle_Measurement	39
Abbildung 5.10: Subsystem Switch	39
Abbildung 5.11: Möglichkeiten der Fehlerpositionierung	41

Abbildung 5.12: Subsystem Array 32	43
Abbildung 5.13: Ablaufdiagramm des Python-Skripts	44
Abbildung 6.1: Messergebnisse	49
Abbildung 6.2: Phasenverlauf der Spannungen	50
Abbildung 6.3: Simulationsansicht von Test-Universe.....	51
Abbildung 6.4: Kurzschlussverlauf des zweipoligen Kurzschlusses (OP5600-Simulation) ...	52
Abbildung 6.5: Kurzschlussverlauf des zweipoligen Kurzschlusses (CMC 356-Test)	53
Abbildung 6.6: Phasor-Diagramm des zweipoligen Kurzschlusses (OP5600-Simulation).....	53
Abbildung 6.7: Phasor-Diagramm des zweipoligen Kurzschlusses (CMC 356-Test).....	54
Abbildung 6.8: Phasor-Diagramm des einpoligen Kurzschlusses im Fall mit Erdungstransformator	55
Abbildung 6.9: Kurzschlussverlauf des einpoligen Kurzschlusses (OP5600-Simulation).....	57
Abbildung 6.10: Phasor-Diagramm des einpoligen Kurzschlusses (OP5600-Simulation).....	57
Abbildung 6.11: Kurzschlussverlauf des einpoligen Kurzschlusses (CMC 356-Test)	58
Abbildung 6.12: Phasor-Diagramm des einpoligen Kurzschlusses (CMC 356-Test)	58
Abbildung 7.1: Monitoring-Panel	60
Abbildung 7.2: Matrixreduktion durch ARTEMiS [21]	61
Abbildung 7.3: Ersatzschaltbild von Three-Phase PI Section Line [11].....	62

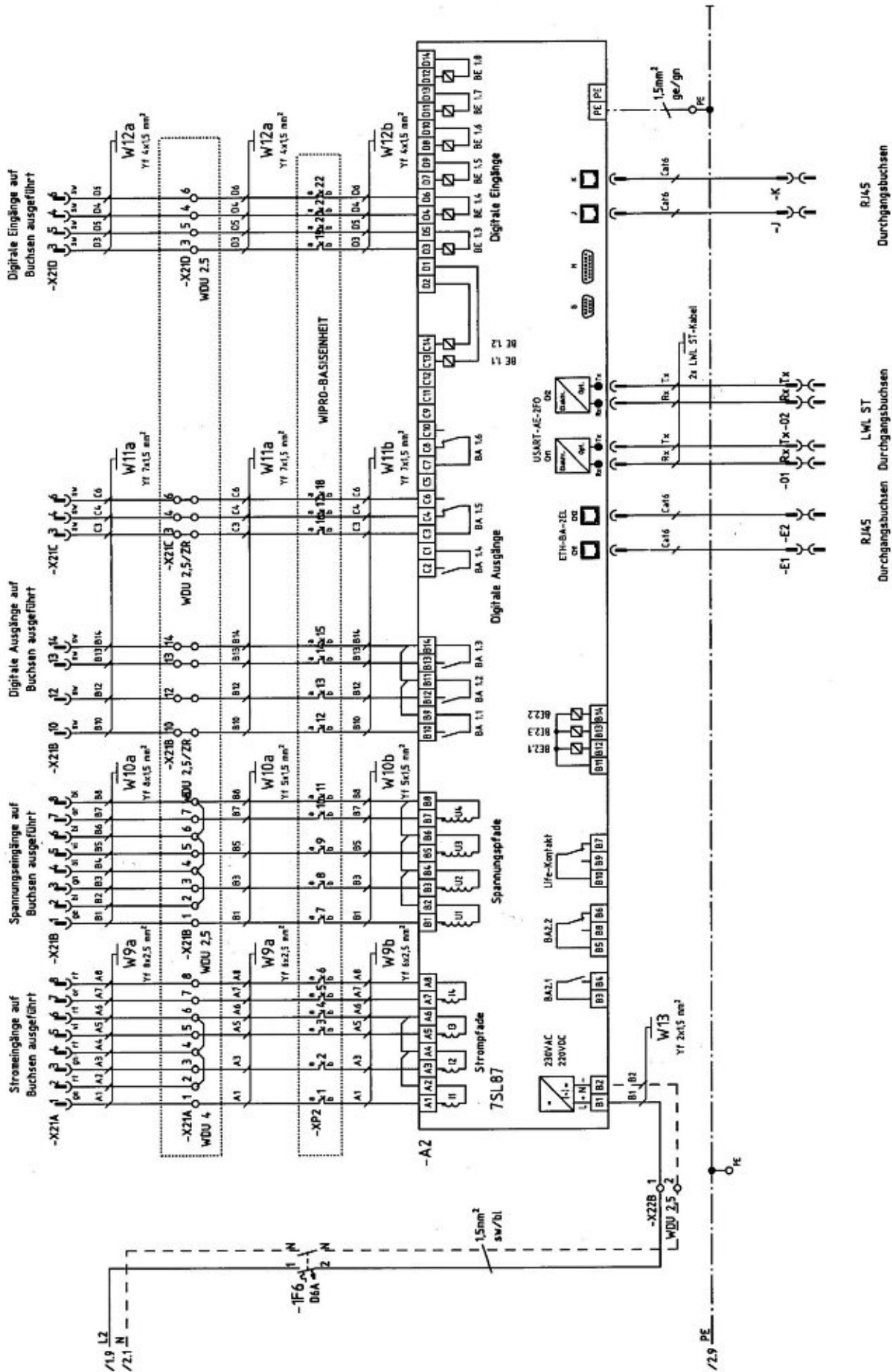
Tabellenverzeichnis

Tabelle 4.1: Benennungsvorschriften der Subsysteme	25
Tabelle 4.2: Liste die Eigenschaften von OP5600 [16]	26
Tabelle 4.3: Einstellungen des Verstärkers	28
Tabelle 6.1: Messergebnisse	51

Abkürzungsverzeichnis

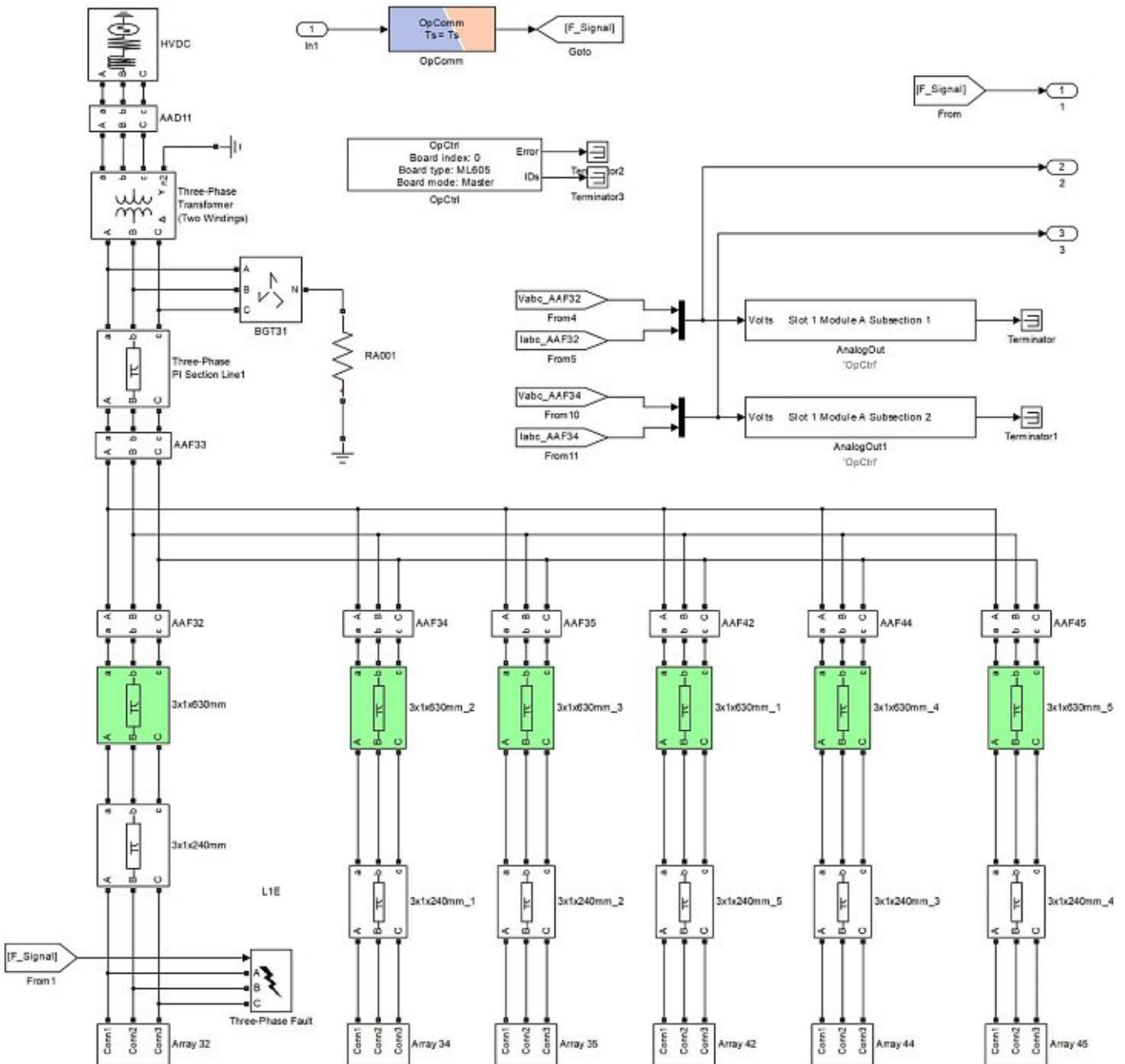
ARTEMiS	Advanced Real-Time Electro-Magnetic Solvers
DPL	Distributed Parameters Lines
GUI	Graphical User Interface
SPS	SimPowerSystem
SSN	State Space Nodal

Anhang A.: Schaltplan der Testeinrichtung von SIPROTEC 7SL87



Anhang B.: Subsystem SM_MASTER V.1.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

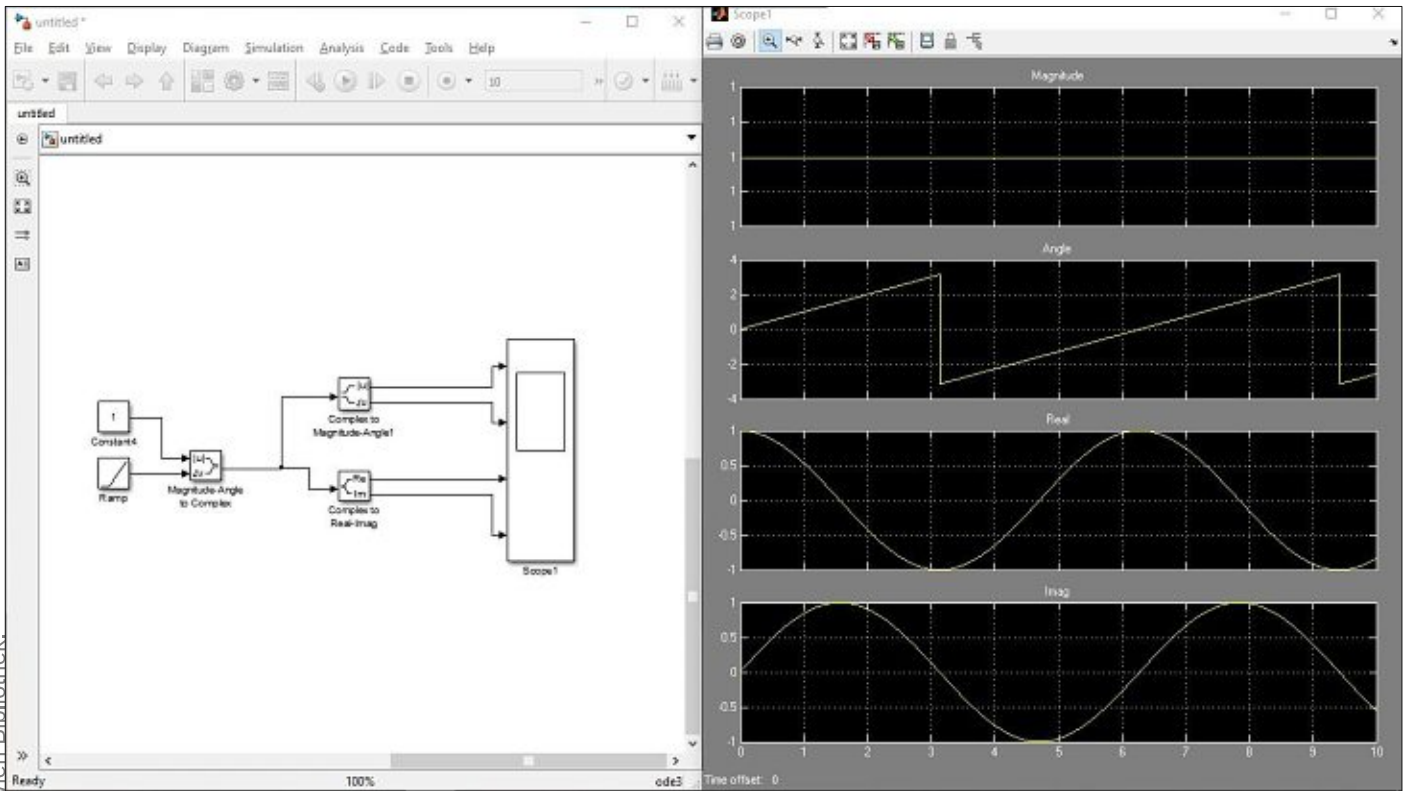


Anhang C.: Kommunikationskanälen

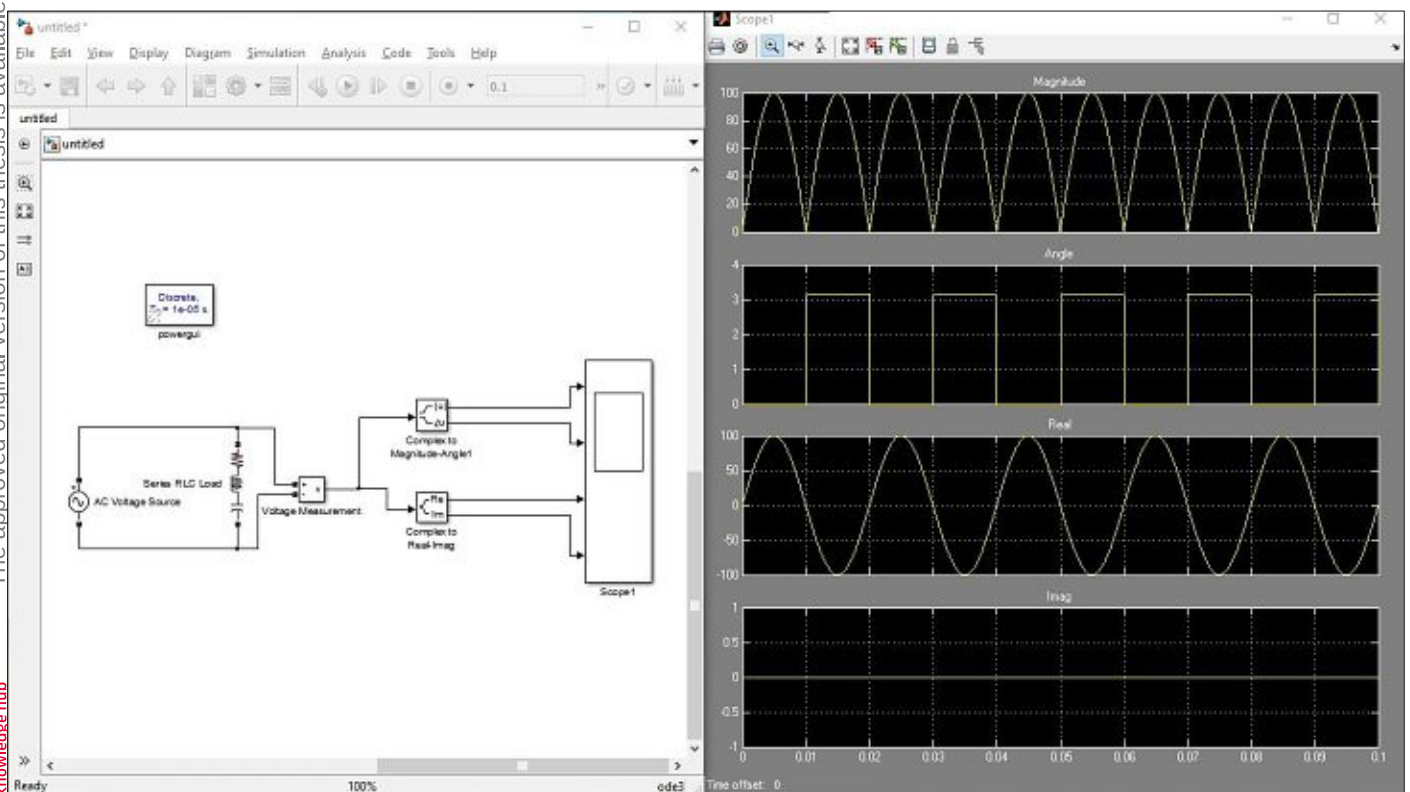
Host PC	OP5600			Doble F6350e			Siemens SIPROTEC 7SL87
Messsignal	Port	Kanal	Pins	Pin	Kanal	Quelle	Buchse
Va_AAF32	1	A0	1 - 20	12	Amp1-Ch1	VA	B1 (SR2)
Vb_AAF32	1	A1	2 - 21	13	Amp2-Ch1	VB	B3 (SR2)
Vc_AAF32	1	A2	3 - 22	14	Amp3-Ch1	VC	B5 (SR2)
Ia_AAF32	1	A3	4 - 23	9	Amp4-Ch1	I1	A1 (SR2)
Ib_AAF32	1	A4	5 - 24	10	Amp5-Ch1	I2	A3 (SR2)
Ic_AAF32	1	A5	6 - 25	11	Amp6-Ch1	I3	A5 (SR2)
Va_AAF34	2	A8	9 - 28	5	Amp1-Ch2	VR	B1 (SR1)
Vb_AAF34	2	A9	10 - 29	6	Amp2-Ch2	VS	B3 (SR1)
Vc_AAF34	2	A10	11 - 30	7	Amp3-Ch2	VT	B5 (SR1)
Ia_AAF34	2	A11	12 - 31	2	Amp4-Ch2	I4	A1 (SR1)
Ib_AAF34	2	A12	13 - 32	3	Amp5-Ch2	I5	A3 (SR1)
Ic_AAF34	2	A13	14 - 33	4	Amp6-Ch2	I6	A5 (SR1)
						Erde	A8 – B8 (SR1 und SR2)
Breaker_Status1	3	D0	1 – 20				B14 (SR2)
Breaker_Status2	3	D1	2 - 21				B14 (SR1)
5V- Einspeisung							B13 (SR1 und SR2)

Anhang D.: Winkelmessungsproblem

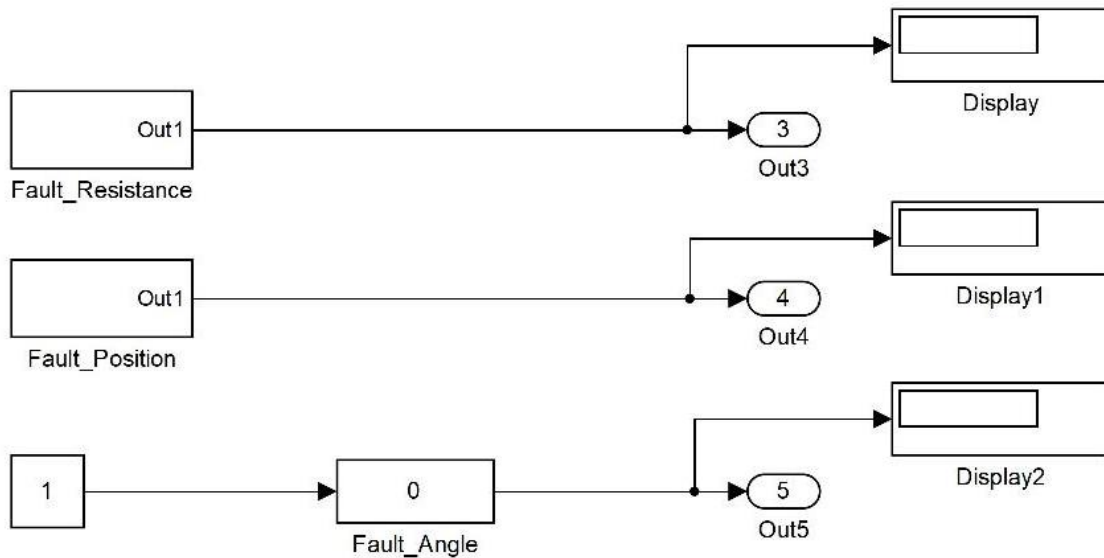
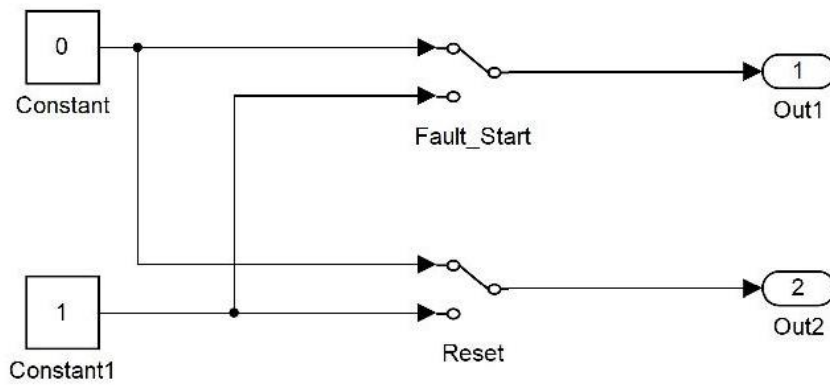
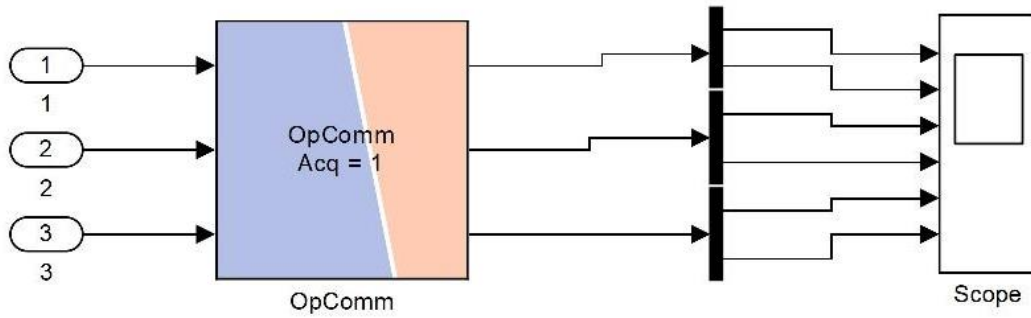
A. Bildung eines rotierenden Zeigers im Simulink



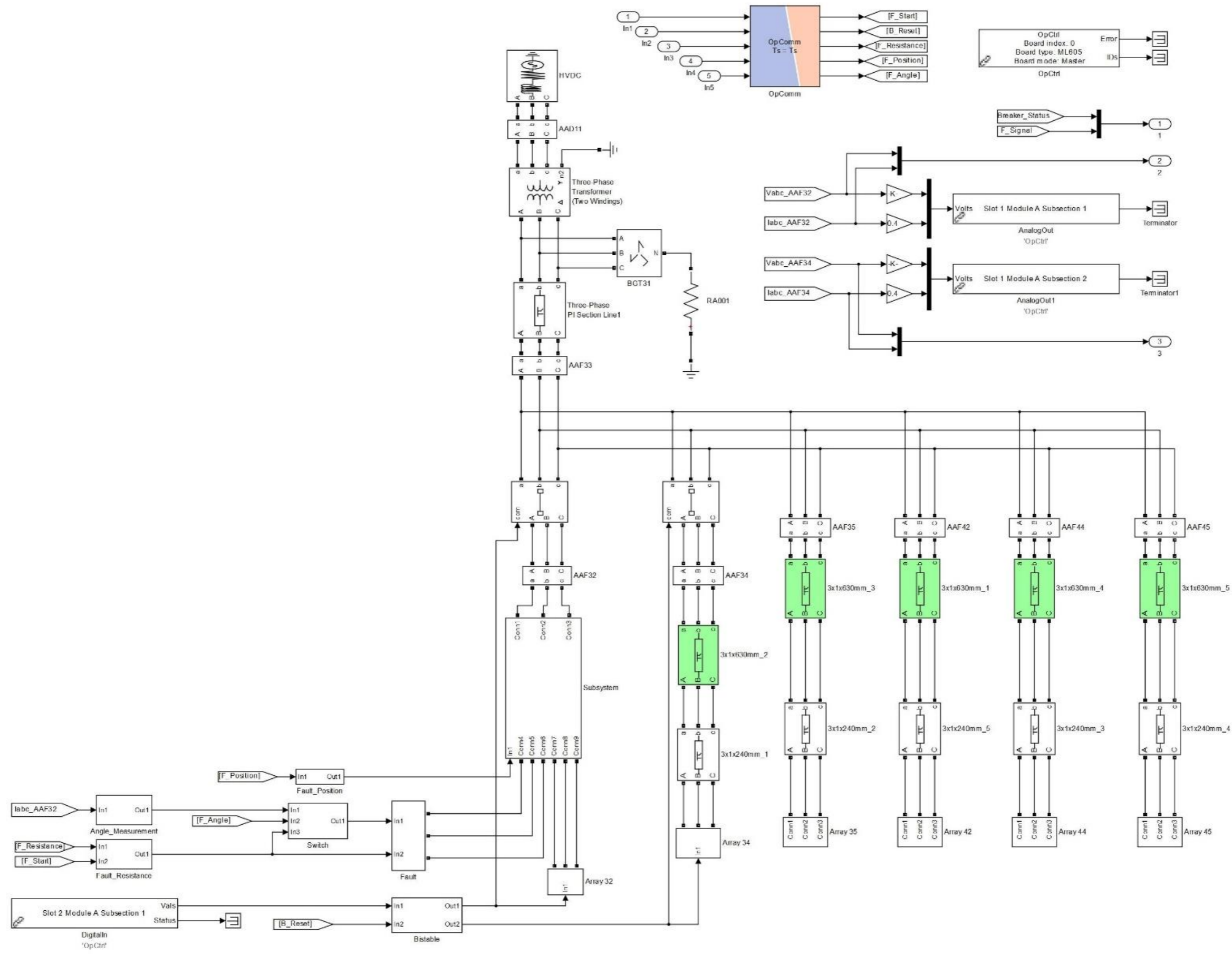
B. Komplexe Spannungsmessung im Simulink



Anhang E.: Subsystem SC_GUI V.2



Anhang F.: Subsystem SM_MASTER V2



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
 The approved original version of this thesis is available in print at TU Wien Bibliothek.



Anhang G.: Python-Skript V.1.

```
import RtlabApi
import time
import os
import shutil

Windpark = 'C:\Users\nagy\OPAL-RT\RT-LAB_Workspace\Breaker_Test\Breaker_Test.llp'
path = 'C:\Users\nagy\OPAL-RT\RT-
LAB_Workspace\Breaker_Test\models\Breaker_Test\Results'

try:
    os.mkdir(path)
except OSError:
    print ("Creation of the directory %s failed" % path)
else:
    print ("Successfully created the directory %s " % path)

#Switch 0-1
Switch = 'Breaker_Test/SC_GUI/port2'
#Breaker reset 0-1
Reset = 'Breaker_Test/SC_GUI/port3'
#Fault resistance 0-4
Resistance = 'Breaker_Test/SC_GUI/port1'
#Fault position 0- 10
Position = 'Breaker_Test/SC_GUI/port4'
#Fault angle 0-90
Angle = 'Breaker_Test/SC_GUI/port5'

RtlabApi.OpenProject(Windpark)
print 'The project is open'

realTimeMode = RtlabApi.HARD_SYNC_MODE
RtlabApi.Load(realTimeMode,1)
print 'The model is loaded'

RtlabApi.Execute(1)
print 'The model is executing'
time.sleep(3)
RtlabApi.GetParameterControl(1)
RtlabApi.GetAcquisitionControl(1)

varName = 'plant_response'

for i in range(5):
    R = i*2
    RtlabApi.SetParametersByName(Resistance,1)
    print 'Fault resistance: ',R, ' Ohm'

    for x in range(11):
        P = x*10
        RtlabApi.SetParametersByName(Position,x)
        print 'Fault position from wind farm: ',P, ' %'

    for y in range(4):
        A = y*30
        RtlabApi.SetParametersByName(Angle,60)
        print 'Phase angle: ',A, ' (degree)'
        fileName = 'R'+str(R) + 'P'+str(P) + 'A'+str(A)
        RtlabApi.SetAcqWriteFile(0, fileName, varName, 1)
        print (fileName)
        time.sleep(0.5)
        RtlabApi.SetParametersByName(Switch,1)
        print 'Switch = 1'
```

```

time.sleep(0.5)
RtlabApi.SetParametersByName(Switch,0)
print 'Switch = 0'
RtlabApi.SetParametersByName(Reset,1)
print 'Reset = 1'
time.sleep(0.5)
RtlabApi.SetParametersByName(Reset,0)
print 'Reset = 0'
RtlabApi.SetAcqWriteFile(0, fileName, varName, 0)

RtlabApi.GetParameterControl(0)
RtlabApi.GetAcquisitionControl(0)

RtlabApi.Reset()
print 'The model has been reset'

for i in range(5):
    R = i*2

    for x in range(11):
        P = x*10

        for y in range(4):
            A = y*30
            fileName = 'R'+str(R) + 'P'+str(P) + 'A'+str(A)
            oldpath = 'C:\Users\nagyz\OPAL-RT\RT-
LAB_Workspace\Windpark\models\Test_Windpark_5_3\'+ fileName + '.mat'
            newpath = 'C:\Users\nagyz\OPAL-RT\RT-
LAB_Workspace\Windpark\models\Test_Windpark_5_3\Results\'+ fileName + '.mat'
            shutil.move(oldpath, newpath)

RtlabApi.CloseProject()
print 'Finish'

```

Anhang H.: Python-Skript V.2.

```
import RtlabApi
import time

Windpark = 'C:\Users\nagy\OPAL-RT\RT-LAB_Workspace\Breaker_Test\Breaker_Test.llp'
#Switch 0-1
Switch = 'Breaker_Test/SC_GUI/port2'
#Breaker reset 0-1
Reset = 'Breaker_Test/SC_GUI/port3'
#Fault resistance 0-4
Resistance = 'Breaker_Test/SC_GUI/port1'
#Fault position 0- 10
Position = 'Breaker_Test/SC_GUI/port4'
#Fault angle 0-90
Angle = 'Breaker_Test/SC_GUI/port5'

signalName = 'Breaker_Test/SM_MASTER/port3'
signalValue = 0

RtlabApi.OpenProject(Windpark)
print 'The project is open'

realTimeMode = RtlabApi.HARD_SYNC_MODE
RtlabApi.Load(realTimeMode,1)
print 'The model is loaded'

RtlabApi.Execute(1)
print 'The model is executing'
time.sleep(3)
RtlabApi.GetParameterControl(1)
RtlabApi.GetSignalControl(1)
RtlabApi.GetAcquisitionControl(1)
RtlabApi.SetSignalsByName(Resistance,1)
RtlabApi.SetSignalsByName(Position,2)
print 'Position Angle Zone1 Zone2'

for x in range(11):
    A = x * 30
    RtlabApi.SetSignalsByName(Angle,A)
    Z1 = 0
    Z2 = 0
    for i in range(50):
        time.sleep(1)
        RtlabApi.SetSignalsByName(Switch,1)
        time.sleep(0.1)
        signalValue = RtlabApi.GetSignalsByName(signalName)
        if signalValue == 0:
            Z1 = Z1 + 1
        else:
            Z2 = Z2 + 1
        time.sleep(0.4)
        RtlabApi.SetSignalsByName(Switch,0)
        RtlabApi.SetSignalsByName(Reset,1)
        time.sleep(0.5)
        RtlabApi.SetSignalsByName(Reset,0)

RtlabApi.GetParameterControl(0)
RtlabApi.GetSignalControl(0)
RtlabApi.GetAcquisitionControl(0)
RtlabApi.Reset()
print 'The model has been reset'
RtlabApi.CloseProject()
print 'Finish'
```

Eidesstattliche Erklärung

Hiermit erkläre ich, dass die vorliegende Arbeit gemäß dem Code of Conduct, insbesondere ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel, angefertigt wurde. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.

Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder in ähnlicher Form in anderen Prüfungsverfahren vorgelegt.

Wien, am 05.04.2021


Zoltán Simon Nagy