



MASTERARBEIT

Herr B.Sc.
J r mie Foulquier

**Entwicklung eines
echtzeitf higen Algorithmus
zur Ermittlung
und Kompensation
der Netzberschwingungen
und Implementierung
im Rechner
der „Intelligenten Lades ule“**

Mittweida, 2013

MASTERARBEIT

Entwicklung eines echtzeitfähigen Algorithmus zur Ermittlung und Kompensation der Netzüberschwingungen und Implementierung im Rechner der „Intelligenten Ladesäule“

Autor:

Herr B.Sc Jérémie Foulquier

Studiengang:

Elektrotechnik

Seminargruppe:

ET12sE-M

Erstprüfer:

Prof. Dr.-Ing. Lutz Rauchfuß

Zweitprüfer:

Prof. Dr.-Ing. Swen Schmeißer

Einreichung:

Mittweida, 22.07.2013

Verteidigung/Bewertung:

Mittweida, 2013

MASTER THESIS

Development of a real-time algorithm for the measurement and the compensation of harmonics of the grid and implementation in the processor of the intelligent charging station

author:

Mr. B.Sc. Jérémie Foulquier

course of studies:

Electrical Engineering

seminar group:

ET12sE-M

first examiner:

Prof. Dr.-Ing. Lutz Rauchfuß

second examiner:

Prof. Dr.-Ing. Swen Schmeißer

submission:

Mittweida, 22.07.2013

defence/ evaluation:

Mittweida, 2013

Bibliografische Beschreibung:

Foulquier, Jeremie:

Entwicklung eines echtzeitfähigen Algorithmus zur Ermittlung und Kompensation der Netzoberschwingungen und Implementierung im Rechner der „Intelligenten Ladesäule“. - 2013. - X, 86, 45 S.

Mittweida, Hochschule Mittweida, Fakultät Elektro- und Informationstechnik, Masterarbeit, 2013

Referat:

Zur Versorgung der von der Bundesregierung angestrebten Flotte von Elektrofahrzeugen muss die entsprechende Infrastruktur geschaffen werden. Daraus ergibt sich die Möglichkeit den Energiefluss zwischen Netz und Batterie des Elektrofahrzeuges gezielt über eine ansteuerbare Ladesäule zu beeinflussen. Die Intelligente Ladesäule ist dauerhaft mit dem Netz verbunden und kann für den Netzbetreiber ein dezentrales Stellglied zur Beeinflussung des Energieflusses sein und zusätzlich zur Verbesserung der Netzspannungsqualität beitragen. Die vorliegende Arbeit befasst sich daher mit der Funktionsentwicklung zur „Kompensation von Netzoberschwingungen“ mit Hilfe des Wechselrichters der Intelligenten Ladesäule. Das Hauptziel ist die Entwicklung eines echtzeitfähigen Algorithmus zur Oberschwingungserfassung, die der Regelung als Istwerte dienen, um über das phasennegierte Einprägen der Ströme, die Spannungsüberschwingungen im Netz zu kompensieren.

Aufgabenstellung Masterarbeit für Jeremie Foulquier

Thema: Entwicklung eines echtzeitfähigen Algorithmus zur Ermittlung und Kompensation der Netzoberschwingungen und Implementierung im Rechner der „Intelligenten Ladesäule“

Mittweida, 01.03.2013

Die „Intelligente Ladesäule“ dient E-Fahrzeugen zur Aufladung der Batterie. Sie ist als Schnittstelle zwischen dem Verteilernetz des Netzbetreibers und dem E-Fahrzeug angeordnet. Daraus ergibt sich die Möglichkeit zur definierten Bereitstellung eines Gleichstromes zum Aufladen der Batterie, aber auch zur gezielten Beeinflussung der Netzspannungsqualität.

Arbeitspakete:

1. Entwicklung und Simulation der vollständigen Regelstruktur zur Erfassung und Kompensation von Oberschwingungen der Netzspannung
 - Erweiterung der feldorientierten Regelung der Intelligenten Ladesäule, um die Oberschwingungserfassung
 - Simulation der Oberschwingungskompensation für konstante Netzimpedanz
 - Simulation bei vollständiger und bei Teilkompensation einer Spannungsüberschwingung
2. Implementierung der entwickelten Regelstruktur in den Rechner (DSP) der Intelligenten Ladesäule
 - Programmierung des echtzeitfähigen Algorithmus
 - Inbetriebnahme der vollständigen Regelstruktur
 - Test der Oberschwingungskompensation für die Netzspannung am Labornetz
 - Programmierung der Softwaremodule als C-Code im Zahlenformat integer, dabei steht die Wiederverwertbarkeit der Module im Vordergrund (sinnvolle Aufteilung der Softwaremodule nach Funktionen, Kommentare beschreiben den physikalischen Ablauf)

Hochschulbetreuer:



Prof. Dr.-Ing. Lutz Rauchfuß

Inhalt

Inhalt	I
Abbildungsverzeichnis	IV
Tabellenverzeichnis	VII
Abkürzungsverzeichnis	VIII
0 Übersicht	1
0.1 <i>Motivation</i>	1
0.2 <i>Zielsetzung</i>	1
0.3 <i>Kapitelübersicht</i>	2
1 Theoretische Grundlagen	3
1.1 <i>Komplexe Zeiger im Dreiphasensystem</i>	3
1.1.1 Komplexe Zeiger	3
1.1.2 Raumzeiger und Clark Transformation	4
1.1.3 Nullsystem, Mitsystem und Gegensystem	5
1.1.4 Park Transformation	7
1.2 <i>Mittelwertbildung</i>	8
1.3 <i>Spektralanalyse</i>	10
1.3.1 Fourierreihenentwicklung	10
1.3.2 Gleitende Diskrete Fourier Transformation	14
1.4 <i>Oberschwingungsklassifizierung nach der symmetrischen Komponente</i>	16
1.5 <i>Zusammenfassung</i>	16
2 Normen	17
2.1 <i>Oberschwingungsgrenzwerte</i>	17
2.1.1 Verträglichkeitspegel für Oberschwingungsspannungen in öffentlichen Niederspannungsnetzen	17
2.1.2 Grenzwerte für Oberschwingungsströme	18
2.2 <i>Oberschwingungsmessverfahren DIN 61000-4-7</i>	20
2.2.1 Gruppierungsverfahren für die Auswertung der Oberschwingungsströme	20
2.2.2 Oberschwingungsuntergruppe für die Auswertung der Oberschwingungsspannung	22

2.3	<i>Zusammenfassung</i>	23
3	Theorie der Oberschwingungskompensation	24
3.1	<i>Entstehung der Oberschwingungsspannungen</i>	24
3.2	<i>Reduzierung einer Spannungsüberschwingung durch Einprägung eines Kompensationsstromes</i>	27
3.3	<i>Netzimpedanz im Niederspannungsnetz</i>	29
3.4	<i>Wirkung der Oberschwingungskompensation</i>	31
3.5	<i>Zusammenfassung</i>	33
4	Regelung der Oberschwingungskompensation	34
4.1	<i>Zweipulswechselrichter</i>	34
4.2	<i>Regelung der Grundschiwingung</i>	38
4.3	<i>Regelung der Stromüberschwingung</i>	40
4.4	<i>Regelung der Oberschwingungskompensation</i>	42
4.4.1	50Hz Filter	43
4.4.2	Messung der Oberschwingungsspannung	44
4.4.3	Mittelwert der Phasenwinkel	46
4.4.4	Regelung für die Schätzung der Netzimpedanz	50
4.5	<i>Zusammenfassung</i>	50
5	Prototyp der Intelligenten Ladesäule	51
5.1	<i>Hardware</i>	51
5.1.1	Umrichter, Stelltransformator, Glättungs-drosseln	52
5.1.2	Strom- und Spannungswandler, A/D-Wandler	53
5.1.3	DSP	56
5.2	<i>Software</i>	57
5.2.1	Zahlenformat und Variablentyp	57
5.2.2	Hauptprogramm	60
5.2.3	Optimierung der Rechenzeit	62
5.2.4	Initialisierungsunterprogramm	64
5.2.5	Unterprogramme der diskreten Regelung	66
5.3	<i>Messergebnisse</i>	78
5.4	<i>Zusammenfassung</i>	84
6	Zusammenfassung und Ausblick	85
6.1	<i>Ergebnisse</i>	85
6.2	<i>Bewertung der Arbeit</i>	86
6.3	<i>Ausblick</i>	86

Anlagen.....	87
Literatur	133
Selbstständigkeitserklärung	135

Abbildungsverzeichnis

Abbildung 1: Darstellung der Cosinus- und Sinusfunktion in der komplexen Zahlenebene	4
Abbildung 2: Mitsystem	6
Abbildung 3: Gegensystem	6
Abbildung 4: Nullsystem.....	7
Abbildung 5: Amplitudengang des Mittelwertes und eines Tiefpasses 1. Ord. für $T = 20\text{ms}$	9
Abbildung 6: Phasengang des Mittelwertes und eines Tiefpasses 1. Ord. für $T = 20\text{ms}$	9
Abbildung 7: Fourierreihenentwicklung: 1. Beispielsignal	11
Abbildung 8: Fourierreihenentwicklung: Verschiebung des Spektrums	11
Abbildung 9: Fourierreihenentwicklung: Berechnung des Gleichanteils.....	12
Abbildung 10: Fourierreihenentwicklung: 2. Beispielsignal	12
Abbildung 11: Fourierreihenentwicklung: Abbruchfehler.....	13
Abbildung 12: 2kHz Sinus mit verschiedenen Abtastfrequenzen	14
Abbildung 13: gleitende DFT.....	15
Abbildung 14: Gruppierungsverfahren DIN 61000-4-7 zur Auswertung der Oberschwingungsströme	21
Abbildung 15: Spektrum der Gruppierungsverfahren (DIN 61000-4-7) und im Vergleich eine 50Hz Si-Funktion zur Auswertung der Frequenzkomponente f_0	21
Abbildung 16: Oberschwingungsuntergruppe DIN EN 61000-4-7 zur Auswertung der Harmonischen der Netzspannung.....	22
Abbildung 17: Spektrum der Oberschwingungsuntergruppe (DIN 61000-4-7) und im Vergleich eine 50Hz Si-Funktion zur Auswertung der Frequenzkomponente f_0	23

Abbildung 18: Zweiweg-Gleichrichter mit kapazitiver Glättung	24
Abbildung 19: Oberschwingungsstrom eines Zweiweg-Gleichrichters	25
Abbildung 20: Verbraucherzählpeilsystem.....	25
Abbildung 21: Einphasige Kompensation der Spannungsüberschwingung.....	27
Abbildung 22: Zeigerdiagramm einer optimalen Kompensation.....	28
Abbildung 23: Effektivität der Kompensation	29
Abbildung 24: Phasenwinkel der Bezugsimpedanz in Abhängigkeit von der Frequenz....	30
Abbildung 25: Messung der Netzimpedanz im Niederspannungsnetz nach [SCHU13]....	31
Abbildung 26: Wirkung der Oberschwingungskompensation	31
Abbildung 27: Amplitude der Oberschwingungen der Spannung u ohne Kompensation..	32
Abbildung 28: dreiphasiger Zweipulswechselrichter der Intelligenten Ladesäule	34
Abbildung 29: Raumzeigerdarstellung der Wechselrichterausgangsspannungen.....	35
Abbildung 30: Transistorpuls muster und Wechselrichter Ausgangsströme und – spannungen für zwei Taktperioden.....	36
Abbildung 31: netzgesteuerter Betrieb des Wechselrichters.....	37
Abbildung 32: Regelung der Grundschwingung des Wechselrichterstromes	38
Abbildung 33: Park-Transformation des Raumzeigers des Wechselrichterstromes	39
Abbildung 34: Regelung einer Stromüberschwingung	40
Abbildung 35: ruhender Zeiger einer Stromüberschwingung	41
Abbildung 36: Regelung der Oberschwingungskompensation.....	43
Abbildung 37: 50Hz Filter vor der Oberschwingungsmessung der Netzspannung.....	44
Abbildung 38: Filtereigenschaften der DFT bei 5Hz und 50Hz Auflösung.....	44
Abbildung 39: Bandpassfilter für die Messung der Spannungsüberschwingungen (Gruppierung +/- 10Hz)	45

Abbildung 40: Bandpassfilterung für die Messung der Spannungsüberschwingungen	46
Abbildung 41: DFT Messung des Phasenwinkels der 11. OS mit $f_{DFT} = 550\text{Hz}$, $f_{OS} = 554\text{Hz}$	46
Abbildung 42: Dämpfung der Messabweichungen des Phasenwinkels	47
Abbildung 43: Ableitung des Phasenwinkels = Kreisfrequenz	47
Abbildung 44: gleitender Mittelwert der Kreisfrequenz	48
Abbildung 45: Integration der gemittelten Kreisfrequenz → gemittelter Phasenwinkel.....	48
Abbildung 46: gleitender Mittelwert der Phasenkorrektur	49
Abbildung 47: Simulationsergebnisse der Korrektur und Filterung eines Phasenwinkels	49
Abbildung 48: Übersichtplan des Prototyps der Intelligenten Ladesäule	51
Abbildung 49: Amplitudenfrequenzgang der Impedanz Z_d	52
Abbildung 50: Alias-Effekt (Quelle: Wikipedia)	54
Abbildung 51: Filter-Komponente des A/D-Wandlers	54
Abbildung 52: Antialiasing-Filter des A/D-Wandlers: Amplitudenfrequenzgang	54
Abbildung 53: Zeitpunkt der Messanstöße zur Strommessung mittels A/D-Wandler [QUAN99]	55
Abbildung 54: Zeitablauf einer 100 μs Hauptprogrammschleife	61
Abbildung 55: Speicherung der Kehrwerte in der Divisionstabelle.....	62
Abbildung 56: Kompensation der 5. Spannungsüberschwingung des 0,4kV Netzes	79
Abbildung 57: Kompensation der 5. Oberschwingungsspannung des 0,4kV Netzes	80
Abbildung 58: Kompensation der 5. Oberschwingungsspannung des 0,4kV Netzes	82
Abbildung 59: Oberschwingungsgehalt des Wechselrichterstroms	83

Tabellenverzeichnis

Tabelle 1: DFT-Auswertung eines 2kHz Sinus mit Verschiedenen Abtastfrequenzen.....	15
Tabelle 2: Normen und Richtlinien der Oberschwingungsspannungen [TÜVS12].....	17
Tabelle 3: Verträglichkeitspegel für Oberschwingungsspannungen in öffentlichen Niederspannungsnetzen nach DIN EN 61000-2-2.....	18
Tabelle 4: Störaussendungsgrenzwerte nach DIN 61000-3-2 Gerät mit Eingangsstrom $\leq 16A$; λ : Leistungsfaktor.....	19
Tabelle 5: Störaussendungsgrenzwert nach DIN 61000-3-12 für dreiphasige symmetrische Geräte mit Eingangsstrom $< 75A$	20
Tabelle 6: Flussrichtung von Oberschwingungen	26
Tabelle 7: Phasenwinkel der Oberschwingungen eines Zweiweggleichrichters.....	26
Tabelle 8: Wirkung der Oberschwingungskompensation	32
Tabelle 9: Wertebereich der Eingangsdaten des A/D-Wandlers.....	53
Tabelle 10: Oversampling ratio des A/D-Wandlers	55
Tabelle 11: Portbelegung des DSP	56
Tabelle 12: angewendete Variablentypen in der DSP.....	57
Tabelle 13: Zahlenformat und Wertebereich der Messwerte.....	58
Tabelle 14: Bsp. Division durch Multiplikation ersetzen	62
Tabelle 15: Substitution der Funktion Arkustangens durch Tabellen	63
Tabelle 16: Erweiterung des Arkustangens auf 2π	64
Tabelle 17: Ergebnisse der Rechenzeitoptimierung.....	64
Tabelle 18: Programmablauf und Aktivität des Wechselrichters	66
Tabelle 19: Proportional- und Integralverstärkungen der PI-Regler	74

Abkürzungsverzeichnis

Bezeichnungsgrundsätze

Grundsätzlich gelten für die vorliegende Arbeit folgende Konventionen, die anhand der virtuellen Größe x erläutert werden:

x	-	skalare Größe
\underline{x}	-	komplexe Größe bzw. Zeiger
X	-	Effektivwert
\hat{x}	-	Amplitude
\bar{x}	-	Mittelwert

Wichtige Indizes

1	-	Grundschwingung
2, 3 ...	-	Oberschwingung
(0), (1), (2)	-	Null-, Mit- und Gegensystem
<i>DFT</i>	-	diskrete Fourier Transformation
d, q	-	rotierende Koordinatensystem
g	-	Grundschwingung
K	-	Kompensation
ν	-	Ordnung einer Harmonischen
N	-	Netz
n	-	Samplenummer
<i>OS</i>	-	Oberschwingung
<i>OSK</i>	-	kompensierte Oberschwingung

TP	-	Tiefpassfilter
u, v, w	-	Phasen u, v, w
WR	-	Wechselrichter
Zh	-	Zwischenharmonische
ZK	-	Zwischenkreis
α, β	-	festes Koordinatensystem

Zeiten, Frequenzen und allgemeine Symbole

\underline{a}	-	Phasenoperator
f	-	Frequenz
f_s	-	Abtastfrequenz
f_{DFT}	-	DFT-Analyse-Frequenz
f_{netz}	-	Netzfrequenz
\mathcal{F}	-	Fourier Transformation
h	-	Impulsantwort
Im	-	Imaginärteil
m	-	Mittelwert, Nachkommastellen
N	-	Anzahl der Sample in einer Periode T
n	-	Vorkommastellen
Re	-	Realteil
T	-	Periode
T_p	-	Taktperiode des Wechselrichters
z	-	diskrete Zeit
φ	-	Phasenwinkel
δ	-	Dirac-Impuls
θ	-	Drehwinkel
τ	-	Zeitkonstant

ω - Kreisfrequenz

Elektrische Größen

i - Strom

R_{sce} - Kurzschlussverhältnis

$S_{k,VP}$ - Kurzschlussleistung des Netzes am Verknüpfungspunkt

$S_{r,G}$ - Geräteleitung

u - Spannung

$\underline{u}_0 \dots \underline{u}_7$ - Standardvektoren

\underline{u}_s - Vektor der Raumzeigermodulation

U_{ZK} - Zwischenkreisspannung

\underline{Z}_N - Netzimpedanz

Abkürzungen

DFT - Diskrete Fourier Transformation

DIN - Deutsches Institut für Normung

EN - Europäische Norm

DSP - Digitaler Signalprozessor

Gl., Gln. - Gleichung(en)

IEC - International Electrotechnical Commission

IGBT - Insulated Gate Bipolar Transistor

NS - Niederspannung

OS - Oberschwingung

THD - Total Harmonic Distortion

TR - Transistor

VDE - Verband der Elektrotechnik, Elektronik und Informationstechnik

0 Übersicht

0.1 Motivation

Die politischen Ziele zur Verlangsamung des Klimawandels haben die Energiewende in Deutschland eingeleitet, damit verbunden sind die Steigerung der Energieeffizienz und das Anwachsen dezentral eingespeister Energie aus erneuerbaren Quellen. Der Einsatz von Leistungselektronik in Verbraucher- und Erzeugeranlagen verbessert die Energieeffizienz und macht die Einspeisung von umweltfreundlicher Energie möglich. Die Übertragung und Verteilung dieser dezentral erzeugten Energie ist für das elektrische Netz eine neue Herausforderung, die auch eine dezentrale Regelung der Spannung und des Energieflusses erfordert. Sie stellt damit eine Schlüsselaufgabe bei der Modernisierung des öffentlichen Energieversorgungsnetzes dar und führt unweigerlich zum Smart Grid.

Eng verbunden mit der Energiewende verfolgt die Bundesregierung das Ziel, bis zum Jahr 2020 eine Million Elektrofahrzeuge auf deutsche Straßen zu bringen. Durch E-Fahrzeuge, gespeist aus erneuerbarer Energie, sollen die CO₂-Emissionen drastisch reduziert werden. Die zukünftige E-Fahrzeugflotte stellt mit einer Speicherkapazität von mehreren Gigawattstunden eine interessante Möglichkeit zur Pufferung von Energieschwankungen dar [NPE12] und entlastet damit das elektrische Netz. Intelligente Ladesäulen verbinden die E-Fahrzeuge mit dem Energienetz. Durch bidirektionales Laden wird eine positive wie negative Reserveleistung zur Verfügung gestellt. Die Regulierung der Netzspannung durch dezentrale Wirk- und Blindleistungseinspeisung, aber auch die Kompensation von Oberschwingungen sind technische Herausforderungen, die mit Hilfe der „Intelligenten Ladesäule“ lösbar sind. Die Verbesserung der Netzspannungsqualität und Netzsicherheit mit Hilfe der „Intelligenten Ladesäule“ ist neben der Reduktion von CO₂-Emissionen ein weiterer Vorteil, der der E-Mobilität zum Durchbruch verhelfen kann.

0.2 Zielsetzung

Das Forschungsvorhaben mit dem übergeordneten Thema Elektromobilität, der Professur Antriebs- und Regelungstechnik, befasst sich mit der Entwicklung und Realisierung einer „Intelligenten Ladesäule“. Ein Prototyp der Ladesäule wurde durch mehrere Graduiierungsarbeiten an der Hochschule entwickelt. Die Funktionalität der Energieflusssteuerung und der Blindleistungskompensation sind am Prototyp bereits umgesetzt. In der vorliegenden Arbeit werden die Entwicklungsschritte zur Kompensation der Netzoberschwingungen mithilfe des Wechselrichters der Intelligenten Ladesäule beschrieben. Ein Meilenstein stellt dabei die Entwicklung eines echtzeitfähigen Algorithmus zur Oberschwingungserfassung und -kompensation dar.

0.3 Kapitelübersicht

Diese Masterarbeit enthält sechs Kapitel.

Kapitel 1 schildert die mathematischen Grundlagen, wie Raumzeiger, symmetrische Komponenten, Mittelwertberechnung und diskrete Fourier Transformation.

Das Kapitel 2 beschreibt die Normung im Bereich der Oberschwingungsmessverfahren und deren Oberschwingungsgrenzwerte.

In Kapitel 3 werden die Entstehung der Oberschwingungen im Niederspannungsnetz sowie die Oberschwingungskompensation beschrieben.

Detailliert erläutert Kapitel 4 die Steuerung des Wechselrichters der Intelligenten Ladesäule bei der Regelung der Blindleistung und Oberschwingungskompensation.

In Kapitel 5 werden die Hardware Komponenten des Prototyps der Intelligenten Ladesäule beschrieben und die Software für die Blindleistungsbereitstellung und Oberschwingungskompensation vorgestellt.

Abschließend werden in Kapitel 6 Ergebnisse diskutiert und ein Ausblick gegeben.

1 Theoretische Grundlagen

1.1 Komplexe Zeiger im Dreiphasensystem

1.1.1 Komplexe Zeiger

Ein Strom oder eine Spannung nach Gl. (1.1)

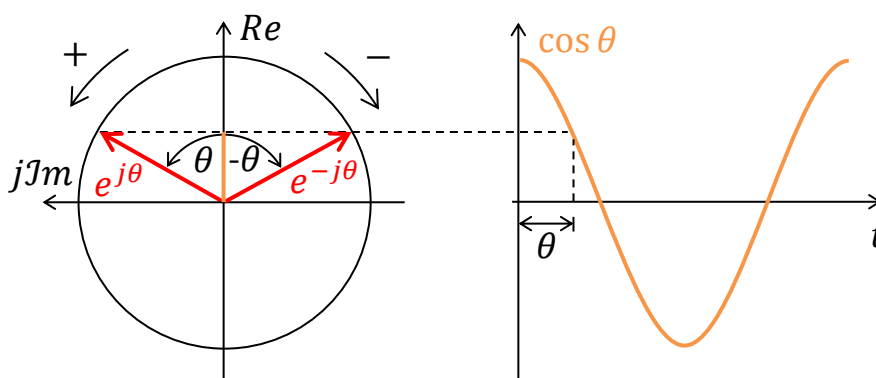
$$x(t) = \hat{x} \cdot \sin(\omega t + \varphi_x) \quad (1.1)$$

ist eine skalare Größe, deren Zeitverlauf mit einem Liniendiagramm dargestellt werden kann und durch differenzielle Gleichungen beschrieben wird. Die Lösung von differenziellen Gleichungen kann graphisch nicht erfolgen und muss mit dem Rechner ermittelt werden. Deshalb ist der Umgang mit differenziellen Gleichungen für den Menschen unübersichtlich und unpraktisch. Mit der Transformation von skalaren Größen in die komplexe Zahlenebene wird die aufwendige Integration und Ableitung durch Division und Multiplikation vereinfacht. Anhand der Eulerformel nach Gln.(1.2) werden Sinus- und Cosinusfunktion in den Bildbereich transformiert.

$$\cos \theta = \frac{1}{2} (e^{j\theta} + e^{-j\theta}) \quad (1.2a)$$

$$\sin \theta = \frac{1}{2j} (e^{j\theta} - e^{-j\theta}) = \frac{1}{2} (e^{j(\theta-\frac{\pi}{2})} + e^{-j(\theta-\frac{\pi}{2})}) \quad (1.2b)$$

Die Drehzeigerdarstellung in der komplexen Zahlenebene von Strom und Spannung ist übersichtlich und ermöglicht die Erarbeitung graphischer Lösungswege.



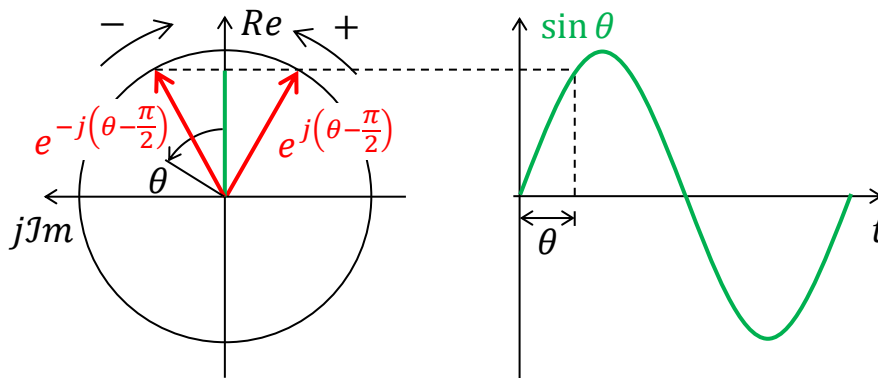


Abbildung 1: Darstellung der Cosinus- und Sinusfunktion in der komplexen Zahlenebene

Ströme und Spannungen müssen vor ihrer Transformation in den Bildbereich in eine Summe von Cosinus- und Sinusfunktionen zerlegt werden (siehe Kap.1.3).

1.1.2 Raumzeiger und Clark Transformation

Die Ströme und Spannungen des Drehstromsystems lassen sich in einem drei achsigen Koordinatensystem u,v,w abbilden. Jedoch ist dieses Koordinatensystem nicht sehr hilfreich für die Anwendung mathematischer Werkzeuge. Vorwiegender wird das Drehstromsystem in einem kartesischen Koordinatensystem dargestellt. Diese Koordinatentransformation wird anhand der Clarke-Transformation und dem Phasenoperator \underline{a} durchgeführt.

$$\underline{a} = e^{j\frac{2\pi}{3}} = -\frac{1}{2} + j\frac{\sqrt{3}}{2} \quad (1.3a)$$

$$\underline{a}^2 = e^{-j\frac{2\pi}{3}} = -\frac{1}{2} - j\frac{\sqrt{3}}{2} \quad (1.3b)$$

Mit der Clarke-Transformation werden nach Gln. (1.4) die drei skalaren Ströme oder Spannungen des Drehstromsystems in einen komplexen Raumzeiger $\underline{x}_{\alpha\beta}$ und eine skalare Nullkomponente x_γ transformiert.

$$\underline{x}_{\alpha\beta} = x_\alpha + jx_\beta = \frac{2}{3}(x_u + \underline{a} \cdot x_v + \underline{a}^2 \cdot x_w) \quad (1.4a)$$

$$\begin{bmatrix} x_\alpha \\ x_\beta \\ x_\gamma \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} x_u \\ x_v \\ x_w \end{bmatrix} \quad (1.4b)$$

Der Raumzeiger bildet die Relation zwischen den drei Phasen u,v,w ab, während die Nullkomponente das Verhältnis der drei Phasen u,v,w mit dem Neutralleiter abbildet.

Eine symmetrische dreiphasige Netzspannung nach den Gln. (1.5)

$$u_u(t) = 325V \cdot \sin(\omega t + \varphi) \quad (1.5a)$$

$$u_v(t) = 325V \cdot \sin\left(\omega t + \varphi - \frac{2\pi}{3}\right) \quad (1.5b)$$

$$u_w(t) = 325V \cdot \sin\left(\omega t + \varphi + \frac{2\pi}{3}\right) \quad (1.5c)$$

erzeugt einen Raumzeiger nach der Gl.(1.6), der im mathematisch positiven Sinn entgegen der Uhrzeigerrichtung rotiert.

$$\underline{u}_{\alpha\beta}(t) = 325V \cdot e^{j(\omega t + \varphi - \frac{\pi}{2})} \quad (1.6)$$

1.1.3 Nullsystem, Mitsystem und Gegensystem

Die symmetrischen Komponenten beschreiben das Drehstromsystem mit drei Systemen, das Nullsystem, das Mitsystem und das Gegensystem. Diese werden jeweils durch einen Zeiger dargestellt.

$$\begin{bmatrix} \underline{x}_{(0)} \\ \underline{x}_{(1)} \\ \underline{x}_{(2)} \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & \underline{a} & \underline{a}^2 \\ 1 & \underline{a}^2 & \underline{a} \end{bmatrix} \cdot \begin{bmatrix} \underline{x}_u \\ \underline{x}_v \\ \underline{x}_w \end{bmatrix} \quad (1.7)$$

Für die symmetrischen Komponenten wird die Transformation zwischen dem Originalbereich und dem Bildbereich nicht mit der Eulerformel nach Gl.(1.2) sondern durch die Beziehung

$$x(t) = \hat{x} \cdot \sin(\omega t + \varphi_x) \Leftrightarrow \underline{x}(t) = \hat{x} \cdot e^{j(\omega t + \varphi_x)} \quad (1.8)$$

definiert. Diese vereinfachte Darstellung einer skalaren Sinusfunktion durch nur einen rotierenden Zeiger führt zu einer mathematische Trennung zwischen dem Originalbereich und dem Bildbereich. Mit der Eulerformel nach Gl.(1.2) sowie dem Raumzeiger nach Gl.(1.4) ist ein direkter Übergang zwischen skalarer und komplexer Zahl gegeben und findet in dieser Arbeit Anwendung. Ein Vergleich der symmetrischen Komponenten nach Gl. (1.7) und der Clarke-Transformation nach Gln.(1.4) zeigt, dass der Raumzeiger $\underline{x}_{\alpha\beta}$ das Mitsystem und das Gegensystem zusammenfasst, und dass beide Nullkomponenten identisch sind.

$$\underline{x}_{\alpha\beta}(t) = -j \cdot \underline{x}_{(1)}(t) + j \cdot \underline{x}_{(2)}^*(t) \quad (1.9a)$$

$$x_\gamma(t) = \frac{1}{2} \left(-j \cdot \underline{x}_{(0)}(t) + j \cdot \underline{x}_{(0)}^*(t) \right) \quad (1.9b)$$

Das Mitsystem beschreibt den konventionellen 120° Phasenversatz der 3 Spannungen eines starren Netzes. Der Raumzeiger rotiert entgegen dem Uhrzeigersinn, seine Kreisfrequenz ist positiv.

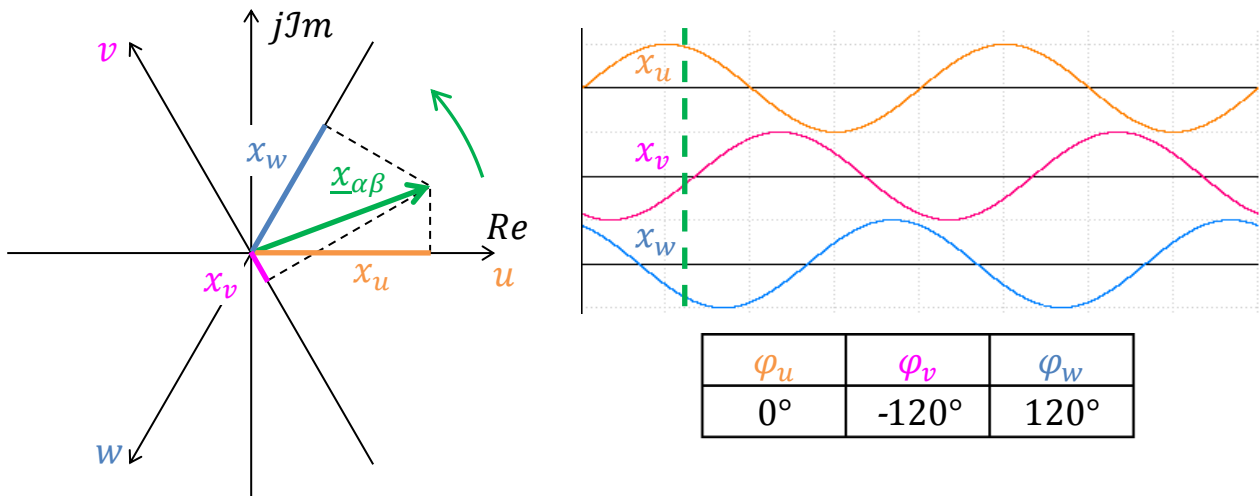


Abbildung 2: Mitsystem

Das Gegensystem wird durch den Austausch der Phasen v und w des Drehsystems gebildet. Der Raumzeiger rotiert im Uhrzeigersinn, seine Kreisfrequenz ist negativ.

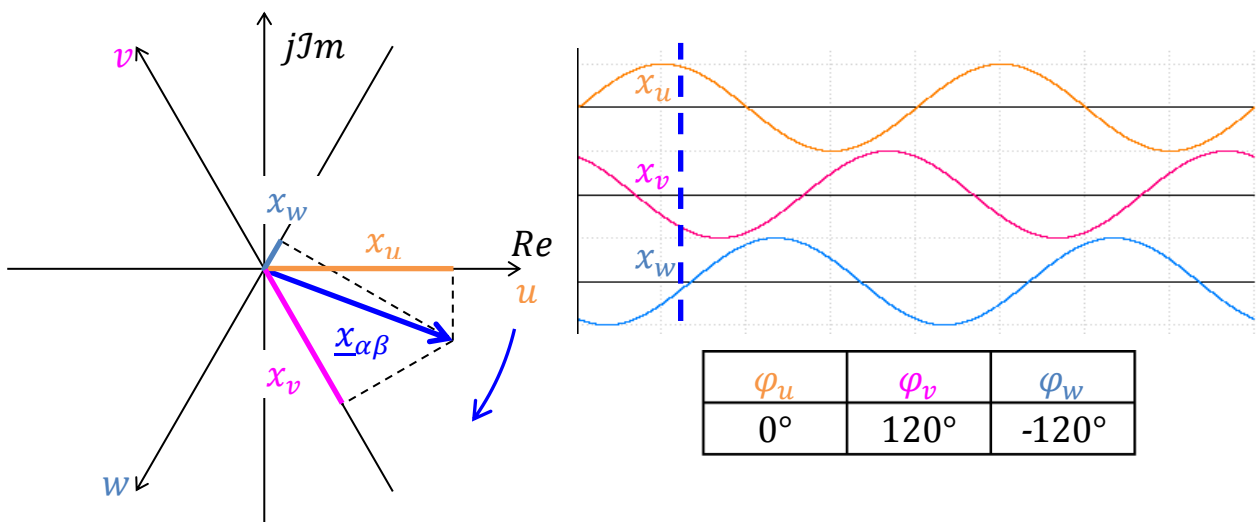


Abbildung 3: Gegensystem

Das Nullsystem beschreibt die Beziehung der drei Phasen u,v,w mit dem Neutralleiter. Alle 3 Achsen u,v,w des Nullsystems sind auf die reelle Achse ausgerichtet.

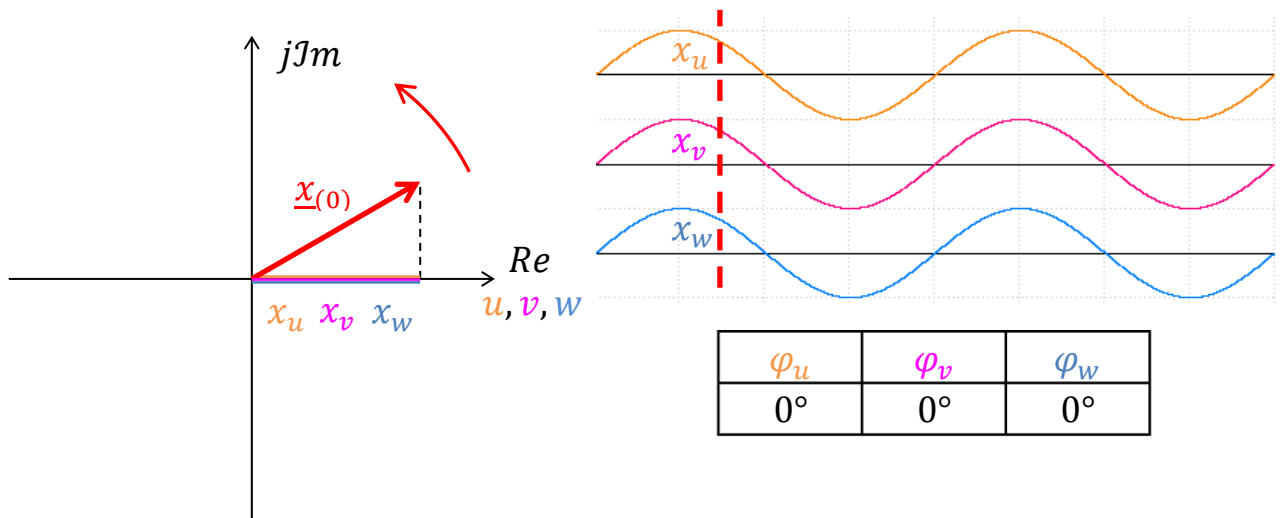


Abbildung 4: Nullsystem

Die Nullkomponente des Stromes aller drei Leiter u,v,w summiert sich in den Neutralleiter. Die Leiter-Leiter Spannungen enthalten keine Nullkomponente. Ein Aggregat kann einen Strom im Nullsystem verbrauchen oder erzeugen nur wenn er mit dem Neutralleiter verbunden ist.

1.1.4 Park Transformation

Die Anpassung von einem rotierenden Zeiger in einen ruhenden Zeiger bzw. von Wechselgröße in Gleichgröße ist in der Regelungstechnik notwendig. Ein Zeiger $\underline{x}_{\alpha\beta}$ rotiert in einem festen α, β Koordinatensystem. Anhand der Park-Transformation nach Gln. (1.10) wird er als ruhender Zeiger \underline{x}_{dq} in einem rotierenden d,q Koordinatensystem dargestellt.

$$\underline{x}_{dq} = x_d + jx_q = \underline{x}_{\alpha\beta} \cdot e^{j\theta} \quad (1.10a)$$

$$\begin{bmatrix} x_d \\ x_q \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} \quad (1.10b)$$

1.2 Mittelwertbildung

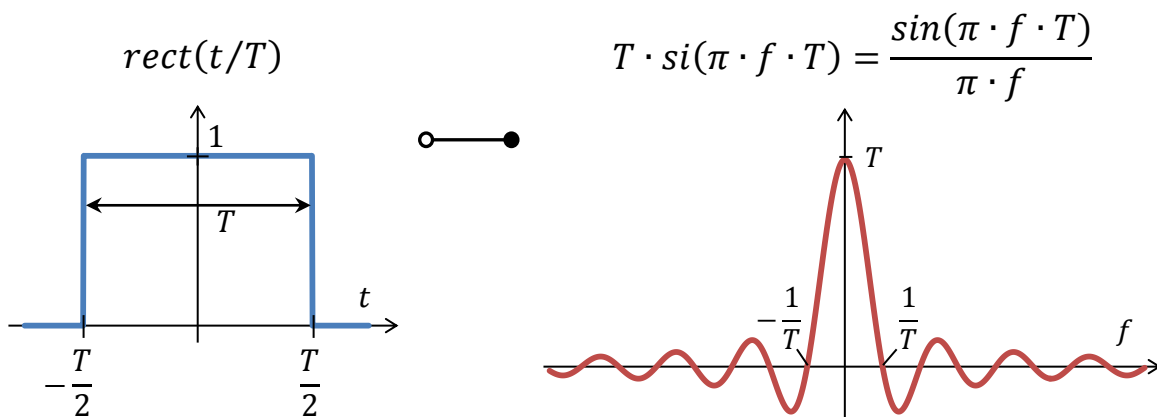
Die Berechnung eines Mittelwertes nach Gl. (1.11) findet einerseits in der Fourier-Analyse Anwendung und wird andererseits eingesetzt, um die hochfrequenten Störungen eines Messsignals zu dämpfen.

$$\bar{x}(t) = \frac{1}{T} \int_t^{t+T} x(\tau) \cdot d\tau \quad (1.11)$$

Die Systemfunktion des Mittelwertes als Regelungsblock ist anhand seiner Impulsantwort definiert. Die Impulsantwort des Mittelwertes

$$h(t) = \frac{1}{T} \int_t^{t+T} \delta(\tau) \cdot d\tau = \frac{1}{T} \cdot \text{rect}\left(\frac{t - \frac{T}{2}}{T}\right) \quad (1.12)$$

ist eine Rechteckfunktion von Breite T , Amplitude $1/T$ und Verzögerung $T/2$. Die Rechteckfunktion bildet im Frequenzbereich eine Si-Funktion ab.



Die Breite der Rechteckfunktion definiert die Polstellen der Si-Funktion. Die Impulsantwort des Mittelwertes $h(f)$ im Frequenzbereich ist eine Si-Funktion mit einer Verschiebung $T/2$.

$$h(f) = \text{si}(\pi \cdot f \cdot T) \cdot e^{-j\pi f T} \quad (1.13)$$

Der Amplitudengang des Mittelwertes ist gleich dem Amplitudengang einer Si-Funktion

$$|h(f)| = |\text{si}(\pi \cdot f \cdot T)| \quad (1.14)$$

und ist angenähert dem Amplitudengang eines Tiefpasses 1. Ordnung

$$h_{TP}(f) = \frac{1}{1 + \pi \cdot f \cdot T} \quad (1.15)$$

mit einer zusätzlichen starken Dämpfung an den Polstellen der Si-Funktion. Die beste Filterung ist dann erreicht, wenn die Fensterbreite des Mittelwertes auf die Frequenz der Störungen angepasst ist.

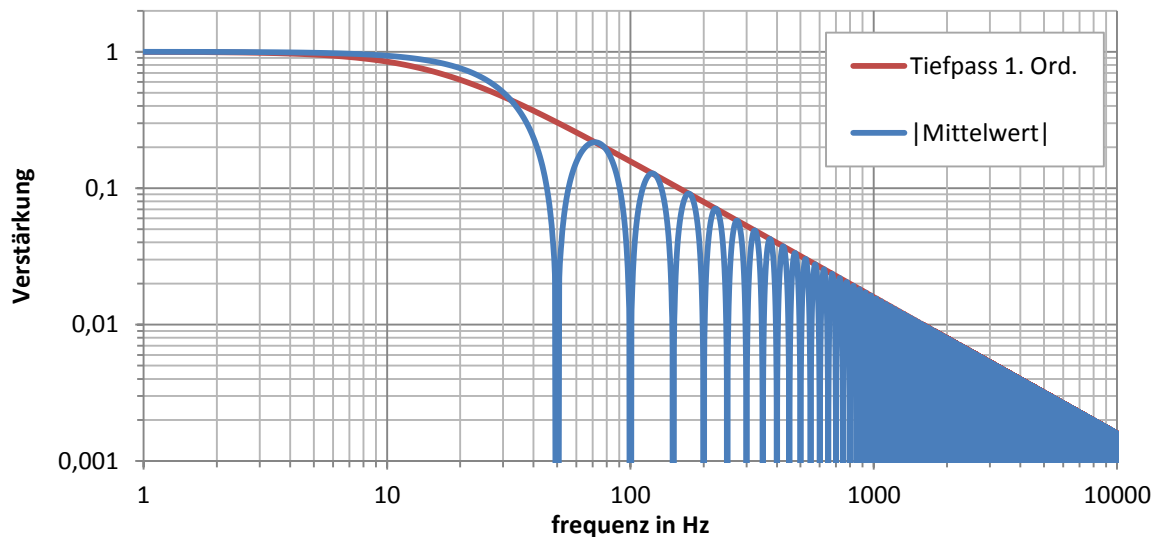


Abbildung 5: Amplitudengang des Mittelwertes und eines Tiefpasses 1. Ord. für $T = 20\text{ms}$

Die Si-Funktion ist eine reelle Funktion und besitzt nur Phasen von 0 und π . Der Phasengang des Mittelwertes ist eine Sägezahnfunktion von Periode T .

$$\begin{aligned} \arg\{h(f)\} &= \arg\{Si(\pi \cdot f \cdot T)\} - \pi \cdot f \cdot T \\ &= \begin{cases} -\pi \cdot f \cdot T & \text{wenn } Si(\pi \cdot f \cdot T) \geq 0 \\ \pi - \pi \cdot f \cdot T & \text{wenn } Si(\pi \cdot f \cdot T) < 0 \end{cases} \end{aligned} \quad (1.16)$$

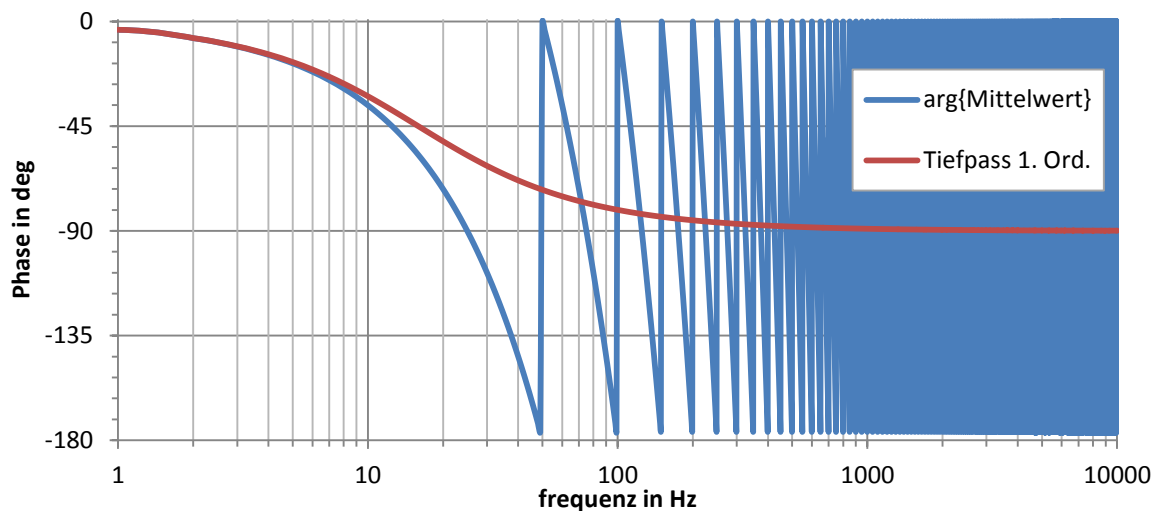


Abbildung 6: Phasengang des Mittelwertes und eines Tiefpasses 1. Ord. für $T = 20\text{ms}$

Der Mittelwert bewirkt eine Phasenverschiebung in dem gemittelten Messsignal. Für ihre Anwendung in der Oberschwingungskompensation wird eine Phasenverschiebungskorrektur durchgeführt.

1.3 Spektralanalyse

Die Regelung der Oberschwingungskompensation beruht auf Messungen und Reduzierungen gezielter Frequenzkomponenten der Netzspannung. Dazu wird die Spektralanalyse der Ströme und Spannungen durchgeführt.

Ein elektrisches Signal kann entweder im Zeitbereich durch seinen Zeitverlauf oder im Frequenzbereich anhand seines Spektrums dargestellt werden. Die Spektraldarstellung definiert das Signal als eine Summe von Schwingungen. Eine Schwingung entspricht einer Sinusfunktion. Eine Harmonische bzw. eine Oberschwingung ist eine Schwingung, deren Frequenz f_{OS} ein ganzzahliges Vielfaches der Grundschwingung ist, und wird mit Ordnung ν nummeriert. Die Grundschwingung in dieser Arbeit entspricht einer Schwingung mit der Netzfrequenz f_{Netz} .

$$f_{OS} = \nu \cdot f_{Netz} \quad (1.17)$$

1.3.1 Fourierreihenentwicklung

Die kontinuierliche Fourier Transformation legt die Korrespondenz zwischen Zeitbereich und Frequenzbereich fest und ist definiert durch:

$$\underline{x}(\omega) = \mathcal{F}\{x(t)\} = \int_{-\infty}^{+\infty} (x(t) \cdot e^{-j\omega t}) dt \quad (1.18)$$

Die Funktion $\underline{x}(\omega)$ ist das kontinuierliche Spektrum der Funktion $x(t)$. Die Fourier Transformation wird in digitale Prozessoren anhand der Fourierreihenentwicklung nach Gl. (1.19) implementiert.

$$\underline{x}_\nu = \mathcal{F}\{x(t)\} = \frac{2}{T} \cdot \int_t^{t+T} (x(t) \cdot e^{-j\nu\omega_1 t}) dt \quad (1.19)$$

Ordnung der Harmonischen: $\nu = 1, 2, 3 \dots$

Kreisfrequenz der Grundschwingung: ω_1

Das Ergebnis ist ein diskretes Spektrum mit Frequenzbandabstand ω_1 . Eine Harmonische ν kann als ruhender Zeiger oder als rotierender Zeiger oder als Sinusfunktion dargestellt werden.

$$\text{ruhende Zeiger: } \underline{x}_{\nu,dq} = \hat{x}_\nu \cdot e^{j\varphi_\nu} \quad (1.20a)$$

$$\text{Amplitude: } \hat{x}_\nu = |\underline{x}_\nu| \quad (1.20b)$$

$$\text{Phasenwinkel: } \varphi_\nu = \arg\{\underline{x}_\nu\} \quad (1.20c)$$

$$\text{rotierende Zeiger: } \underline{x}_{\nu,\alpha\beta} = \hat{x}_\nu \cdot e^{j(\nu\omega_1 t + \varphi_\nu)} \quad (1.20d)$$

$$\text{Sinusfunktion: } x_\nu(t) = \hat{x}_\nu \cdot \sin\left(\nu\omega_1 t + \varphi_\nu - \frac{\pi}{2}\right) = 2 \cdot \text{Im}\{\underline{x}_{\nu,\alpha\beta}\} \quad (1.20e)$$

Anhand zweier Beispiele wird die Auswertung der Grundschiwingung mit der Fourierreihenentwicklung für den idealen Fall und für den fehlerbehafteten Fall erläutert. Im ersten Beispiel enthält das Signal nach Gln. (1.21) die 2. Harmonische und die Grundschiwingung.

$$x(t) = \hat{x}_1 \cos(\omega_1 t) + \hat{x}_2 \cos(2\omega_1 t) \quad (1.21a)$$

$$x(t) = \frac{\hat{x}_1}{2} \cdot (e^{j\omega_1 t} + e^{-j\omega_1 t}) + \frac{\hat{x}_2}{2} \cdot (e^{j2\omega_1 t} + e^{-j2\omega_1 t}) \quad (1.21b)$$

Das Amplitudenspektrum des Signals ist

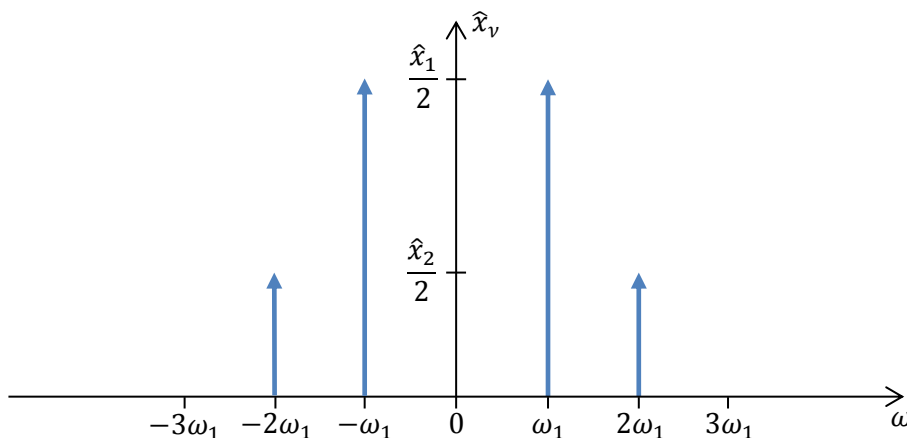


Abbildung 7: Fourierreihenentwicklung: 1. Beispielsignal

Die Auswertung der Grundschiwingung wird anhand der Fourierreihenentwicklung nach Gl. (1.19) in zwei Schritten beschrieben:

- 1) Multiplikation des Signals mit der komplexen Exponentialfunktion nach Gl.(1.22)

$$x(t) \cdot e^{-j\omega_1 t} = \frac{\hat{x}_1}{2} \cdot (1 + e^{-2j\omega_1 t}) + \frac{\hat{x}_2}{2} \cdot (e^{j\omega_1 t} + e^{-j3\omega_1 t}) \quad (1.22)$$

Die Multiplikation mit der komplexen Exponentialfunktion entspricht im Frequenzbereich einer Verschiebung des Spektrums um $-\omega_1$. Nach der Multiplikation ist die Grundschiwingung eine Gleichgröße. Das resultierende Amplitudenspektrum ist:

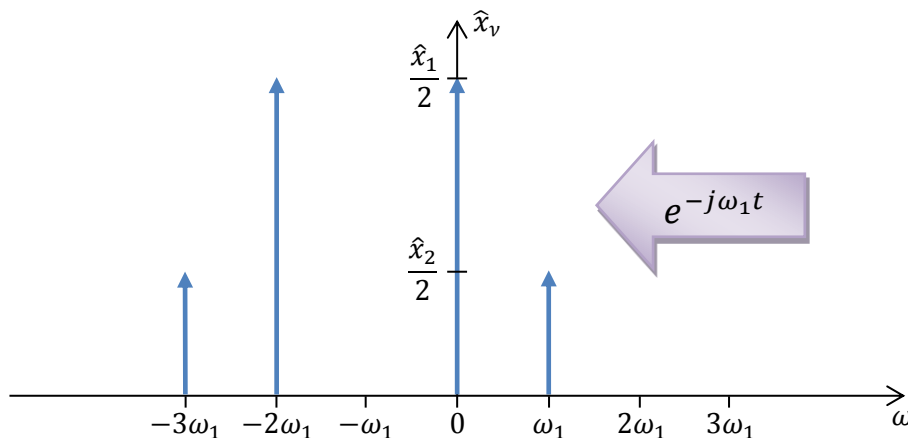


Abbildung 8: Fourierreihenentwicklung: Verschiebung des Spektrums

2) Berechnung des Gleichanteils anhand eines Mittelwertes.

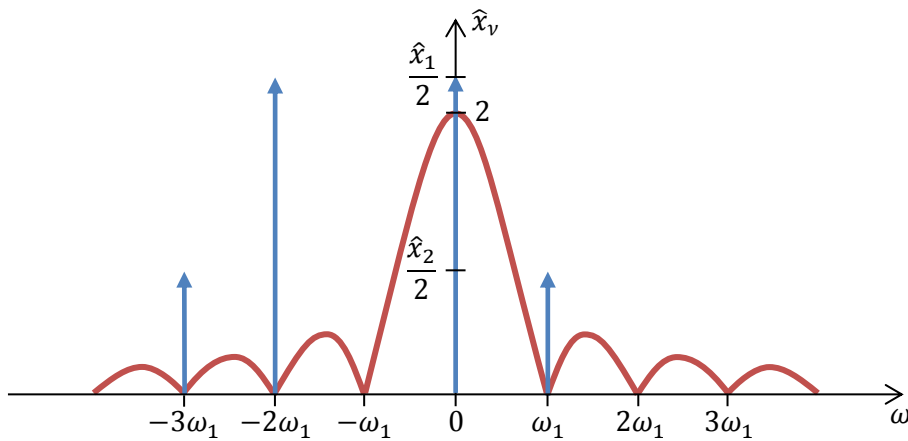


Abbildung 9: Fourierreihenentwicklung: Berechnung des Gleichanteils

In diesem Beispiel beträgt die Fensterbreite des Mittelwertes nach Gl. (1.23) eine Periode der Grundschwingung.

$$T = \frac{2\pi}{\omega_1} \quad (1.23)$$

Die Frequenzen $-3\omega_1, -2\omega_1, \omega_1$ sind dann eine Polstelle der Si-Funktion, der Mittelwert entfernt diese Frequenzanteile vollständig, nur der Gleichanteil bleibt. Das Ergebnis der Fourierreihenentwicklung ist:

$$\underline{x}_1 = \frac{\hat{x}_1}{2} \cdot 2 \cdot \text{si}(\pi \cdot 0 \cdot T) \cdot e^{-j\pi \cdot 0 \cdot T} = \hat{x}_1 \quad (1.24)$$

Dieses Beispiel stellt den Fall einer idealen Auswertung der Frequenz ω_1 eines periodischen Signals vor. In der Praxis ist die Netzfrequenz nicht konstant, die Ströme und die Spannungen enthalten Frequenzen, die nicht ein genaues Vielfaches der Fensterbreite des Mittelwertes sind. Im zweiten Beispiel enthält das Signal eine Grundschwingung mit einer Frequenzabweichung $\Delta\omega$ und eine Zwischenharmonische mit der Frequenz ω_{zh} .

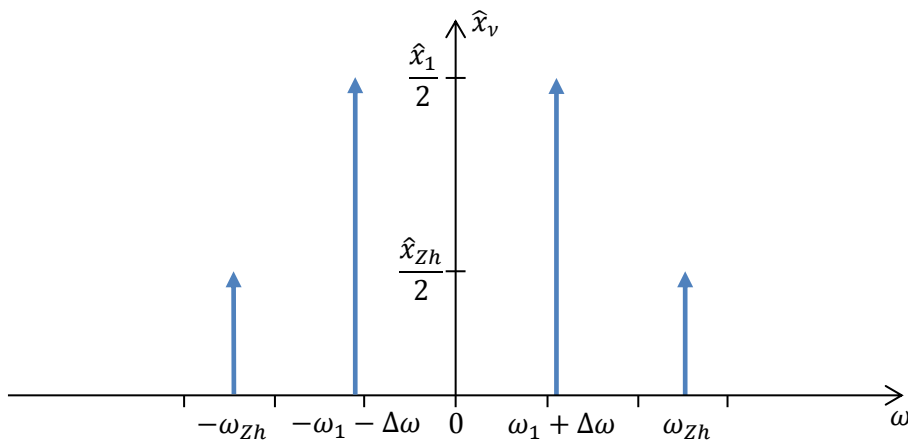


Abbildung 10: Fourierreihenentwicklung: 2. Beispielsignal

Nach der Verschiebung des Spektrums um die Frequenz ω_1 , befindet sich die Grundschwingung dann an der Stelle $\Delta\omega$. Die Filterung anhand des Mittelwertes wird die Grundschwingung nicht vollständig isolieren. Die ausgewertete Grundschwingung lautet:

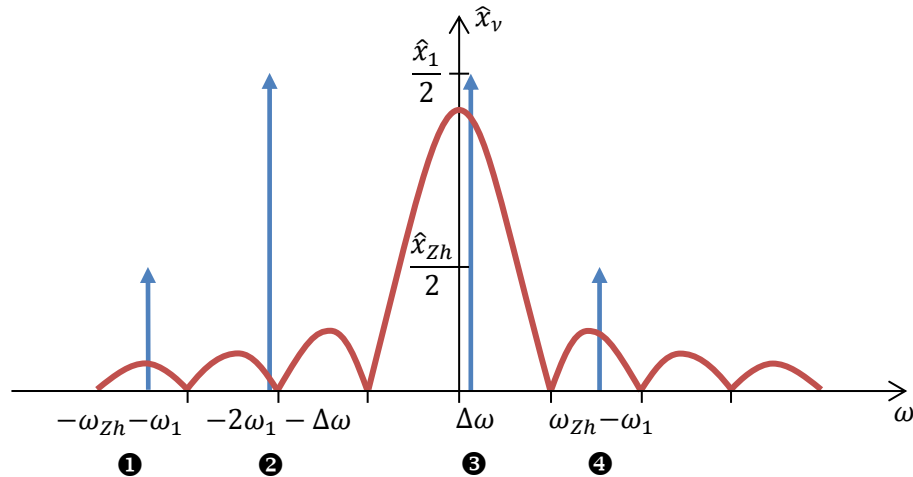


Abbildung 11: Fourierreihenentwicklung: Abbruchfehler

$$\underline{x}_1 = \textcircled{1} + \textcircled{2} + \textcircled{3} + \textcircled{4} \quad (1.25a)$$

$$\textcircled{1} \Leftrightarrow \hat{x}_{zh} \cdot \text{si} \left(\frac{(-\omega_{zh}-\omega_1) \cdot T}{2} \right) \cdot e^{-j \frac{(-\omega_{zh}-\omega_1) \cdot T}{2}} \quad (1.25b)$$

$$\textcircled{2} \Leftrightarrow \hat{x}_1 \cdot \text{si} \left(\frac{(-2\omega_1-\Delta\omega) \cdot T}{2} \right) \cdot e^{-j \frac{(-2\omega_1-\Delta\omega) \cdot T}{2}} \quad (1.25c)$$

$$\textcircled{3} \Leftrightarrow \hat{x}_1 \cdot \text{si} \left(\frac{(\Delta\omega) \cdot T}{2} \right) \cdot e^{-j \frac{(\Delta\omega) \cdot T}{2}} \quad (1.25d)$$

$$\textcircled{4} \Leftrightarrow \hat{x}_2 \cdot \text{si} \left(\frac{(\omega_{zh}-\omega_1) \cdot T}{2} \right) \cdot e^{-j \frac{(\omega_{zh}-\omega_1) \cdot T}{2}} \quad (1.25e)$$

Die Spektralanteile $\textcircled{1}, \textcircled{2}, \textcircled{4}$ sind nach der Mittelwertberechnung nicht vollständig gedämpft. Dieser Einfluss hat einen Abbruchfehler auf die Auswertung der Grundschwingung zur Folge. Für die Reduzierung der Abbruchfehler werden statt der rechteckigen Zeitfenster des Mittelwertes andere Funktionen angewendet, die das Signal gewichten. Somit sind die irrelevanten Harmonischen stärker unterdrückt als mit der Si-Funktion. In der Intelligenten Ladesäule wird der Algorithmus der gleitenden diskreten Fourier Transformation (DFT) angewendet. Die gleitende DFT ist mit einer Gewichtung des Zeitsignals nicht kompatibel. Eine andere Möglichkeit die Abbruchfehler zu reduzieren ist die Verlängerung der Zeitfenster des Mittelwertes für eine höhere Auflösung des diskreten Spektrums.

1.3.2 Gleitende Diskrete Fourier Transformation

Die Fourierreihenentwicklung eines digitalen Signals wird auf dem Prozessor der Intelligenzen Ladesäule anhand der diskreten Fourier Transformation implementiert. Die Messwerte der Ströme und Spannungen werden mit einer Taktfrequenz f_s digitalisiert und sind mit dem allgemeinen diskreten Signal x_n bezeichnet. Die DFT lautet

$$\underline{x}_v = \mathcal{F}\{x_n\} = \frac{2}{N} \cdot \sum_{n=0}^{N-1} \left(x_n \cdot e^{-j2\pi \frac{v}{N} n} \right) \quad (1.26a)$$

Ordnung der Oberschwingung:	v	
Taktfrequenz des Eingangssignals:	f_s	
Anzahl der Sample in dem Zeitfenster T :	$N = f_s \cdot T$	(1.26b)
Samplenummer in dem Zeitfenster T :	n	
diskrete Kreisfrequenz:	$\frac{2\pi \cdot v}{N}$	(1.26c)

und ist die Fourierreihenentwicklung eines zeitdiskreten Signals. Der Einfluss der Abtastfrequenz f_s auf die Genauigkeit der DFT ist minimal. Das Theorem von Shannon nach Gl. (1.27) muss für die Digitalisierung des Messsignals eingehalten werden, damit der höchste Frequenzanteil f_g des Messsignals mit mindestens 2 diskreten Werten pro Periode T_g digitalisiert ist.

$$f_s > 2 \cdot f_g \rightarrow T_g > 2 \cdot T_s \quad (1.27)$$

Die Abbildung 12 zeigt die Digitalisierung der 40. Oberschwingung (2kHz) mit zwei verschiedenen Abtastfrequenzen, einmal 200kHz und einmal 10kHz. Die niedrige Auflösung von 10kHz bildet das sinusförmige Signal mit 5 Werten pro Periode ab. Trotzdem weichen die Ergebnisse der DFT bei 200kHz und 10kHz Abtastfrequenz wenig voneinander ab (Tabelle 1). Die relative Abweichung zwischen 200kHz und 10kHz liegt bei 1%. Die Konstanz der Abtastfrequenz hat einen stärkeren Einfluss auf die Genauigkeit der DFT.

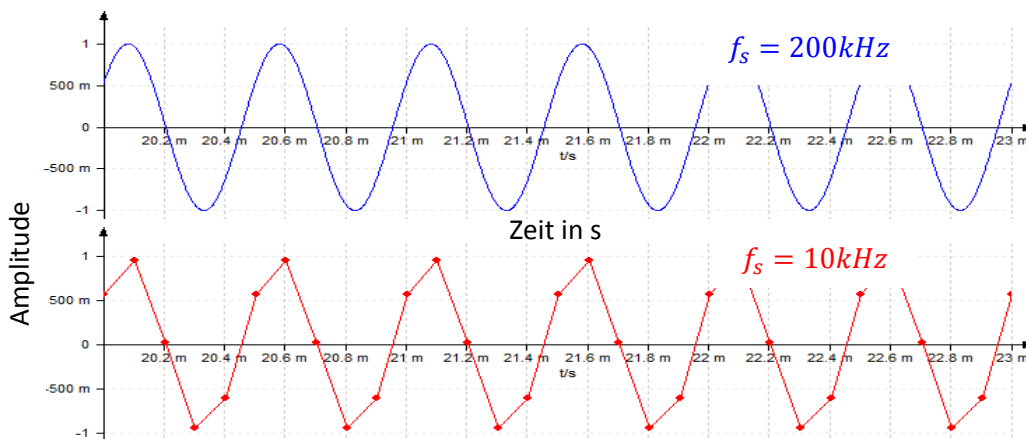


Abbildung 12: 2kHz Sinus mit verschiedenen Abtastfrequenzen

Abtastfrequenz	Amplitude	Phase	diskrete Werte pro Periode
Original	1V	35°	∞
200kHz	1,00018V	35,0146°	100
10kHz	1,01464V	35,2845°	5
5kHz	1,00021V	34,9964°	2,5

Tabelle 1: DFT-Auswertung eines 2kHz Sinus mit Verschiedenen Abtastfrequenzen

In der Intelligenten Ladesäule ist die Berechnung der DFT mit der Abtastfrequenz des diskreten Eingangssignals synchronisiert und erfolgt bei jedem neuen Eingangssample x_n . Die Summation aller N Werte der Zeitfenster T ist Rechenzeitaufwendig und wird durch den Einsatz der gleitenden DFT verhindert. Die Abbildung 13 zeigt, dass eine neue Summation durch die Subtraktion des ältesten Wertes (oranger Punkt) und die Addition des neuesten Wertes (grüner Punkt) erfolgen kann.

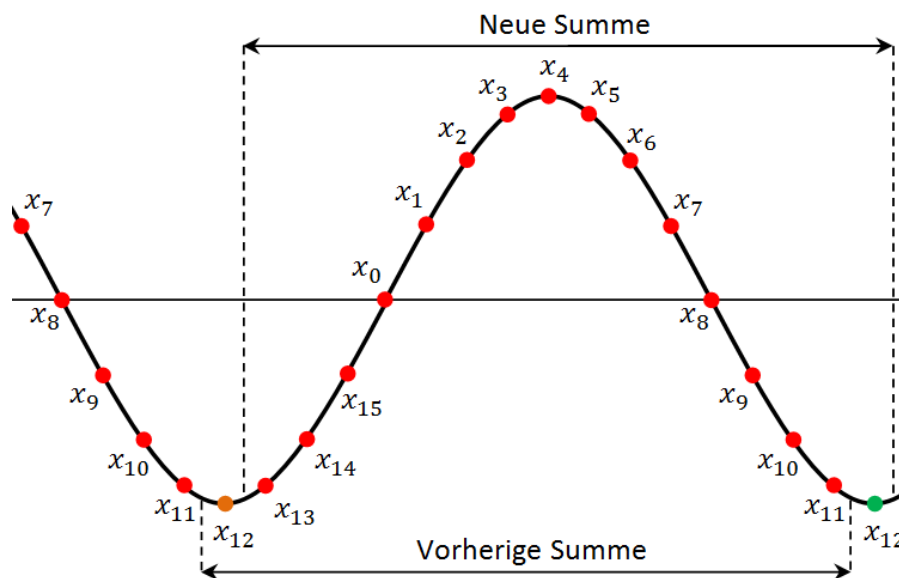


Abbildung 13: gleitende DFT

Die Gleitende DFT lautet:

$$\underline{x}_v = \underbrace{\underline{x}_{v,n-1}}_{\substack{\text{Summe der} \\ \text{letzten} \\ \text{Abtastperiode}}} + \underbrace{x_n \cdot \frac{2e^{-j2\pi\frac{v}{N}n}}{N}}_{\text{neuer Wert}} - \underbrace{x_{n-N} \cdot \frac{2e^{-j2\pi\frac{v}{N}n}}{N}}_{\text{alter Wert}} \quad (1.28)$$

1.4 Oberschwingungsklassifizierung nach der symmetrischen Komponente

In dem Drehstromnetz ist die Ordnung der Oberschwingungen ein Merkmal für Ihre Klassifizierung nach der symmetrischen Komponente. Die Beziehung der Phasenwinkel der Oberschwingungen in dem dreiphasigen System lautet:

$$\varphi_{v,v} = \nu \cdot (-120^\circ) + \varphi_{v,u} \quad (1.29a)$$

$$\varphi_{v,w} = \nu \cdot (120^\circ) + \varphi_{v,u} \quad (1.29b)$$

Daraus folgt die Klassifizierung der Oberschwingungen nach ihrer Ordnung:

$$\nu = 3 \cdot g = \{3, 6, 9, 12 \dots\} \quad \text{Nullsystem} \quad (1.30a)$$

$$\nu = 3 \cdot g + 1 = \{4, 7, 10, 13 \dots\} \quad \text{Mitsystem} \quad (1.30b)$$

$$\nu = 3 \cdot g - 1 = \{2, 5, 8, 11 \dots\} \quad \text{Gegensystem} \quad (1.30c)$$

$$g = \{1, 2, 3, 4 \dots \infty\}$$

Die Oberschwingungen des Nullsystems addieren sich im Neutralleiter und sind nur im 4 Leiter-System des Niederspannungsnetzes präsent.

1.5 Zusammenfassung

Die Darstellung von Strömen und Spannungen als komplexe Zeiger und die Darstellung des Drehstromsystems mit einem Raumzeiger wurde definiert. Die Beziehung zwischen einem Raumzeiger und den symmetrischen Komponenten wurde angedeutet. Die Beziehung zwischen einem rotierenden Zeiger und einem ruhenden Zeiger wurde mit der Park-Transformation erläutert. Die Filterung von hochfrequenten Störungen durch einen Mittelwert wurde erklärt. Die Messung der Oberschwingungen anhand der gleitenden diskreten Fourier Transformation und das Eintreten von Abbruchfehlern in dieser Messung, wurde ausführlich beschrieben. Abschliessend wurde die Aufteilung der Oberschwingungen nach den symmetrischen Komponenten angedeutet.

2 Normen

Oberschwingungen sind periodische Verzerrungen der sinusförmigen Netzspannungen und Netzströme. Sie entstehen durch Betriebsmittel mit nichtlinearer Kennlinie wie z.B. Transformatoren und Leuchtstofflampen sowie durch leistungselektronische Betriebsmittel wie Gleichrichter, Thyristoren, Triacs etc. Die ständig wachsende Anzahl dieser nichtlinearen Geräte, nicht zuletzt aus Gründen der Energieeffizienz, macht die Problematik der Netzoerschwingungen zu einem aktuellen Thema. Laut Schlabbach und Mombauer [SCMO08] betrug 2008 der Einsatz leistungselektronischer Lasten an der Haushaltlast bereits 33%.

2.1 Oberschwingungsgrenzwerte

2.1.1 Verträglichkeitspegel für Oberschwingungsspannungen in öffentlichen Niederspannungsnetzen

In den Normen und Richtlinien für das Niederspannungsnetz sind Grenzwerte für jede einzelne Harmonische sowie für den Gesamtoerschwingungsgehalt gegeben.

Verträglichkeitspegel für Oberschwingungsspannungen	
IEC 61000-2-4 VDE 0839-2-4	in Industrienetzen
IEC 61000-2-2 VDE 0839-2-2 EN 50160 VDEW-Richtlinie	im öffentlichen Niederspannungsnetz

Tabelle 2: Normen und Richtlinien der Oberschwingungsspannungen [TÜVS12]

95% der 10min-Mittelwerte des Effektivwertes der Oberschwingung dürfen in einem beliebigen Wochenintervall die in Tabelle 3 genannten Grenzwerte nicht überschreiten. Die Grenzwerte beziehen sich auf den Effektivwert der Grundschwingung.

Ungerade Harmonische				Gerade Harmonische	
Nichtvielfache von 3		Vielfache von 3			
Ordnung v	U_v/U_1 in %	Ordnung v	U_v/U_1 in %	Ordnung v	U_v/U_1 in %
5	6,0	3	5,0	2	2,0
7	5,0	9	1,5	4	1,0
11	3,5	15	0,4	6	0,5
13	3,0	21	0,3	8	0,5
$17 \leq v \leq 49$	$2,27 \frac{17}{v} - 0,27$	$27 \leq v \leq 45$	0,2	$10 \leq v \leq 50$	$0,25 \frac{10}{v} + 0,25$

Tabelle 3: Verträglichkeitspegel für Oberschwingungsspannungen in öffentlichen Niederspannungsnetzen nach DIN EN 61000-2-2

Der Gesamtoberschwingungsgehalt (Total Harmonic Distortion THD) darf einen Wert von 8% nicht überschreiten. Für die THD-Berechnung nach Gl. (2.31) müssen die Oberschwingungsspannungen bis zur 40. Ordnung betrachtet werden.

$$THD_U = \frac{\sqrt{\sum_{v=2}^{40} (U_v)^2}}{U_1} \quad (2.31)$$

Die Messung der Oberschwingungsspannungen müssen mit Messgerät nach DIN 61000-4-7 mit der Oberschwingungsuntergruppe durchgeführt werden, siehe hierzu Kapitel 2.2.2.

2.1.2 Grenzwerte für Oberschwingungsströme

Die Grenzwerte der Oberschwingungsströme werden anhand des Nennstromes des Gerätes in zwei Kategorien unterschieden.

EN 61000-3-2: Geräte mit einem Nennstrom $\leq 16 A$

Die Geräte werden wie folgt den einzelnen Klassen zugeordnet:

- Klasse A** symmetrische dreiphasige Geräte und alle Geräte ausgenommen Geräte der Klasse B,C,D
- Klasse B** tragbare Elektrowerkzeuge
- Klasse C** Beleuchtungseinrichtungen
- Klasse D** PC, PC Monitore, Fernseh- und Rundfunkempfänger mit Nennleistung $75 W < P \leq 600 W$

Die Grenzwerte der Klasse B betragen das 1,5-fache der Grenzwerte der Klasse A. Die Störaussendungen der Klasse D, dürfen die Werte der Klasse A nicht überschreiten. Grenzwerte sind nicht festgelegt für Geräte mit einer Nennleistung $P \leq 75W$, diese dürfen keine Beleuchtungseinrichtungen sein. Für die Klassen A, B und D sind in Tabelle 4 absolute Oberschwingungsgrenzwerte in Ampere für jede einzelne Oberschwingung bis zur 40. Ordnung angegeben. Ein Grenzwert für den Gesamtoberschwingungsgehalt (THD) ist nicht angegeben.

OS- Ordnung	Klasse				
	A	B	C	D	
v	I_v in A	I_v in A	I_v/I_1 in %	I_v in A	I_v in mA/W
ungeradzahlige Oberschwingungen					
3	2,3	3,45	$30 \cdot \lambda$	Identisch mit Klasse A	3,4
5	1,14	1,71	10		1,9
7	0,77	1,05	7		1
9	0,4	0,6	5		0,5
11	0,33	0,495	3		0,35
13	0,21	0,315	3		0,296
$15 \leq v \leq 39$	$0,15 \cdot 15/v$	$0,225 \cdot 15/v$	3		$3,85/v$
geradzahlige Oberschwingungen					
2	1,08	1,62	2	-	-
4	0,43	0,645	-	-	-
6	0,3	0,45	-	-	-
$8 \leq v \leq 40$	$0,23 \cdot 8/v$	$0,345 \cdot 8/v$	-	-	-

**Tabelle 4: Störaussendungsgrenzwerte nach DIN 61000-3-2 Gerät mit Eingangsstrom $\leq 16A$;
 λ : Leistungsfaktor**

EN 61000-3-12: Geräte mit einem Nennstrom von 16A bis 75 A

Die Grenzwerte sind in Abhängigkeit von der Kurzschlussleistung des Versorgungsnetzes am Verknüpfungspunkt des Gerätes definiert. Entscheidend hierfür ist das Kurzschlussverhältnis R_{sce} , dass sich nach Gl. (2.32) als Verhältnis der dreipoligen Kurzschlussleistung $S_{k,VP}$ am Verknüpfungspunkt VP zur Geräteleitung $S_{r,G}$ berechnet.

$$R_{sce} = \frac{S_{k,VP}}{S_{r,G}} \quad (2.32)$$

Die Grenzwerte unterscheiden sich zusätzlich nach der dreiphasigen Symmetrie des Eingangsstroms des Gerätes. In Tabelle 5 sind die Störaussendungsgrenzwerte für symmetrische Geräte angegeben.

Minimalwert R_{sce}	I_v/I_1 in %					THD
	5	7	11	13	geradzahlig	
33	10,7	7,2	3,1	2	16/v	13
66	14	9	5	3		16
120	19	12	7	4		22
250	31	20	12	7		37
≥350	40	25	15	10		48

Tabelle 5: Störaussendungsgrenzwert nach DIN 61000-3-12 für dreiphasige symmetrische Geräte mit Eingangsstrom < 75A

Die Berechnung des Gesamtüberschwingungsgehalts (THD) nach Gl.(2.33) berücksichtigt die Überschwingungsströme bis zur 40. Ordnung.

$$THD_I = \frac{\sqrt{\sum_{v=2}^{40} (I_v)^2}}{I_1} \quad (2.33)$$

Die Messung der Überschwingungsströme müssen mit Messgerät nach DIN 61000-4-7 mit dem Gruppierungsverfahren durchgeführt werden,

2.2 Überschwingungsmessverfahren DIN 61000-4-7

Die DIN EN 61000-4-7 (VDE 0847-4-7) beschreibt ein Messverfahren zur Auswertung von Überschwingungen in Stromversorgungsnetzen. Anhand der Fourieranalyse sind die zeitlichen Strom- und Spannungsabläufe in einzelne Frequenzanteile zerlegt. Der Algorithmus zur Durchführung der Fourier Transformation ist nicht vorgeschrieben. Die Anwendung des vorgeschriebenen rechteckigen Zeitfensters von 0,2s führt zu einer Frequenzauflösung von 5Hz. Dieses 5Hz Spektrum muss anhand Gruppierungsverfahren und Überschwingungsuntergruppen in ein 50Hz Spektrum umgerechnet werden.

2.2.1 Gruppierungsverfahren für die Auswertung der Überschwingungsströme

Das Gruppierungsverfahren berücksichtigt den Energiegehalt des gesamten Spektrums, sowie die zusätzlichen Spektralanteile, die bei zeitlichen Veränderungen der Überschwingungen auftreten. Nach dem Parseval'schen Theorem

$$\int_{-\infty}^{\infty} |x(t)|^2 dt = \int_{-\infty}^{\infty} |X(f)|^2 df \quad (2.34)$$

sind die Energie im Zeitsignal und die Energie im Spektrum gleich groß. Die Energie der Oberschwingung I_v im 50Hz Spektrum lässt sich durch die Summierung der Energie der benachbarten Spektrallinien I_k im 5Hz Spektrum ermitteln; man spricht von Gruppierung (Abbildung 14). Der Effektivwert beträgt:

$$I_v = \sqrt{\frac{I_{k-5}^2}{2} + \sum_{i=-4}^4 I_{k+i}^2 + \frac{I_{k+5}^2}{2}} \quad (2.35)$$

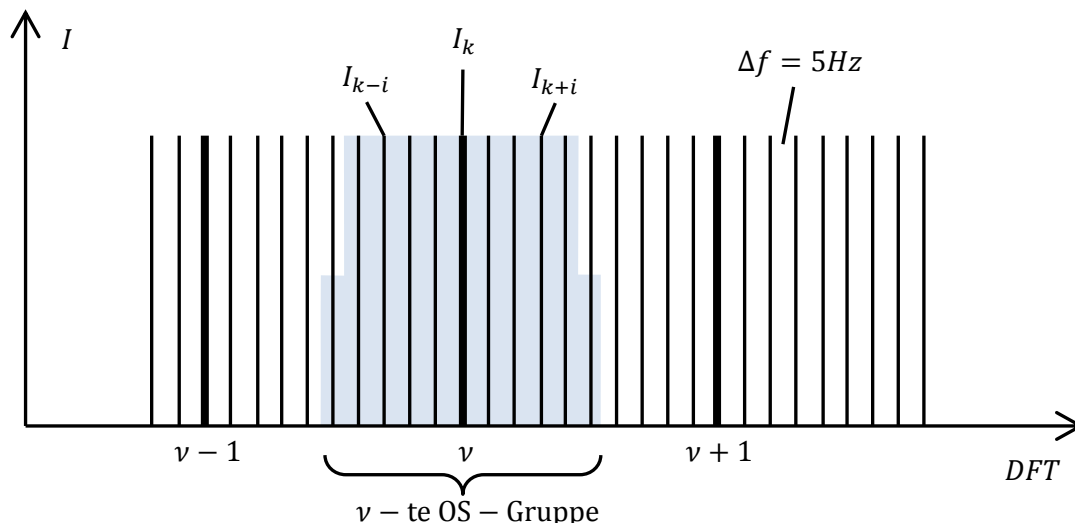


Abbildung 14: Gruppierungsverfahren DIN 61000-4-7 zur Auswertung der Oberschwingungsströme

Durch die Gruppierungsverfahren bilden die Si-Funktionen der 5Hz DFT eine neue Filterung des Spektrums (Abbildung 15).

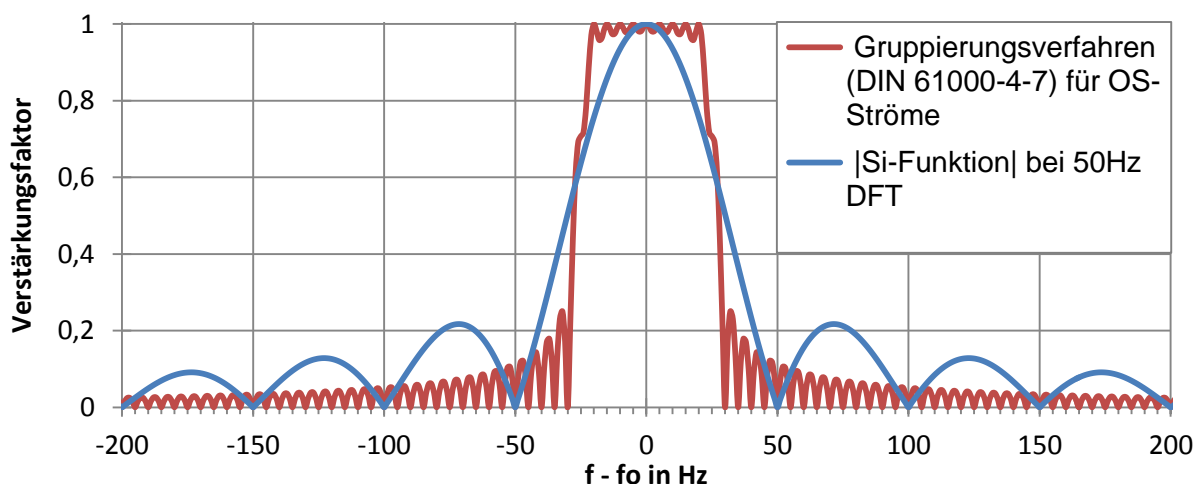


Abbildung 15: Spektrum der Gruppierungsverfahren (DIN 61000-4-7) und im Vergleich eine 50Hz Si-Funktion zur Auswertung der Frequenzkomponente f_0

Das Spektrum der Gruppierungsverfahren ähnelt einem rechteckigen Bandpassfilter. Der Spektralanteil $\pm 25\text{Hz}$ um f_0 ist vollständig ausgewertet, während die restlichen

Frequenzanteile stark gedämpft sind. In der Nähe der Polestellen der 50Hz Si-Funktion besitzt das Gruppierungsverfahren keine bessere Dämpfung als die 50Hz Si-Funktion. Die Reduzierung der Abbruchfehler ist nur im Fall von Zwischenharmonischen verbessert. Harmonische mit leichten Frequenzabweichungen (Beispiel 3. OS mit 150,5Hz statt 150Hz) verursachen jedoch im selben Maß Abbruchfehler wie bei der 50Hz DFT. Um die Genauigkeitsanforderung an Oberschwingungsmessgerät zu halten, ist es nötig die Zeitfenster der DFT an der Netzfrequenz anzupassen.

2.2.2 Oberschwingungsuntergruppe für die Auswertung der Oberschwingungsspannung

Die Oberschwingungsspannungen ändern sich nicht so stark wie die Oberschwingungsströme, deshalb werden die Harmonischen der Netzspannung mit Oberschwingungsuntergruppen ausgewertet. Zur Oberschwingungsuntergruppe nach Gl. (2.36) werden nur die beiden benachbarten Spektrallinien zur Oberschwingung gruppiert (Abbildung 16).

$$U_v = \sqrt{\sum_{i=-1}^1 U_{k+i}^2} \quad (2.36)$$

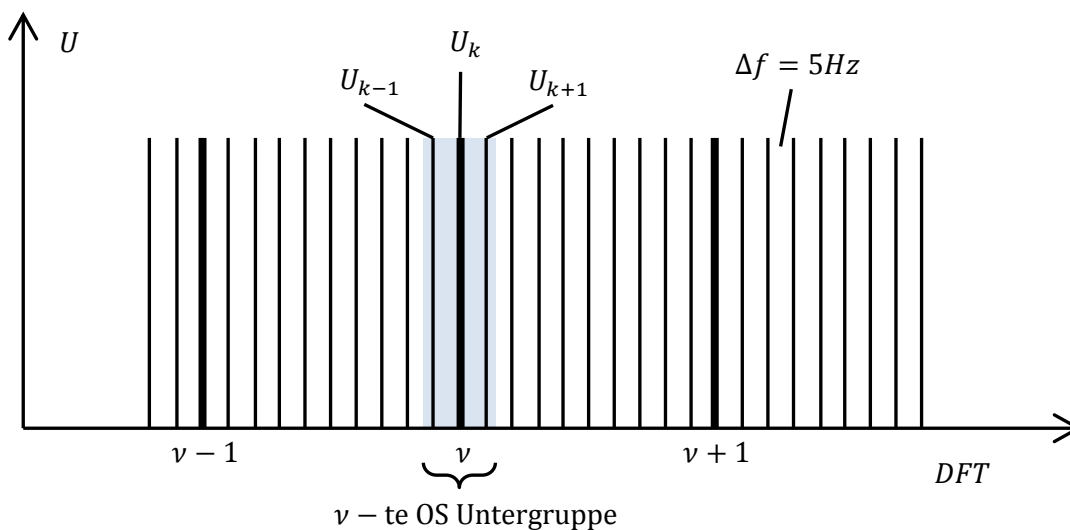


Abbildung 16: Oberschwingungsuntergruppe DIN EN 61000-4-7 zur Auswertung der Harmonischen der Netzspannung

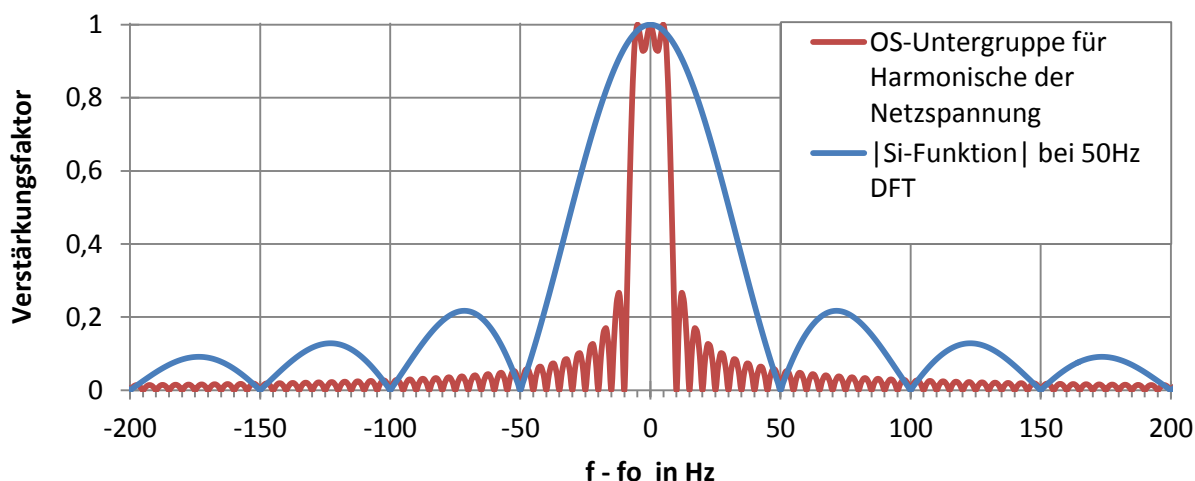


Abbildung 17: Spektrum der Oberschwingungsuntergruppe (DIN 61000-4-7) und im Vergleich eine 50Hz Si-Funktion zur Auswertung der Frequenzkomponente f_0

Die Oberschwingungsuntergruppe bildet einen schmalen Bandpassfilter mit 15Hz Bandbreite. Die Amplitude der Grundschwingung der Netzspannung entspricht einem 100-fachen der Amplitude der restlichen Oberschwingungen. Die starke Dämpfung der Grundschwingung ist äußerst wichtig für eine wertvolle Auswertung ohne Abbruchfehler der anderen Oberschwingungen der Netzspannung. Die Grundschwingung ist stark gedämpft, wenn das Zeitfenster der DFT mit der Netzfrequenz angepasst wird oder wenn eine Vorfilterung der Grundschwingung der Netzspannung erfolgt.

2.3 Zusammenfassung

Verbraucher mit nichtlinearen Kennlinien erzeugen Oberschwingungsströme, die sich über die Netzimpedanz in Oberschwingungsspannungen verbreiten. Die Grenzwerte für die Oberschwingungsstromaussendung eines Gerätes aus der DIN 61000-3-2 und DIN 61000-3-12 sowie die Grenzwerte für die Oberschwingungsspannungen des Niederspannungsnetz aus der DIN 61000-2-2 wurden angegeben. Das Oberschwingungsmessverfahren nach DIN 61000-4-7 wurde erläutert. Die Oberschwingungsmessung des Stromes mit einem Gruppierungsverfahren dazu die Oberschwingungsmessung der Spannung mit einer Oberschwingungsuntergruppe wird vorgeschrieben.

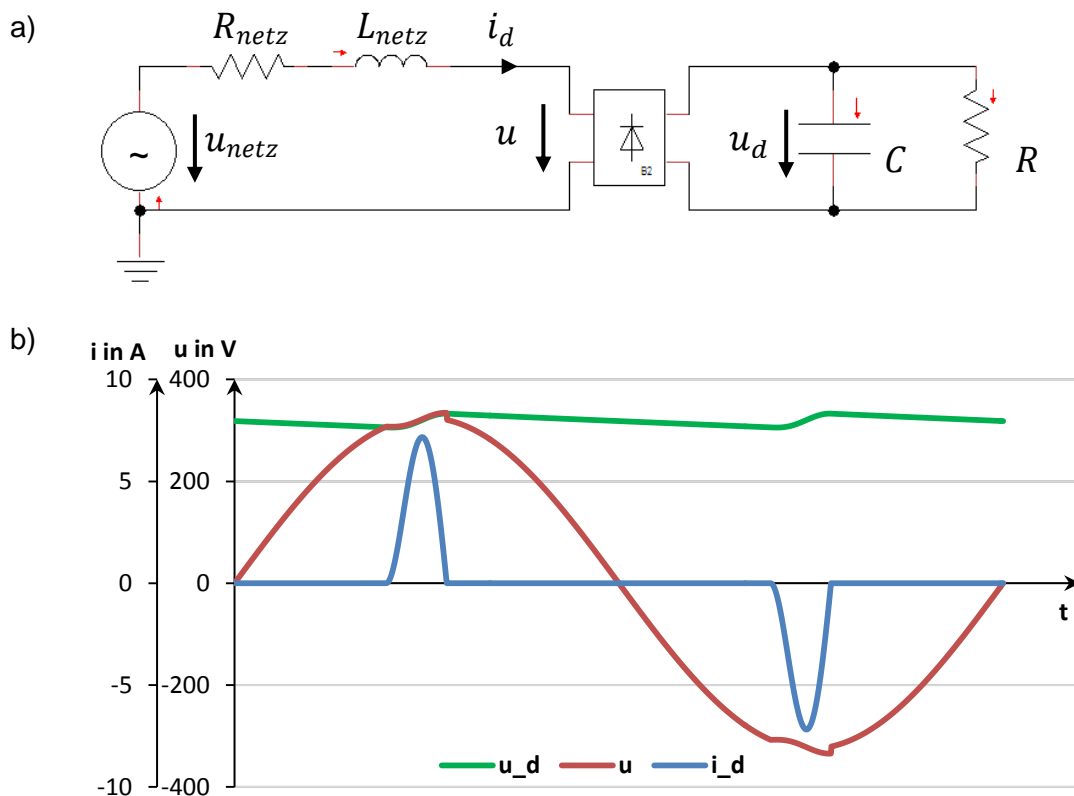
Dieses Oberschwingungsmessverfahren nach DIN 61000-4-7 sollte in der Intelligenten Ladesäule angewendet werden. Wegen der Rechenzeitbelastung des Prozessors wurde dieses Oberschwingungsmessverfahren in der Intelligenten Ladesäule nur separat erprobt, jedoch nicht mit der Regelung der Oberschwingungskompensation kombiniert.

3 Theorie der Oberschwingungskompensation

In dem ersten Abschnitt wird die Ursache für die Oberschwingungen des Netzes beschrieben. Die Verzerrung der Netzspannung durch einen nichtlinearen Verbraucher wird anhand einer Simulation analysiert. Aus dieser Erkenntnis wird die mögliche Kompensation einer Oberschwingungsspannung hergeleitet und die Anforderungen für die Oberschwingungskompensation mit der Intelligenten Ladesäule festgelegt. Die Netzimpedanz ist eine wichtige Komponente des Kompensationsprozesses, aus diesem Grund wird sie für das Niederspannungsnetz beschrieben. Zuletzt wird die positive Wirkung eines Kompensationsstroms für die Reduzierung der Oberschwingungen der Netzspannung durch eine Simulation bewiesen.

3.1 Entstehung der Oberschwingungsspannungen

Bevor die Kompensation der Oberschwingungsspannung mit der Intelligenten Ladesäule erläutert wird, ist es notwendig die Entstehung der Oberschwingungen der Netzspannung anhand eines Beispiels zu beschreiben. Eine aktuell typische nichtlineare Last ist der Zweiweg-Gleichrichter mit kapazitiver Glättung nach Abbildung 18 der in Fernsehgeräten, PCs und Kompaktleuchtstofflampen insbesondere im Haushalts- und Bürobereich verbreitet ist.



Wird die Netzspannung größer als die Spannung der Gleichspannungsseite – die Durchlassspannung der Dioden wird vernachlässigt –, so kann ein Strom fließen, der den Kondensator auflädt. Beim Unterschreiten der Spannungswerte des nachgeladenen Kondensators, sperren die Dioden, der Ladestrom wird unterbrochen. Die Spannung der Gleichspannungsseite fällt in Abhängigkeit der Zeitkonstanten der angeschlossenen Last ab. Diese besteht aus Glättungskondensator C und Last R. Dieser Vorgang wiederholt sich bei jeder Halbschwingung der Netzspannung. Anhand des Amplitudenspektrums des Ladestromimpulses (Abbildung 19) wird ersichtlich dass die Ordnung der Oberschwingungen aus der Pulszahl des Gleichrichters nach Gl. (3.37) herzuleiten ist,

$$v = g \cdot p \pm 1 \quad (3.37)$$

p: Pulszahl des Gleichrichters

g = 1,2,3,4, ... ∞

und dass die Frequenz der Oberschwingungen einem genauen Vielfachen einer 50,1Hz Netzfrequenz entspricht.

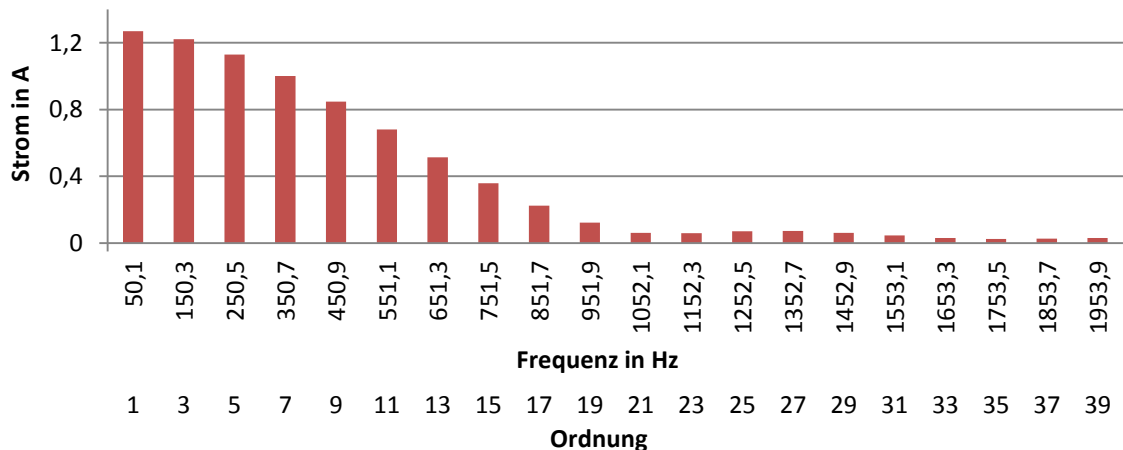


Abbildung 19: Oberschwingungsstrom eines Zweiweg-Gleichrichters

Anhand der Flussrichtung der Oberschwingungen, wird festgelegt ob ein Gerät Oberschwingungen erzeugt oder verbraucht. Die Flussrichtung von Oberschwingungen ist durch die Phasenverschiebung zwischen Strom und Spannung für jede Oberschwingung definiert (Tabelle 6) und muss nicht mit der Richtung des Leistungsflusses übereinstimmen. Strom und Spannung werden in diesem Fall nach dem Verbraucherzählpeilsystem gemessen (Abbildung 20).

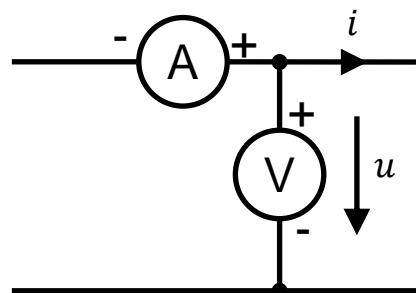


Abbildung 20: Verbraucherzählpeilsystem

Phasenwinkel $\varphi = \varphi_u - \varphi_i$ der Oberschwingungen	Flussrichtung
$0^\circ \leq \varphi \leq 90^\circ$	zur Last (Aufnahme)
$90^\circ \leq \varphi \leq 180^\circ$	in das Netz (Abgabe)
$180^\circ \leq \varphi \leq 270^\circ$	in das Netz (Abgabe)
$270^\circ \leq \varphi \leq 360^\circ$	zur Last (Aufnahme)

Tabelle 6: Flussrichtung von Oberschwingungen

In dem Beispiel des Zweiweg-Gleichrichters ist die Netzspannungsquelle u_{netz} überschwingungsfrei. Der Phasenwinkel der Oberschwingungen zeigt, dass der Zweiweg-Gleichrichter Oberschwingungsströme ins Netz einspeist.

Ordnung ν	Phasenwinkel $\varphi = \varphi_u - \varphi_i$
3	242°
5	252°
7	257°
9	259°
11	261°
13	262°
15	263°
17	263°

Tabelle 7: Phasenwinkel der Oberschwingungen eines Zweiweggleichrichters

In ein überschwingungsfreies Netz speist ein nichtlinearer Verbraucher Oberschwingungsströme ins Netz. Es entstehen Oberschwingungsspannungen über der Netzimpedanz, welche zu einer Verzerrung der Netzspannung führt. Die stärkste Verzerrung der Netzspannung findet am Netzanschlusspunkt des nichtlinearen Verbrauchers statt.

Dem gegebenüber gestellt nimmt in einem Oberschwingungsbelasteten Netz ein Oberschwingungsverbraucher den Oberschwingungsstrom auf, welches zu einer Reduzierung der Oberschwingungsspannung führt.

Auf dieser Kenntnisnahme beruht die Oberschwingungskompensation der Netzspannung und wird im folgenden Kapitel erläutert.

3.2 Reduzierung einer Spannungsüberschwingung durch Einprägung eines Kompensationsstromes

Die Intelligente Ladesäule kann eine Spannungsüberschwingung auf die drei Leiter des Drehstromnetzes kompensieren. Für ein besseres Verständnis wird die Kompensation in einem einphasigen Netz (Abbildung 21) erläutert. Das elektrische Netz ist mit einer Spannungsquelle und einer konstanten ohmsch-induktiven Netzimpedanz dargestellt. Dabei ist die Netzspannung verzerrt, sie enthält eine Spannungsüberschwingung \underline{u}_{OS} mit konstanter Frequenz und Amplitude.

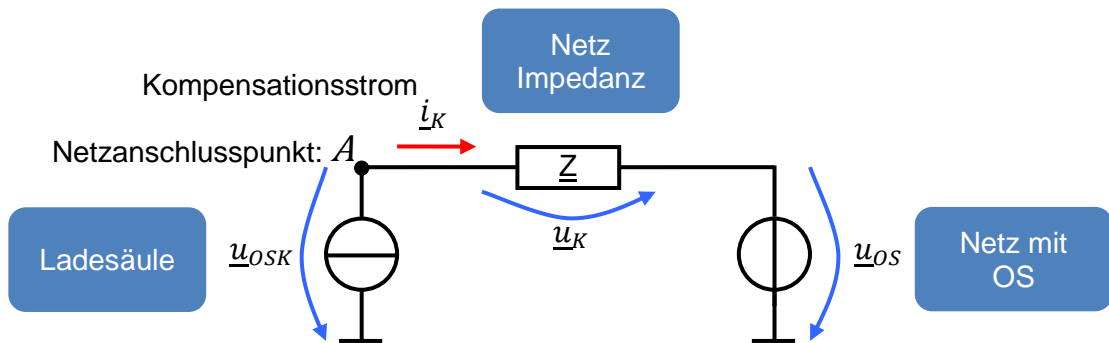


Abbildung 21: Einphasige Kompensation der Spannungsüberschwingung

Die Ladesäule prägt einen sinusförmigen Kompensationsstrom i_K ein. Dieser Strom verursacht über die Netzimpedanz einen Spannungsabfall \underline{u}_K , der Kompensationsspannung heißt. Die Summation der Spannungsüberschwingung \underline{u}_{OS} des Netzes mit der phasenregierten Kompensationsspannung \underline{u}_K führt zu einer Reduzierung der Amplitude dieser Spannungsüberschwingung an dem Netzanschlusspunkt A.

Die Messung der Überschwingung mit der gleitenden DFT ist hier als ideal angenommen. Der Strom und die Spannungen nach Gl. (3.38) werden mit ruhenden Zeigern in einem rotierenden d,q Koordinatensystem dargestellt.

$$\text{Kompensationsprozess:} \quad \underline{u}_{OSK} = \underline{u}_K + \underline{u}_{OS} \quad (3.38a)$$

$$\text{Kompensierte OS-Spannung:} \quad \underline{u}_{OSK} = \hat{u}_{OSK} \cdot e^{j\varphi_{OSK}} \quad (3.38b)$$

$$\text{Kompensationsspannung:} \quad \underline{u}_K = \hat{u}_K \cdot e^{j\varphi_{uK}} \quad (3.38c)$$

$$\text{OS-Spannung ohne Kompensation:} \quad \underline{u}_{OS} = \hat{u}_{OS} \cdot e^{j\varphi_{OS}} \quad (3.38d)$$

$$\text{Kompensationsstrom:} \quad \underline{i}_K = \hat{i}_K \cdot e^{j\varphi_{iK}} \quad (3.38e)$$

$$\text{Netzimpedanz:} \quad \underline{Z}_N = Z \cdot e^{j\varphi_{zN}} \quad (3.38f)$$

Die Effektivität des Kompensationsprozesses nach Gl (3.38a) hängt von dem Phasenwinkel zwischen der Spannungsüberschwingung \underline{u}_{OS} und der Kompensationsspannung \underline{u}_K ab (Abbildung 22).

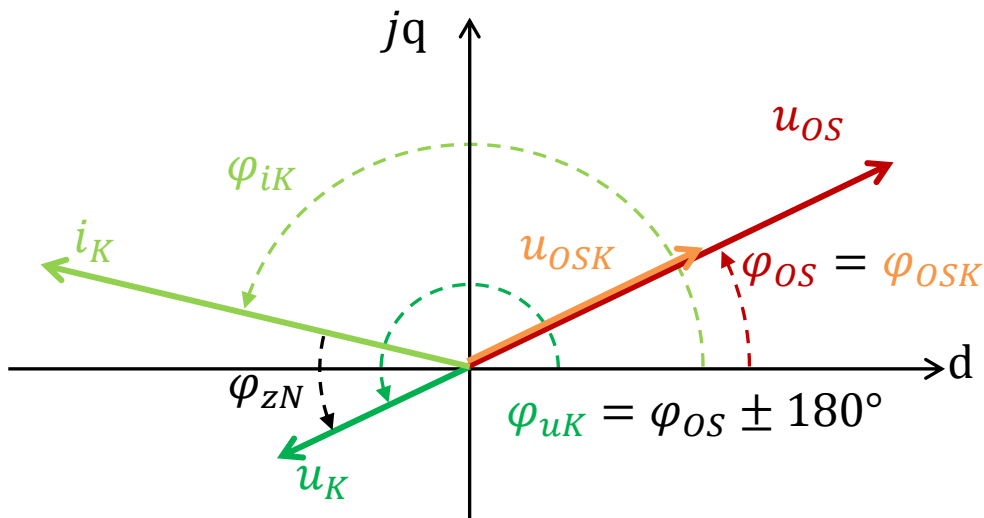


Abbildung 22: Zeigerdiagramm einer optimalen Kompensation

Eine optimale Kompensation wird nach Gl. (3.39) bei einer Phasenverschiebung von 180° zwischen \underline{u}_{OS} und \underline{u}_K erreicht.

$$\varphi_{uK} = \varphi_{OS} \pm 180^\circ \quad (3.39)$$

Der Phasenwinkel der Kompensationsspannung wird anhand des Kompensationsstromes eingestellt. Die Netzimpedanz stellt nach Gl. (3.40) die Beziehung zwischen dem Kompensationsstrom und der Kompensationsspannung dar:

$$\varphi_{uK} = \varphi_{iK} + \varphi_{zN} \quad (3.40)$$

Die optimale Einstellung der Phasenwinkel des Kompensationsstromes ist wie folgt definiert:

$$\varphi_{iK} = \varphi_{OS} - \varphi_{zN} \pm 180^\circ \quad (3.41)$$

In der Gl. (3.41) sind die Phasenwinkel des Kompensationsstromes und der Spannungsoberschwingung bekannt, sie werden am Netzanschlusspunkt der Intelligenten Ladesäule gemessen. Der Phasenwinkel der Netzimpedanz am Netzanschlusspunkt ist nicht bekannt. Das Ziel für die Regelung der Oberschwingungskompensation, ist die optimale Einstellung der Phasenwinkel des Kompensationsstromes.

Ausgehend vom Kompensationsprozess der Gl. (3.38a) wird die Effektivität der Kompensation in Abhängigkeit des Phasenwinkels der Kompensationsspannung in Abbildung 23 dargestellt. Die Amplitude der Spannungsoberschwingung \underline{u}_{OS} des Netzes beträgt 10V und deren Phasenwinkel 0° . Die Amplitude der kompensierten Oberschwingung ist am niedrigsten wenn die Kompensationsspannung phasennegiert ist bei 180° . Die farbigen Kurven stellen verschiedene Amplituden der Kompensationsspannung dar. Bei 10V Kompensationsspannung (rot) wird bei 180° die Oberschwingung vollständig kompensiert und im Bereich 120° bis 270° reduziert. Bei niedrigerer Kompensationsspannung ist die Oberschwingung nur teilweise kompensiert und der Kompensationsbereich verbreitert sich (orange).

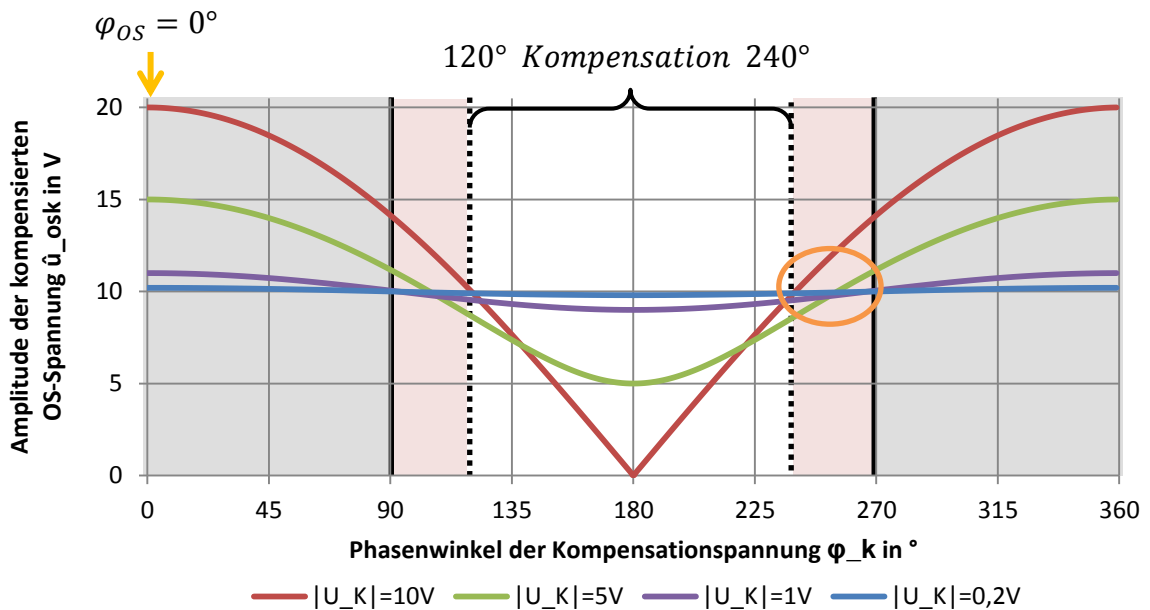


Abbildung 23: Effektivität der Kompensation

Um den optimalen Kompensationspunkt zu finden, wird an der Intelligenten Ladesäule ein Kompensationsstrom mit konstanter Amplitude eingestellt, dessen Phasenwinkel sich variieren lässt. Der optimale Kompensationspunkt wird mit der Amplitude der kompensierten Spannungsüberschwingung \hat{u}_{OSK} bestimmt. Diese Bestimmung der Kompensationswirkung ist bei konstanten Netzparametern erfolgreich. Deren Implementierung bei einer veränderlichen Netzimpedanz und einer veränderlichen Spannungsüberschwingung \underline{u}_{OS} ist eine zukünftige Herausforderung. Diesbezüglich können Angaben aus der Literatur entnommen werden, welche Informationen über den Wertebereich der Netzimpedanz liefern.

3.3 Netzimpedanz im Niederspannungsnetz

Die Netzimpedanz ist abhängig von Leistung, Frequenz und Zeit. Tageszeitabhängig durch die Ab- und Zuschaltung von Lasten sowie frequenzabhängig durch den Frequenzgang der Lasten (Resonanz, nichtlineare Lasten). Durch die ungleichmäßige Aufteilung einphasiger Verbraucher kann die Netzimpedanz auch unsymmetrisch sein. Dabei ist im Niederspannungsnetz der kapazitive Anteil der Netzimpedanz im Vergleich zum ohmsch-induktiven Anteil vernachlässigbar [SCHU04]. Die Netzimpedanz im Niederspannungsnetz ist dann wie folgt definiert:

$$\underline{Z}_N = R_N + j\omega L_N = Z_N \cdot e^{j\varphi_{zN}} \quad (3.42)$$

Der Phasenwinkel der Netzimpedanz nach Gl. (3.43) liegt in einem Bereich von 0° bis 90° .

$$\varphi_{zN} = \arg\{R_N + j\omega L_N\} = \arctan\left(\frac{\omega L_N}{R_N}\right) \quad (3.43)$$

Bei Oberschwingungen niedriger Frequenz, ist die Netzimpedanz überwiegend ohmsch, der Phasenwinkel liegt in der Nähe von 0° . Bei Oberschwingungen höherer Frequenz,

nimmt der reaktive Anteil ωL_N zu und der Phasenwinkel steigt in Richtung 90° .

In der DIN 61000-3-3 sind die Bezugsimpedanzen nach Gln. (3.44) angegeben, die bei 50Hz nicht überschritten werden sollten.

$$\text{Außenleiter:} \quad \underline{Z}_{N_L} = 0,24 \, \Omega + j\omega \cdot 0,478 \, \text{mH} \quad (3.44a)$$

$$\text{Nullleiter} \quad \underline{Z}_{N_{Null}} = 0,16 \, \Omega + j\omega \cdot 0,318 \, \text{mH} \quad (3.44b)$$

$$\text{Außenleiter-Nullleiter:} \quad \underline{Z}_{N_{L,Null}} = 0,4 \, \Omega + j\omega \cdot 0,8 \, \text{mH} \quad (3.44c)$$

In der Praxis liegt diese Bezugsimpedanz nicht immer vor, trotzdem liefert sie eine Abschätzung des Phasenwinkels der Netzimpedanz (Abbildung 24).

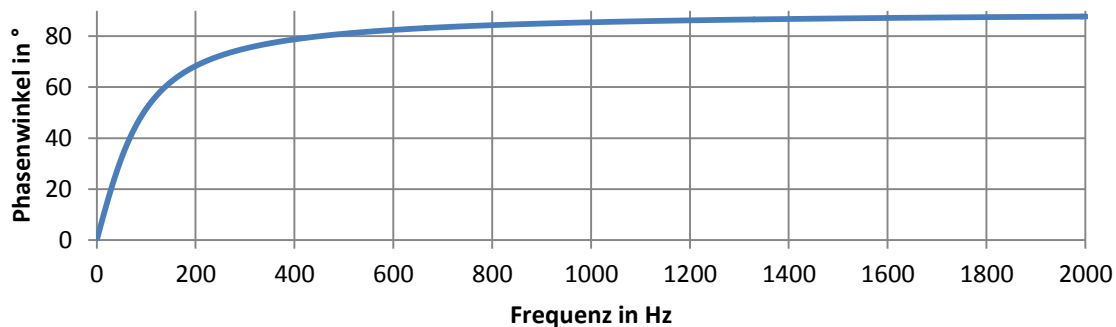


Abbildung 24: Phasenwinkel der Bezugsimpedanz in Abhängigkeit von der Frequenz

Eine genaue Bestimmung der zeit- und frequenzabhängigen Netzimpedanz an einem Netzanschlusspunkt wird anhand von Messungen erreicht. Ein solches Messgerät nach [SCHU13] erzeugt eine Netzanregung durch das Takten einer ohmschen Last. Die Messungen von Strom und Spannung im Leerlauf- und Lastfall liefert nach einer Fourier Transformation einen Frequenzverlauf der Netzimpedanz (Abbildung 25).

Im Bereich 0Hz bis 1000Hz ist die Netzimpedanz dieser Messung näherungsweise:

$$\text{Außenleiter-Null-Leiter:} \quad \underline{Z}_{N_{L,Null}} = 0,386 \, \Omega + j\omega \cdot 0,0732 \, \text{mH} \quad (3.45)$$

Das schnelle Ein- und Ausschalten einer Last mit diesem Messverfahren belastet das Netz. Die Intelligente Ladesäule kompensiert nur einzelne Oberschwingungen und benötigt nur für einzelne Frequenzen einen Schätzwert der Netzimpedanz, ein anderes Messverfahren muss entwickelt werden.

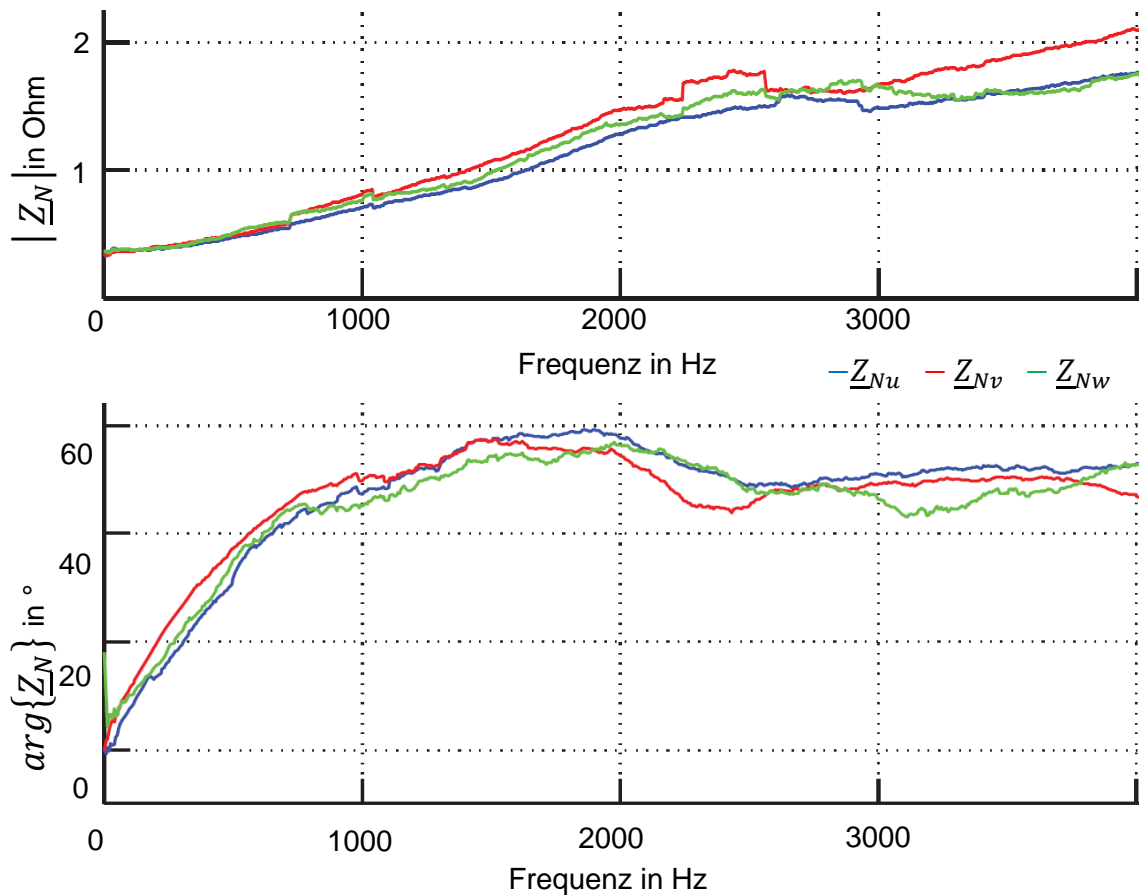


Abbildung 25: Messung der Netzimpedanz im Niederspannungsnetz nach [SCHU13]

3.4 Wirkung der Oberschwingungskompensation

An dem Netzanschlusspunkt der Intelligenten Ladesäule kann eine Spannungsüberschwingung reduziert werden. Die Wirkung der Oberschwingungskompensation im gesamten Netz wird anhand eines Beispiels erläutert. Dabei werden der Zweiweg-Gleichrichter und die Intelligente Ladesäule gemeinsam an das Netz angeschlossen.

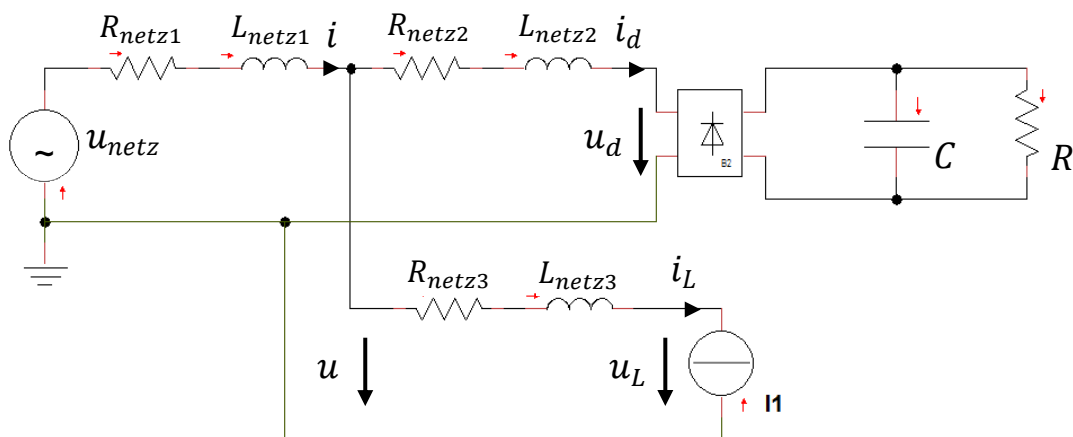


Abbildung 26: Wirkung der Oberschwingungskompensation

Die Netzspannung u_{netz} ist überschwingungsfrei. Die Netzimpedanz Z_{netz} beträgt den Messwert nach Gl. (3.45) und ist wie folgt verteilt:

$$\underline{Z}_{netz3} = \underline{Z}_{netz2} = 0,7 \cdot \underline{Z}_{netz} \quad (3.46a)$$

$$\underline{Z}_{netz1} = 0,3 \cdot \underline{Z}_{netz} \quad (3.46b)$$

Die Verzerrung der Netzspannung u durch den Zweiweg-Gleichrichter ohne Kompensation ist auf Abbildung 27 dargestellt.

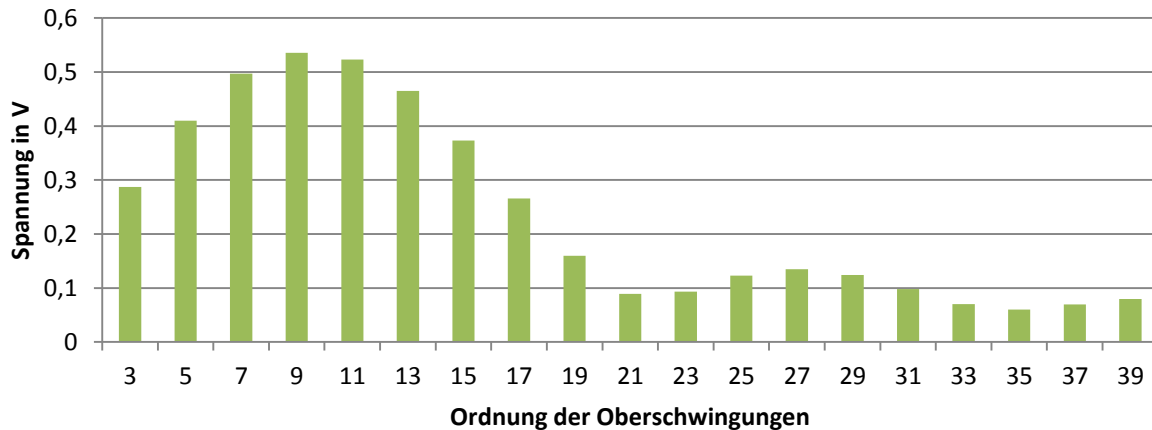


Abbildung 27: Amplitude der Oberschwingungen der Spannung u ohne Kompensation

Die Kompensation der 5. Oberschwingung mit der Intelligenten Ladesäule nach Gl. (3.41) liefert folgende Ergebnisse:

5. OS	Spannung in V			Strom in A		
	Netz: u	LS: u_L	B2: u_d	Netz: i	LS: i_L	B2: i_d
ohne Komp.	0,41	0,41	1,37	1,13	0	1,13
mit Komp.	0,30	0,05	1,26	0,83	0,3	1,13

	Wirkleistung in W			Scheinleistung in VA		
	Netz	LS	B2	Netz	LS	B2
ohne Komp.	206	0	204	431	0	431
mit Komp.	206	0	204	413	48	431

LS: Ladesäule, B2: Zweiweg-Gleichrichter, Komp.:Kompensation, Ampl.:Amplitude, OS: Oberschwingung

Tabelle 8: Wirkung der Oberschwingungskompensation

Durch die phasennegierte Einprägung der 5. Stromoberschwingung sinkt die 5. Spannungsoberschwingung im gesamten Netz und ist am Netzanschlusspunkt der Ladesäule fast vollständig kompensiert. Die Intelligente Ladesäule nimmt einen Teil der 5. Stromoberschwingung des Zweiweg-Gleichrichters auf, dieser Strom fließt in geringeren Mengen in das Netz. Die Scheinleistung des Netzes sinkt, sein Leistungsfaktor hat sich demzufolge verbessert. Die Kompensation der 5. Oberschwingung hat keinen Einfluss auf die anderen Oberschwingungen.

3.5 Zusammenfassung

Mit dem Beispiel eines einphasigen Zweiweg-Gleichrichters wurde gezeigt, dass Verbraucher mit nichtlinearer Kennlinie Oberschwingungsströme einspeisen, wenn sie an eine ideale Netzspannung angeschlossen sind. Diese Oberschwingungsströme verursachen einen Spannungsabfall über der Netzimpedanz und sind die Ursache für die Entstehung von Oberschwingungsspannungen im gesamten Netz.

Dem gegebenüber gestellt wurde gezeigt, dass in einem Oberschwingungsbehafteten Netz das gezielte Verbrauchen einer Stromoverschwingung zu einer Reduzierung der Spannungsüberschwingung führt. Durch das phasennegierte Einprägen einer Stromoverschwingung reduziert die Ladesäule die Spannungsüberschwingung und verbessert damit die Spannungsqualität im gesamten Netz. Eine Ladesäule kann häufig nur einen Teil der Spannungsüberschwingung kompensieren, verstärkt wird die Wirkung durch mehrere Ladesäulen.

Eine Schätzung der Netzimpedanz wurde als notwendig erklärt, um einen optimalen Wirkungsgrad der Kompensation zu erreichen. Der ohmsch-induktive Charakter der Netzimpedanz im Niederspannungsnetz wurde erläutert.

4 Regelung der Oberschwingungskompensation

4.1 Zweipulswechselrichter

Die Steuerung des elektrischen Energieflusses in der Intelligenten Ladesäule findet mit einem dreiphasigen Zweipulswechselrichter statt. Der Wechselrichter beinhaltet sechs Transistoren, die in drei Transistorzweige (TR1 & TR4), (TR2 & TR5), (TR3 & TR6) angeordnet sind, und sechs Freilaufdioden zum Schutz der Transistoren vor negativen Spannungen (Abbildung 28). Alle drei Transistorzweige sind mit der Gleichspannung des Zwischenkreises verbunden, die Gleichspannung wird von dem Zwischenkreis-kondensator geglättet. Auf der Netzseite sind drei Glättungsdrosseln \underline{Z}_d , die jeden Transistorzweig mit einem Strang des Netzes verbinden.

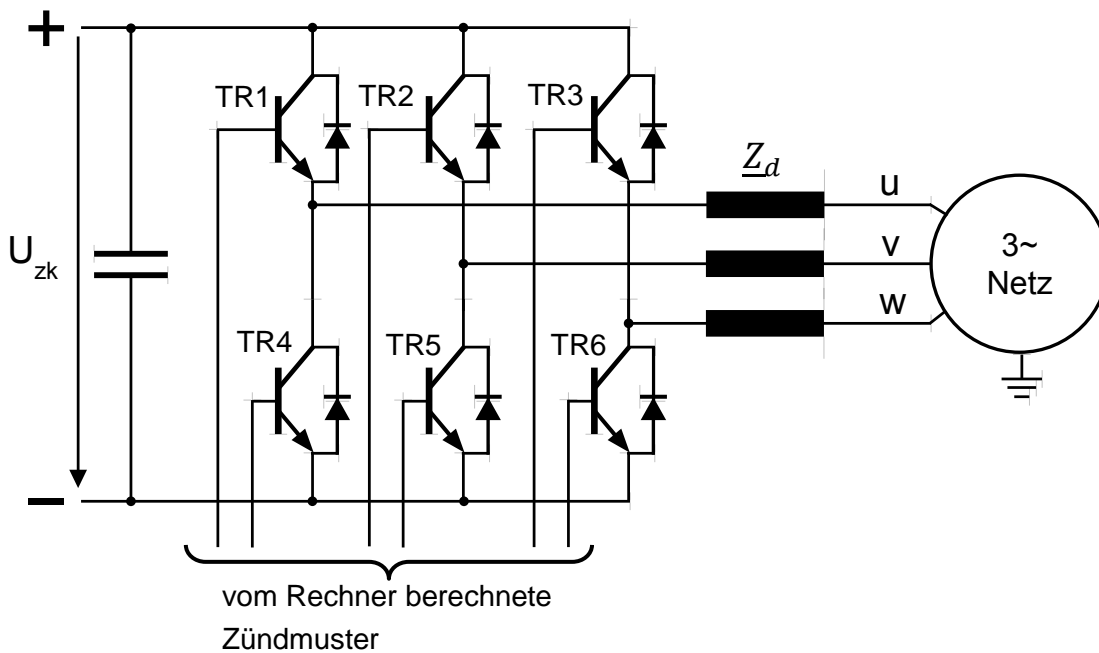


Abbildung 28: dreiphasiger Zweipulswechselrichter der Intelligenten Ladesäule

Während des Betriebs des Wechselrichters ist in jedem Transistorzweig immer ein Transistor leitend und ein Transistor sperrend. Der Zustandswechsel der Transistorzweige führt dazu, dass jeder Strang u, v, w des dreiphasigen Netzes immer mit dem negativen Potential oder mit dem positiven Potential der Zwischenkreisspannung geschaltet ist. Die Schaltungskombinationen der zwei Potentiale der Gleichspannung mit den drei Netzsträngen führen zu $2^3 = 8$ logischen Zuständen bzw. 8 Standardvektoren \underline{u}_0 bis \underline{u}_7 . Für zwei Schaltzustände sind alle drei Stränge u, v, w , mit dem gleichen Potential der Zwischenkreisspannung verbunden, sie sind mit den Nullvektoren \underline{u}_0 und \underline{u}_7 gekennzeichnet. Für die restlichen sechs Schaltzustände verteilen sich die drei Stränge

u, v, w , auf die zwei Potentiale der Zwischenkreisspannung und bilden einen Spannungsteiler (Anlage 1). Es gilt folgendes:

$$|\underline{u}_s|_{max} = |\underline{u}_1| = |\underline{u}_2| = |\underline{u}_3| = |\underline{u}_4| = |\underline{u}_5| = |\underline{u}_6| = \frac{2}{3} U_{ZK} \quad (4.47a)$$

Anhand der Raumzeigerdarstellung der drei Wechselrichterausgangsspannungen wird die räumliche Lage der Standardvektoren in ein kartesisches Koordinatensystem definiert (Abbildung 29). Das sukzessive Schalten zweier benachbarter Standardvektoren \underline{u}_1 bis \underline{u}_6 und zweier Nullvektoren \underline{u}_0 und \underline{u}_7 in einer Taktperiode T_p bildet den Raumzeiger \underline{u}_s .

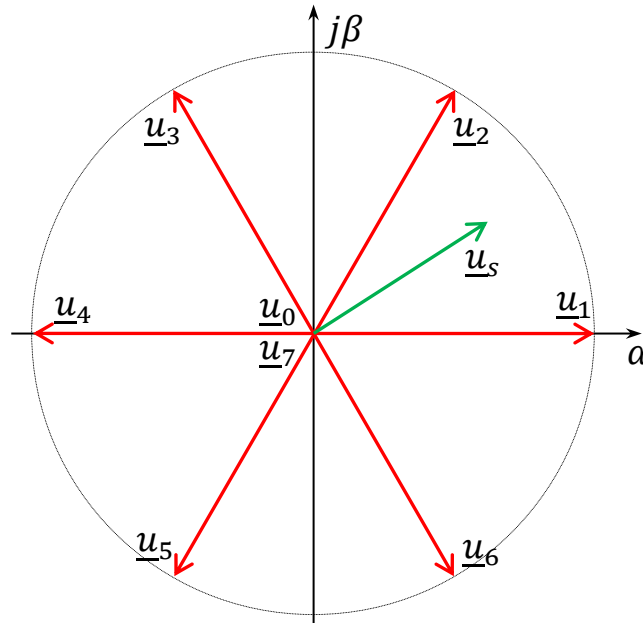


Abbildung 29: Raumzeigerdarstellung der Wechselrichterausgangsspannungen

Mit der Raumzeigermodulation (RZM) werden aus dem Raumzeiger \underline{u}_s die Schaltzeiten für jedes Transistorpaar innerhalb einer Taktperiode T_p berechnet. Die Symmetrie der Schaltzeiten in einer Taktperiode T_p (Abbildung 30) erzeugt ein stabiles Pulsmuster, welches eine Synchronisation der Messgröße und Stellgröße des Regelkreises ermöglicht. Vor der Raumzeigermodulation wird der Raumzeiger \underline{u}_s auf die Zwischenkreisspannung normiert.

$$\underline{u}_{s,norm} = \frac{3}{2} \cdot \frac{\underline{u}_s}{U_{ZK}} \quad (4.48)$$

Die Implementierung der Raumzeigermodulation ist im [QUAN99] ausführlich beschrieben und wurde mit der Diplomarbeit von Herrn Fabian Merkel bereit realisiert.

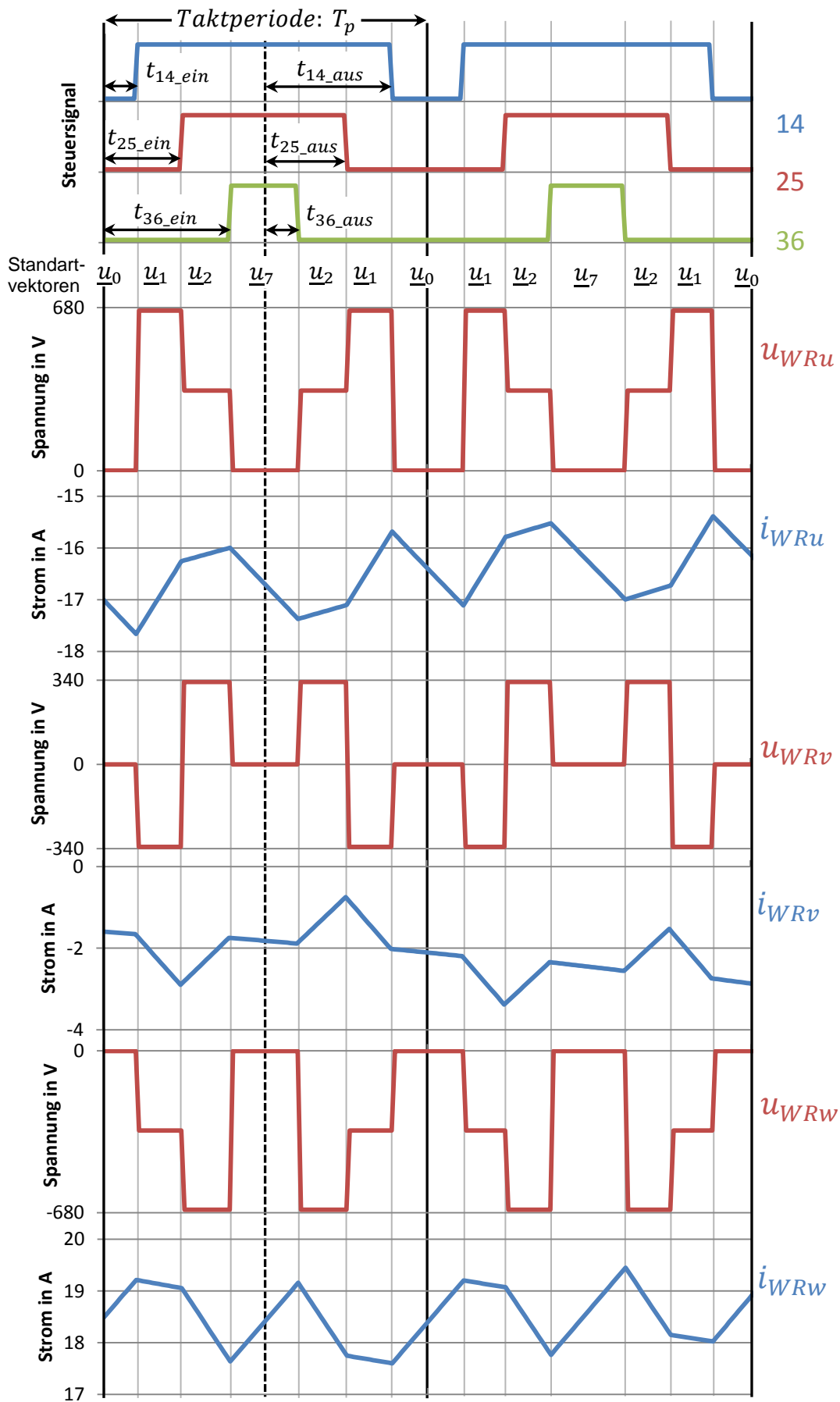


Abbildung 30: Transistorpulsmuster und Wechselrichter Ausgangsströme und –spannungen für zwei Taktperioden

Die Systemfunktion des Wechselrichters mit der Raumzeigermodulation entspricht einem Totzeitglied und ist wie folgt definiert:

$$\underline{u}_{WR}(t) = \underline{u}_s\left(t - \frac{T_p}{2}\right) \quad (4.49a)$$

$$\underline{u}_s = \hat{u}_s \cdot e^{j\theta_s} = u_{s\alpha} + ju_{s\beta} \quad (4.49b)$$

$$\underline{u}_{WR} = \frac{2}{3} \left(u_{WRu} + \underline{a} \cdot u_{WRv} + \underline{a}^2 \cdot u_{WRw} \right) \quad (4.49c)$$

Die Spannungsdifferenz zwischen der Wechselrichterspannung und der Netzspannung liegt über die Glättungsdrosseln an und definiert den Wechselrichterstrom.

$$\underline{i}_{WR} = \frac{\underline{u}_{WR} - \underline{u}_{netz}}{\underline{Z}_d} \quad (4.50a)$$

$$\underline{u}_{netz} = \frac{2}{3} \left(u_{netz,u} + \underline{a} \cdot u_{netz,v} + \underline{a}^2 \cdot u_{netz,w} \right) \quad (4.50b)$$

Aus den Messwerten der Netzspannung erfolgt der netzgesteuerte Betrieb des Wechselrichters (Abbildung 31) und Gl.(4.51).

$$\underline{i}_{WR}(t) = \frac{\underline{u}_{netz}(t - \tau_{WR} - \tau_{pross}) - \underline{u}_{netz}(t)}{\underline{Z}_d} \quad (4.51)$$

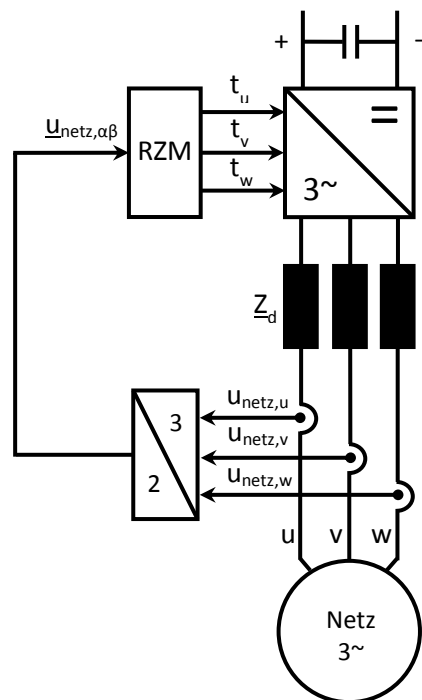


Abbildung 31: netzgesteuerter Betrieb des Wechselrichters

Die Reaktionszeit der Steuerung ist durch die Totzeiten des Wechselrichters und des Prozessors definiert und entspricht ein bis zwei Taktperioden bzw. 100µs bis 200µs. Diese hohe Dynamik ermöglicht eine sofortige Nachbildung der Netzspannung am Ausgang des Wechselrichters und sichert eine schnelle Begrenzung des Wechselrichterstromes beim Einschalten ans Netz.

4.2 Regelung der Grundschwingung

Eine präzise Einstellung des Wechselrichterstromes kann mit dem netzgesteuerten Betrieb nicht erreicht werden, die Messung und die Regelung des Wechselrichterstromes sind notwendig.

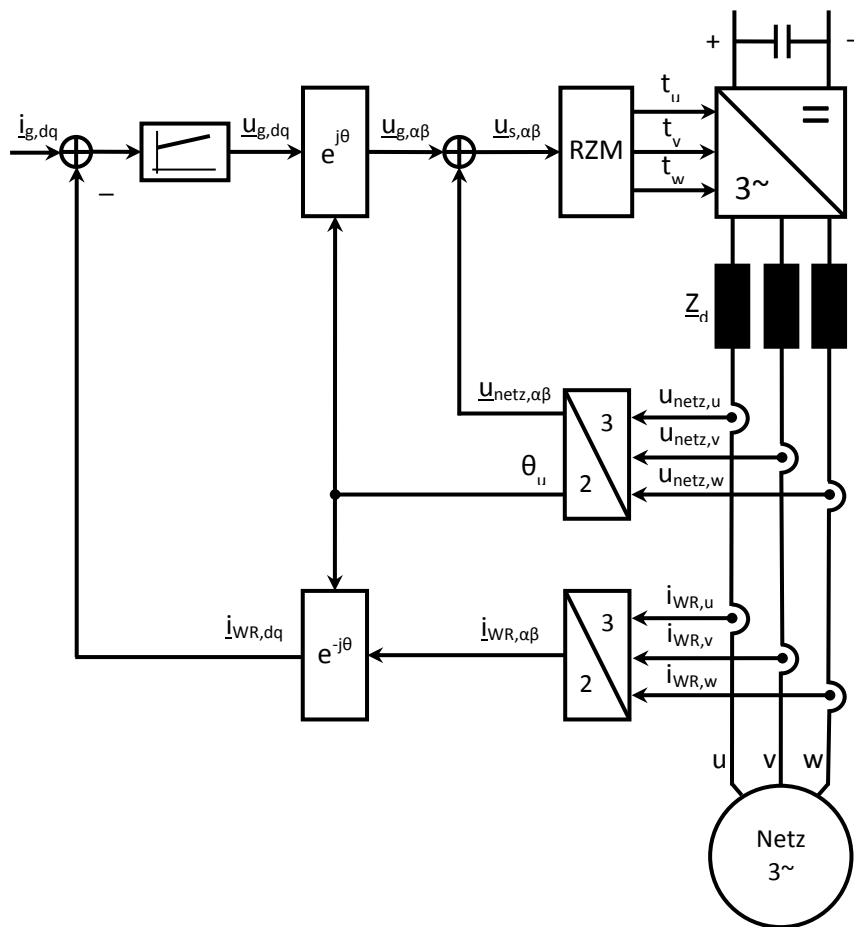


Abbildung 32: Regelung der Grundschwingung des Wechselrichterstromes

Aus den drei Messwerten des Wechselrichterstromes wird nach Gl. (4.52a) ein Raumzeiger \underline{i}_{WR} gebildet, der nach (4.52b) mit der Netzspannung synchronisiert wird.

$$\underline{i}_{WR,\alpha\beta} = \frac{2}{3} (i_{WR,u} + \underline{a} \cdot i_{WR,v} + \underline{a}^2 \cdot i_{WR,w}) \quad (4.52a)$$

$$\underline{i}_{WR,dq} = \underline{i}_{WR,\alpha\beta} \cdot e^{-j\theta_u} \quad (4.52b)$$

$$\theta_u = \arg\{\underline{u}_{netz,\alpha\beta}\} \quad (4.52c)$$

Die reelle Achse des rotierenden d,q Koordinatensystems, ist mit dem rotierenden Raumzeiger der Netzspannung verknüpft. Die Projektierung des Raumzeigers \underline{i}_{WR} in dem d,q Koordinatensystem ermöglicht die Einstellung eines festen Phasenwinkels $\varphi_{i_{WR}}$ zwischen den Strom- und Spannungsraumzeigern (Abbildung 33).

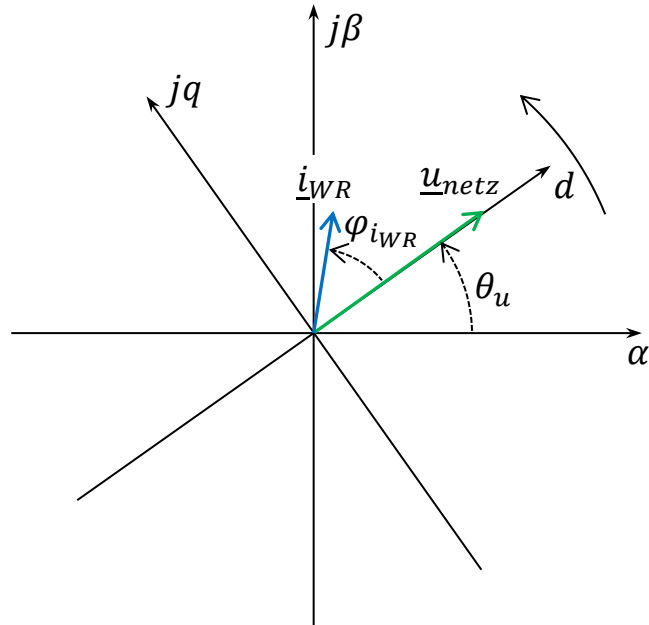


Abbildung 33: Park-Transformation des Raumzeigers des Wechselrichterstromes

Der Wechselrichterstrom wird mit einem PI-Regler eingestellt. Der PI-Regler ist auf die Zeitkonstante der Regelstrecke festgelegt, die im Wesentlichen durch die Glättungsdrosseln \underline{Z}_d bestimmt ist. Die Totzeiten des Wechselrichters und des Prozessors können vernachlässigt werden.

$$\tau_d = \frac{L_d}{R_d} \gg \tau_{WR} + \tau_{pross} \quad (4.53)$$

Die Stellgröße $\underline{u}_{g,dq}$ am Ausgang der PI-Regler wird nach Gl. (4.54a) in das feste α,β Koordinatensystem rücktransformiert und mit dem Raumzeiger der Netzspannung addiert Gl. (4.54b).

$$\underline{u}_{g,\alpha\beta} = \underline{u}_{g,dq} \cdot e^{j\theta_u} \quad (4.54a)$$

$$\underline{u}_{s,\alpha\beta} = \underline{u}_{g,\alpha\beta} + \underline{u}_{netz,\alpha\beta} \quad (4.54b)$$

Anhand des Sollwerts $\underline{i}_{g,dq}$ kann der bidirektionale Energiefluss und die Blindleistungskompensation der Intelligenten Ladesäule eingestellt werden.

$$\underline{i}_{WRv,dq} = \frac{2}{20ms} \cdot \int_t^{t+20ms} (i_{WR,u} \cdot e^{-jv\theta_{DFT}}) dt = \hat{i}_{WRv} \cdot e^{j\varphi_{i_{WRv}}} \quad (4.55)$$

Die Rotation des d,q Koordinatensystems ist mit dem Zeiger $e^{jv\theta_{DFT}}$ der DFT synchronisiert. Der DFT-Zeiger rotiert mit einem v -fach von 50Hz und wird per Programm erzeugt. Das Zeitfenster der gleitenden DFT beträgt 20 ms und führt zu einer Totzeit von 10 ms.

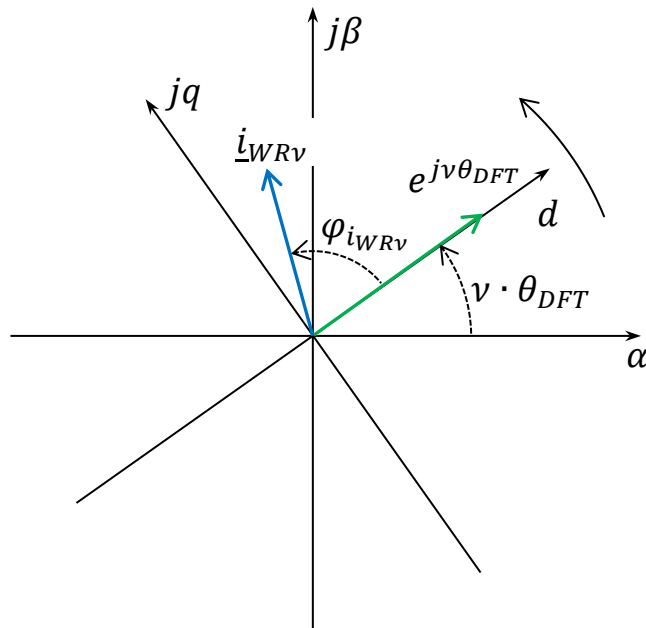


Abbildung 35: ruhender Zeiger einer Stromüberschwingung

Die Stromüberschwingung wird mit einem PI-Regler eingestellt. Dabei wird der PI-Regler auf der 10 ms Totzeit der DFT festgelegt. Die Stellgröße $\underline{u}_{v,dq}$ am Ausgang des PI-Reglers wird nach Gl. (4.56a) in das feste α, β Koordinatensystem rücktransformiert und mit den Raumzeigern der Netzspannung und der Grundschwingungsregelung addiert Gl. (4.56b).

$$\underline{u}_{v,\alpha\beta} = \underline{u}_{v,dq} \cdot e^{jv\theta_{DFT}} \quad (4.56a)$$

$$\underline{u}_{s,\alpha\beta} = \underline{u}_{v,\alpha\beta} + \underline{u}_{g,\alpha\beta} + \underline{u}_{netz,\alpha\beta} \quad (4.56b)$$

Anhand des Sollwerts $\hat{i}_{v,dq}$ kann die Amplitude und der Phasenwinkel der Stromüberschwingung eingestellt werden.

Im Kapitel 1.4 wurde erwähnt, dass die Überschwingungen eines dreiphasigen Netzes in Nullsystem, Mitsystem und Gegensystem klassifiziert sind.

Der Wechselrichter in der Intelligenten Ladesäule ist mit dem Neutralleiter nicht verbunden und kann deshalb keine Nullsystem Stromkomponente erzeugen, die Kompensation von Nullsystemüberschwingungen ist damit ausgeschlossen.

Die Regelung einer Mitsystemstromoberschwingung wurde bereits beschrieben und wird angewendet für die Regelung einer Gegensystemstromoberschwingung mit folgenden Änderungen:

Wenn der Raumzeiger \underline{u}_s entgegen dem Uhrzeigersinn rotiert, besitzt er eine positive Kreisfrequenz und der Wechselrichter erzeugt einen Mitsystemstrom.

Wenn der Raumzeiger \underline{u}_s im Uhrzeigersinn rotiert, besitzt er eine negative Kreisfrequenz und der Wechselrichter erzeugt einen Gegensystemstrom.

Die Rotationsrichtung des Raumzeigers $\underline{u}_{v,\alpha\beta}$ ist während der Rücktransformation der Stellgröße $\underline{u}_{v,dq}$ in dem festen α, β Koordinatensystem festgelegt. Für die Gegensystemstromoberschwingung erfolgt die Rücktransformation nach Gl. (4.57) mit einer negativen Kreisfrequenz: $-\nu \cdot \theta_{DFT}$.

$$\underline{u}_{-v,\alpha\beta} = \underline{u}_{-v,dq} \cdot e^{-j\nu\theta_{DFT}} \quad (4.57)$$

Die Messung der Stromoberschwingung muss angepasst werden, die gleitende DFT wird mit einer negativen Kreisfrequenz berechnet.

$$\hat{i}_{WR-v,dq} = \frac{2}{20ms} \cdot \int_t^{t+20ms} (i_{WR,u} \cdot e^{j\nu\theta_{DFT}}) dt = \hat{i}_{WR-v} \cdot e^{j\varphi_{i_{WR-v}}} \quad (4.58)$$

4.4 Regelung der Oberschwingungskompensation

Für die Oberschwingungskompensation (Abbildung 36) wird der Phasenwinkel zwischen der Stromoberschwingung und der Spannungsoberschwingung des Netzes eingestellt. Aus den Phasenwinkeln der Spannungsoberschwingung und der Netzimpedanz wird ein Sollwert für die Regelung der Stromoberschwingung ermittelt. In diesem Abschnitt wird auf die präzise Messung der Spannungsoberschwingung eingegangen und abschließend wird der Rahmen eines zukünftigen Algorithmus zur Schätzung der Netzimpedanz festgelegt.

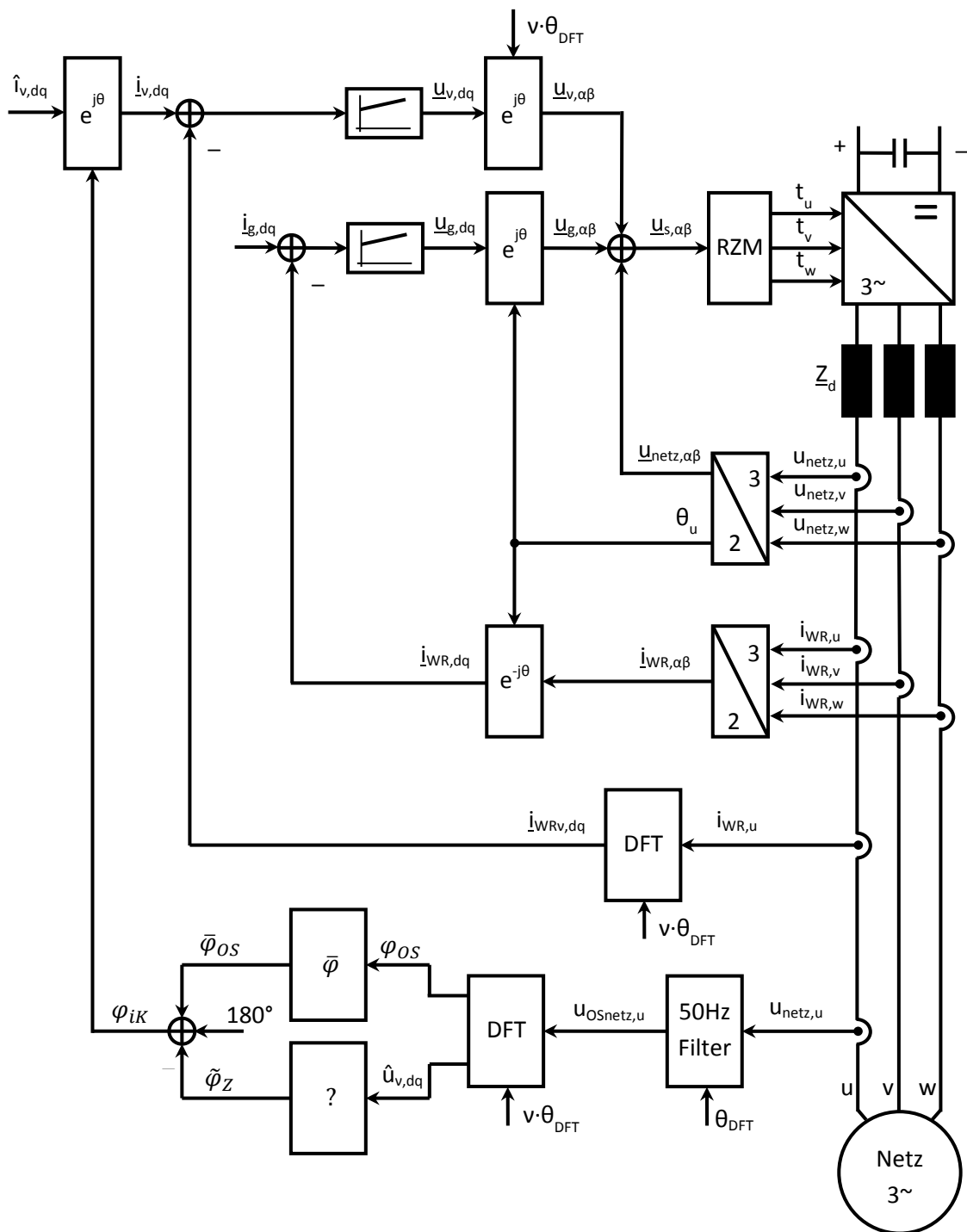


Abbildung 36: Regelung der Oberschwingungskompensation

4.4.1 50Hz Filter

Die Amplitude der Grundschwingung der Netzspannung entspricht einem 100-fachen der Amplitude der restlichen Oberschwingungen. Die starke Dämpfung der Grundschwingung ist äußerst wichtig für eine wertvolle Auswertung ohne Abbruchfehler der

Spannungsüberschwingung. Eine Synchronisierung der Analysefrequenz der DFT mit der Netzfrequenz sichert eine starke Dämpfung der Grundschwingung, liefert aber eine diskontinuierliche Messung des Phasenwinkels der Oberschwingung. Die Regelung der Oberschwingungskompensation setzt eine kontinuierliche Messung der Phasenwinkel der Spannungsüberschwingung voraus. Die Grundschwingung wird mit einem 50Hz Filter gedämpft.

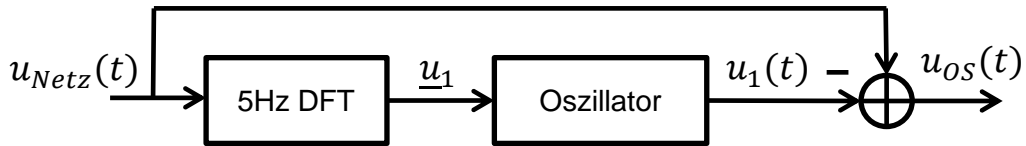


Abbildung 37: 50Hz Filter vor der Oberschwingungsmessung der Netzspannung

Die Grundschwingung der Netzspannung wird mit der gleitenden DFT ausgewertet, mit einem Oszillator generiert und von der Netzspannung abgezogen. Die 100ms Totzeit der 5Hz DFT begrenzt die Reaktionszeit des 50Hz Filters. Die Grundschwingung der Netzspannung bleibt stabil und kann innerhalb einer Sekunde als nahezu konstant angenommen werden. Der 50Hz Filter hat eine Dämpfung von 46 dB. Nach der Filterung ist die Amplitude der Grundschwingung und der Oberschwingungen ähnlich groß.

4.4.2 Messung der Oberschwingungsspannung

Grundlage für die Regelung der Oberschwingungskompensation ist eine hochwertige Auswertung der Spannungsüberschwingung. Die gleitende DFT kann mit verschiedenen Auflösungen eingestellt werden. Mit einem Zeitfenster von 200ms hat die gleitende DFT eine Auflösung von 5Hz, die restlichen Oberschwingungen der Netzspannung werden somit ausreichend gedämpft.

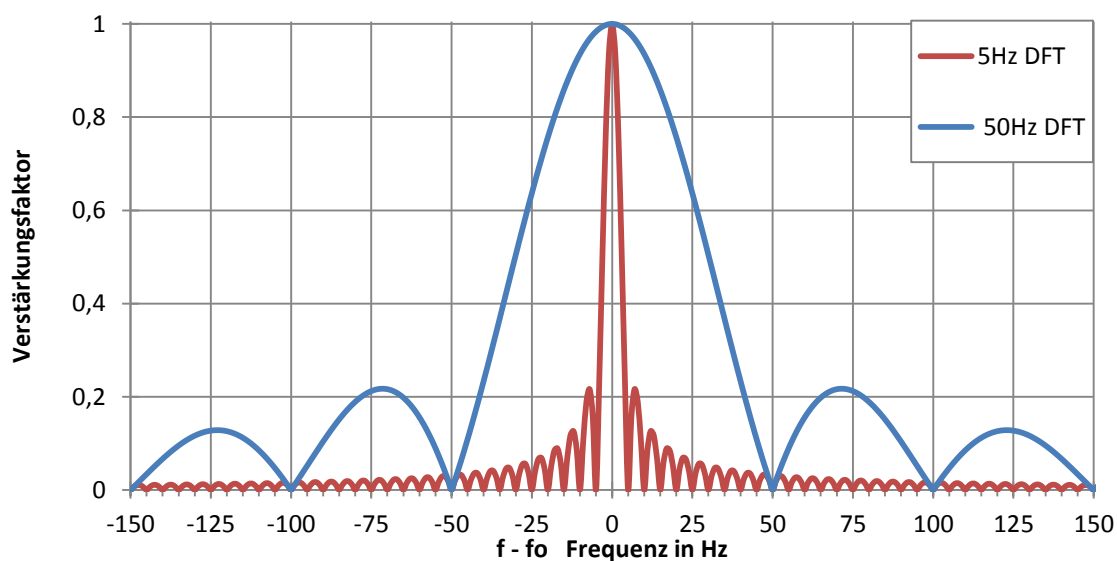


Abbildung 38: Filtereigenschaften der DFT bei 5Hz und 50Hz Auflösung

In Kapitel 3.1 wurde beschrieben, dass die Frequenzen der Oberschwingungsspannungen ein Vielfaches der Netzfrequenz sind. Die Analysefrequenz der DFT ist mit der Netzfrequenz nicht synchronisiert. Bei Oberschwingungen höherer Ordnung werden die Analysefrequenz der DFT und die Oberschwingungsfrequenz zunehmend abweichen.

$$\Delta f = \nu \cdot |f_{DFT} - f_{netz}| = \nu \cdot |50\text{Hz} - f_{netz}| \quad (4.59)$$

Die 5Hz DFT bildet einen Bandpassfilter mit schmaler Bandbreite (Abbildung 38). Bei einer Frequenzabweichung Δf größer als 1Hz wird die Spannungsüberschwingung falsch ausgewertet. Bandpassfilter mit breiterer Bandbreite werden durch die Gruppierung von benachbarten Spektralbändern erzeugt. Jedes Spektralband ist ein Zeiger in einem rotierenden Koordinatensystem. Diese Zeiger müssen vor ihrer Addition in das gleiche rotierende Koordinatensystem mit einer Park-Transformation transformiert werden.

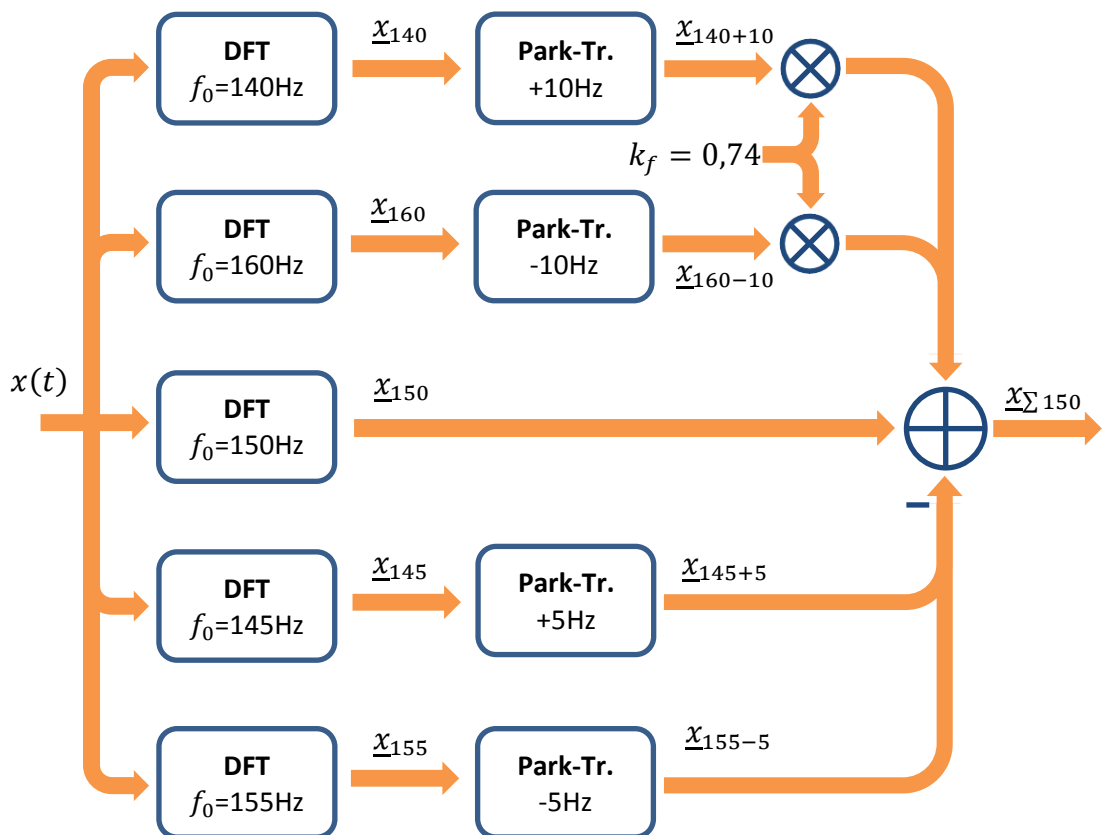


Abbildung 39: Bandpassfilter für die Messung der Spannungsüberschwingungen (Gruppierung +/- 10Hz)

Die Anzahl von Spektralbändern definiert die Bandbreite des Bandpassfilters. Der Faktor k_f beeinflusst die Form des Bandpassfilters.

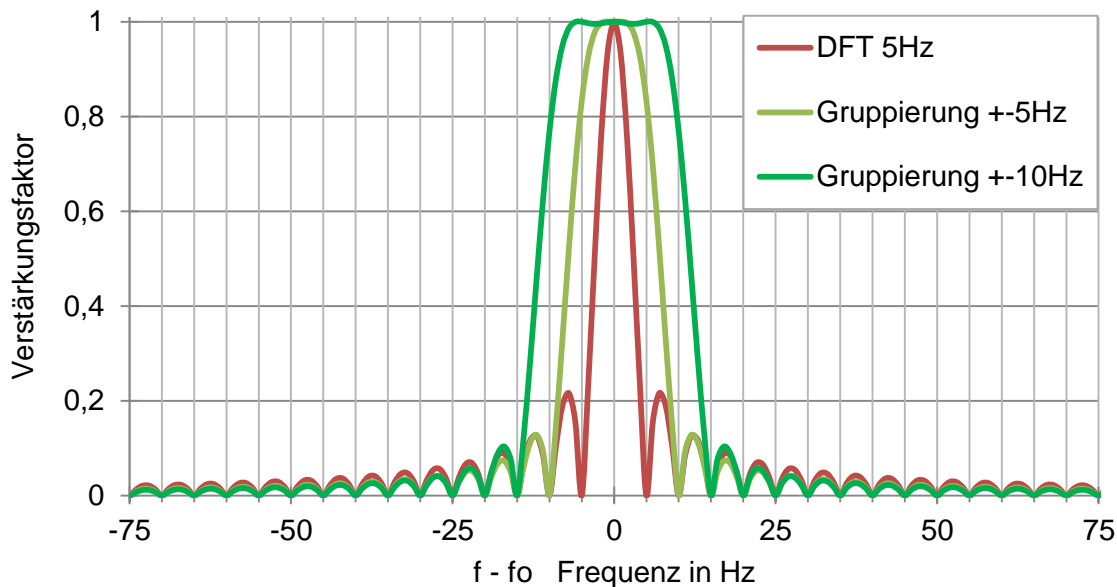


Abbildung 40: Bandpassfilterung für die Messung der Spannungsüberschwingungen

Diese Gruppierung wurde nur für die Simulation der Oberschwingungskompensation angewendet und in der Intelligenten Ladesäule noch nicht implementiert.

4.4.3 Mittelwert der Phasenwinkel

Die Messung des Phasenwinkels der Netzspannung, ist eine wichtige Komponente der Kompensation der Oberschwingungen. Die Abbruchfehler der DFT verursachen Schwingungen und die transienten Spannungsänderungen rufen kurze Abweichungen der Phasenwinkelmesswerte hervor. Diese Abweichungen der Phasenwinkel werden durch eine Art Mittelwert unterdrückt. Die Frequenzabweichung Δf nach Gl (4.59) zwischen der Analysefrequenz der DFT und der Oberschwingungsfrequenz bildet sich im Phasenwinkel der ausgewerteten Oberschwingung ab. Der Phasenwinkel ist eine Sägezahnfunktion von 0° bis 360° .

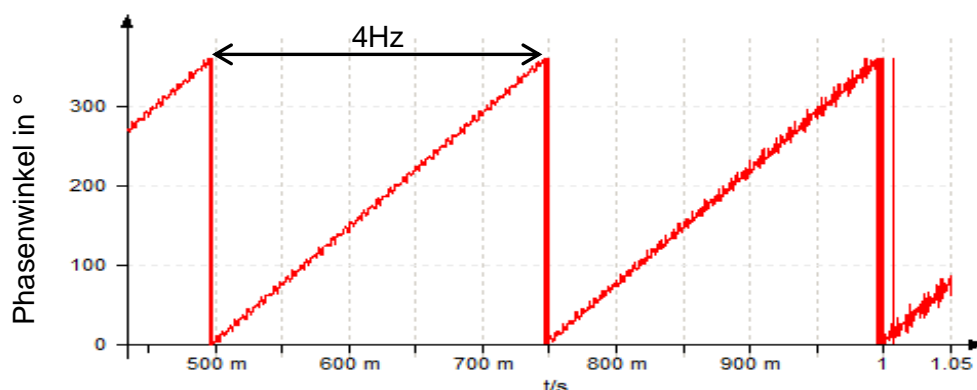


Abbildung 41: DFT Messung des Phasenwinkels der 11. OS mit $f_{DFT} = 550\text{Hz}$, $f_{OS} = 554\text{Hz}$

Durch den Sprung bei $360^\circ = 0^\circ$ ist der Phasenwinkel keine lineare Funktion. Die Messabweichungen können nicht direkt mit einem Mittelwert unterdrückt werden. Anhand der

Kreisfrequenz ω kann ein Mittelwert gebildet werden, der erneut synchronisiert wird (Abbildung 42). Diese Kreisfrequenz wird auch für die Korrektur der Totzeit der DFT benötigt.

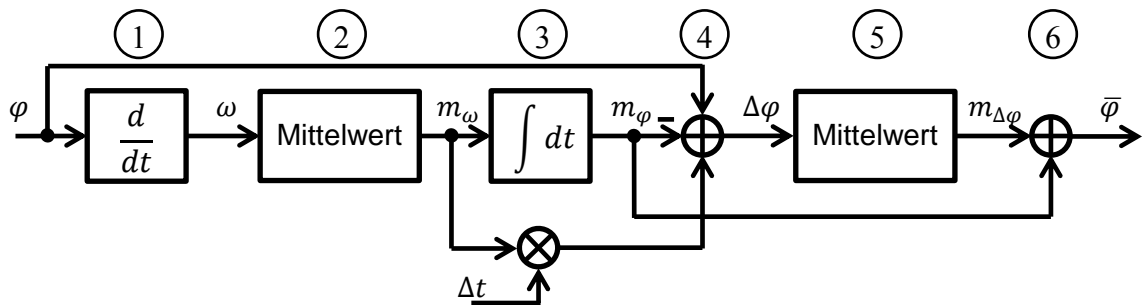


Abbildung 42: Dämpfung der Messabweichungen des Phasenwinkels

Die sechs Schritte der Dämpfung der Messabweichungen des Phasenwinkels:

1. Ableitung des Phasenwinkels $\varphi \rightarrow$ Kreisfrequenz ω

$$\omega = \frac{d\varphi}{dt} \quad (4.60)$$

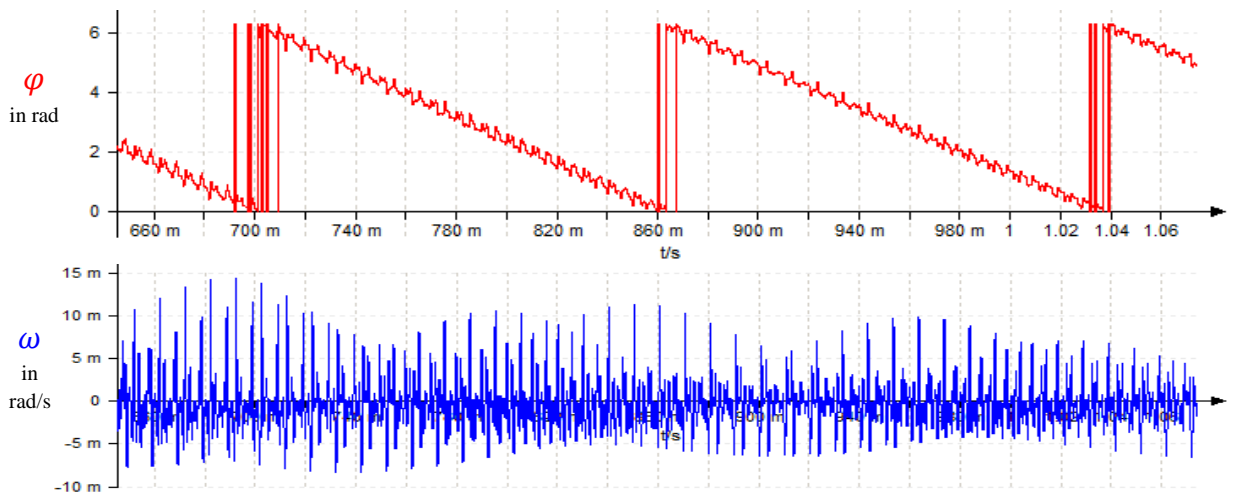


Abbildung 43: Ableitung des Phasenwinkels = Kreisfrequenz

2. gleitender Mittelwert der Kreisfrequenz m_ω

$$m_\omega = \frac{1}{20ms} \int_t^{t+20ms} \omega \cdot dt \quad (4.61)$$

Das Zeitfenster des gleitenden Mittelwerts beträgt 20ms, die Oberschwingungen die in Form von Abbruchfehlern noch präsent sind, befinden sich in der Nähe der Polstellen des gleitenden Mittelwertes und werden stark gedämpft.

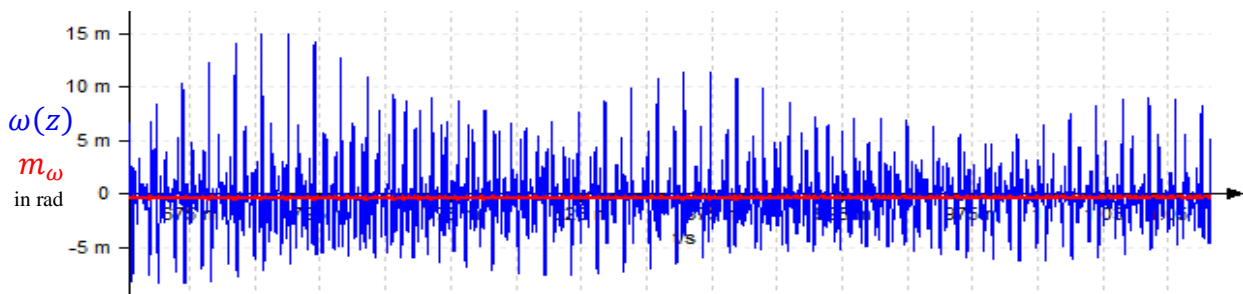


Abbildung 44: gleitender Mittelwert der Kreisfrequenz

3. Integration der gemittelten Kreisfrequenz $m_\omega \rightarrow$ gemittelter Phasenwinkel m_φ

$$m_\varphi = \int_0^t m_\omega \cdot dt \quad (4.62)$$

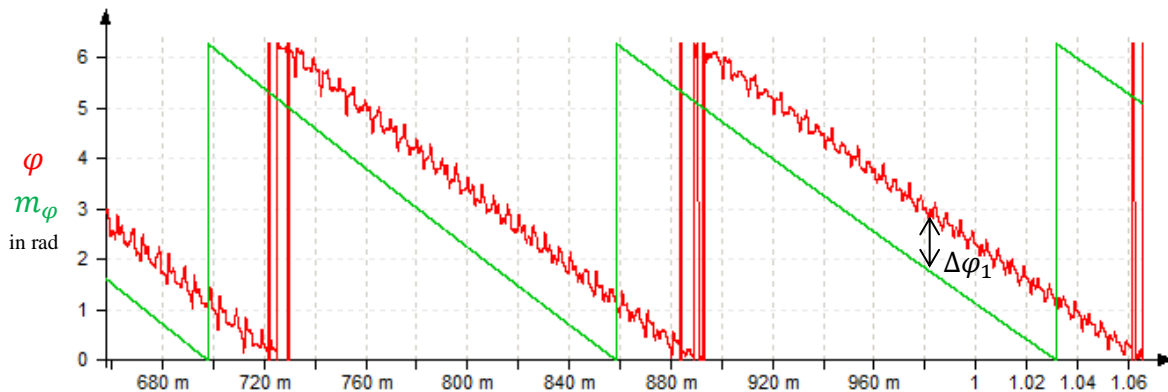


Abbildung 45: Integration der gemittelten Kreisfrequenz → gemittelter Phasenwinkel

Die Abbildung 45 zeigt die Dämpfung der Abweichungen des Phasenwinkels.

4. Korrektur der Phasenverschiebung und Totzeiten

Eine Phasenverschiebung $\Delta\varphi_1$ zwischen m_φ und φ ist entstanden und wird korrigiert.

$$\Delta\varphi_1 = \varphi - m_\varphi \quad (4.63)$$

Die Phasenwinkeldifferenz $\Delta\varphi_1$ ist konstant und enthält die Abweichungen des Phasenwinkels φ . Diese Abweichungen werden mit einem gleitenden Mittelwert gedämpft. Die 100ms Totzeit der 5Hz DFT wird durch die Addition eines Phasenwinkels $\Delta\varphi_2$ korrigiert. Die gesamte Phasenkorrektur ist:

$$\Delta\varphi = \Delta\varphi_1 + \Delta\varphi_2 \quad (4.64a)$$

$$\Delta\varphi_2 = m_\omega \cdot 0,1s \quad (4.64b)$$

5. gleitender Mittelwert der Phasenkorrektur $m_{\Delta\varphi}$

Die Phasenverschiebung $\Delta\varphi$ enthält Abweichungen des Phasenwinkels φ und der Kreisfrequenz m_{ω} , die mit Hilfe eines gleitenden Mittelwerts gedämpft werden.

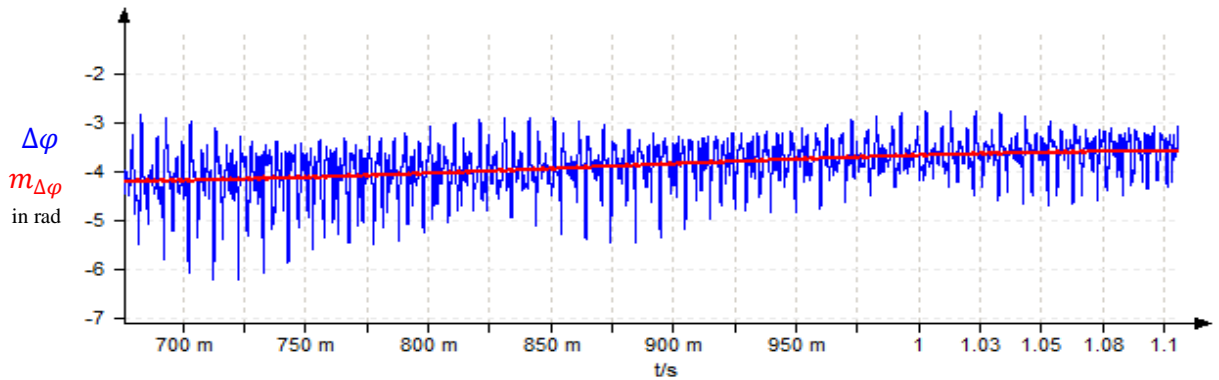
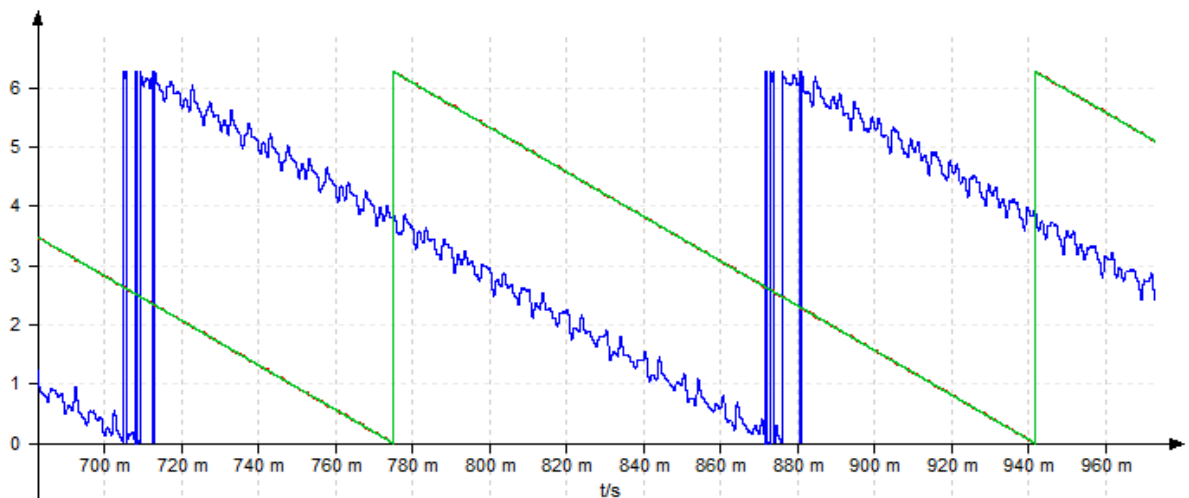


Abbildung 46: gleitender Mittelwert der Phasenkorrektur

6. Ergebnis

Die gemittelte Phasenkorrektur $m_{\Delta\varphi}$ wird zu dem in Schritt 3 gemittelten Phasenwinkel m_{φ} addiert.

$$\bar{\varphi} = m_{\varphi} + m_{\Delta\varphi} \quad (4.65)$$



Phasenwinkel nach der DFT: φ
Mittelwert und Korrektur des Phasenwinkels: $\bar{\varphi}$
tatsächlicher Phasenwinkel der Oberschwingung: φ_t

Abbildung 47: Simulationsergebnisse der Korrektur und Filterung eines Phasenwinkels

Der Phasenwinkel $\bar{\varphi}$ stimmt mit dem tatsächlichen Phasenwinkel der Oberschwingung φ_t überein, die rote Kurve ist von der grünen Kurve bedeckt (Abbildung 47).

4.4.4 Regelung für die Schätzung der Netzimpedanz

Für die Oberschwingungskompensation nach Gl. (3.41) wird ein Schätzwert des Phasenwinkels der Netzimpedanz benötigt. Die Wirkung der Kompensation hängt von der Amplitude und dem Phasenwinkel des Kompensationsstromes ab. Die Amplitude des Kompensationsstromes steuert die Intensität der Kompensation und wird sehr selten geändert. Der Phasenwinkel des Kompensationsstromes steuert den Wirkungsgrad der Kompensation und muss ständig angepasst werden. Durch die kontinuierliche Berechnung eines Schätzwertes des Phasenwinkels der Netzimpedanz wird der optimale Wirkungsgrad der Kompensation erzielt. Ein Indikator für die Verbesserung oder Verschlechterung des Wirkungsgrades der Kompensation ist die Amplitude der kompensierten Spannungsüberschwingung. Diese Amplitude ist aber von mehreren Parametern abhängig, die als Störquellen auftreten. Durch die Filterung dieser Störungen soll die Kompensationswirkung von dem Einfluss der anderen Parameter entkoppelt werden. Die Amplitude einer Spannungsüberschwingung ist abhängig von der Netzimpedanz, von Störaussendungen nichtlinearer Verbraucher sowie von Zwischenharmonischen und Flickern, die Messabweichungen verursachen. Ein Regelalgorithmus für die Schätzung der Netzimpedanz geht über diese Arbeit hinaus, die Oberschwingungskompensation wird für eine reine ohmsche Netzimpedanz betrieben.

4.5 Zusammenfassung

Die Steuerung des Zweipuls-Wechselrichters der Intelligenten Ladesäule erfolgt mit der Raumzeigermodulation nach [QUAN99]: aus den Koordinaten eines Raumzeigers werden die Pulsmuster für die drei Transistorzweige des Wechselrichters berechnet. Drei Glättungsdrosseln verbinden den Wechselrichter mit dem dreiphasigen 0,4kV Netz und filtern die pulsierte Wechselrichterspannung. Grundsätzlich arbeitet der Wechselrichter in dem netzgesteuerten Betrieb, indem die Netzspannung gemessen und am Ausgang des Wechselrichters nachgeahmt wird. Simultan zu diesem netzgesteuerten Betrieb findet die Regelung des Grundschwingungsstromes des Wechselrichters statt. Diese Regelung ermöglicht die Steuerung des Energieflusses und der Blindleistungskompensation an der Intelligenten Ladesäule. Parallel zu dieser Grundschwingungsregelung wird die Regelung der Stromüberschwingung betrieben. Die Stromüberschwingung wird mit der gleitenden diskreten Fourier Transformation (DFT) gemessen und mit PI-Regler eingestellt. Für die Oberschwingungskompensation wird der Phasenwinkel der Spannungsüberschwingung gemessen und in die Stromüberschwingungsregelung eingeführt. Diese Synchronisation der Stromüberschwingung mit dem Phasenwinkel der Spannungsüberschwingung, bildet die Grundlage für die Kompensation einer Oberschwingung der Netzspannung. Um den Wirkungsgrad der Kompensation zu erhöhen muss der Phasenwinkel der Netzimpedanz bei der Synchronisierung einbezogen werden. Ein Regelalgorithmus für die Schätzung der Netzimpedanz, wird in einer zukünftigen Arbeit behandelt.

5 Prototyp der Intelligenten Ladesäule

Der Prototyp der Intelligenten Ladesäule wurde im Labor der Hochschule Mittweida aufgebaut. Die Hardwareverbindung der Komponenten wurde anhand der Diplomarbeit von Herrn Fabian Merkel realisiert. Die Umsetzung einer Regelung des Grundschwingungsstromes und die Anbindung der Intelligenten Ladesäule an das 0,4kV Netz waren Bestandteil der Masterarbeit von Herrn Nick Meyer. Die Regelung eines Oberschwingungsstromes und die gesteuerte Oberschwingungskompensation sind die Ziele dieser vorliegenden Arbeit. Diese wurden in dem vorangegangenen Kapitel beschrieben. Nachfolgend werden nun Hardware und Software der Ladesäule näher erörtert, sowie die Messergebnisse der gelungenen Oberschwingungskompensation präsentiert.

5.1 Hardware

Auf dem Übersichtplan sind die Hardwarekomponenten der Intelligenten Ladesäule dargestellt, deren Eigenschaften werden in diesem Kapitel erläutert.

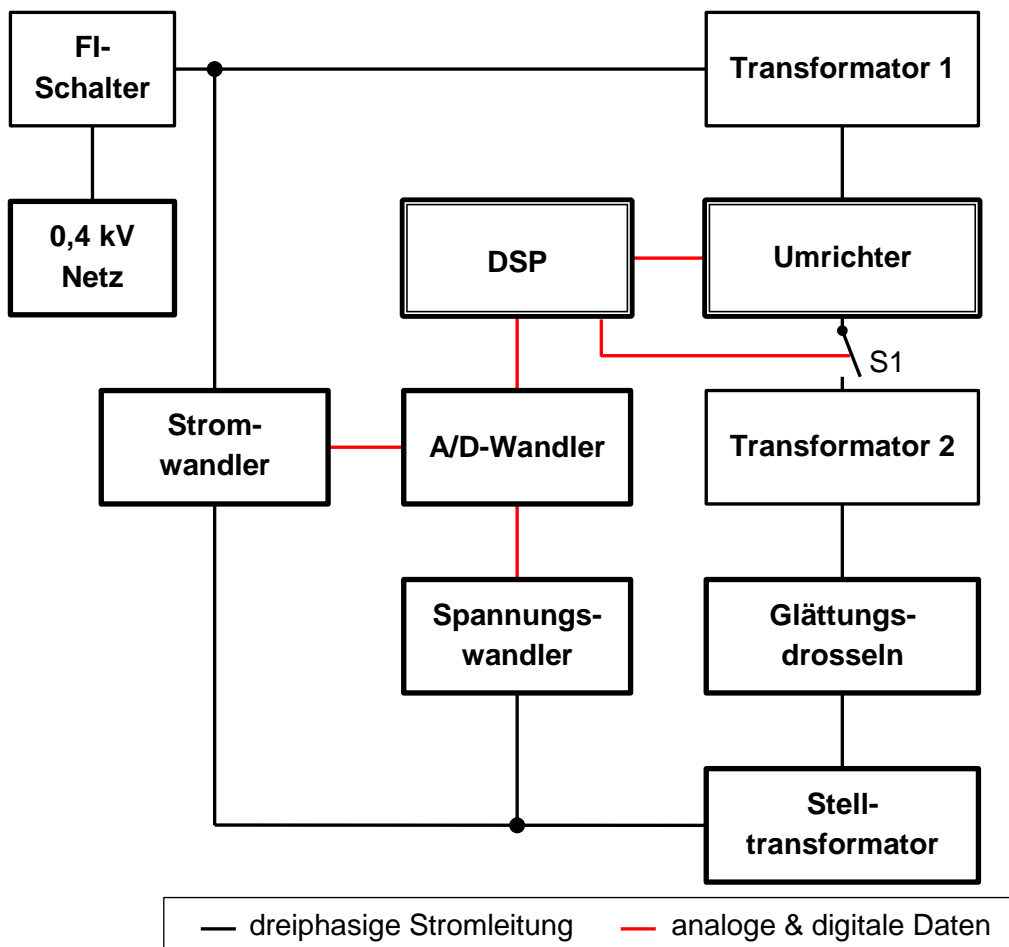


Abbildung 48: Übersichtplan des Prototyps der Intelligenten Ladesäule

5.1.1 Umrichter, Stelltransformator, Glättungsdrosseln

Der Umrichter des Herstellers SEW ist ein MOVDRIVE (Typenbezeichnung MDX61B0110-5A3-4-00) mit einer speziellen Anfertigung. Dabei wurde die Steuerung des Wechselrichters mit dem internen Mikroprozessor ausgeschaltet und die Steuerleitungen der IGBTs herausgeführt. Der Umrichter hat eine Nennleistung von 11kW und besteht aus einem dreiphasigen Gleichrichter mit Einschaltstrombegrenzung, einem 38µF Zwischenkreiskondensator, einem Choppertransistor, einem Chopperwiderstand (39Ω 1,2kW) und einem Zweipuls-Wechselrichter aus sechs IGBTs mit Überstromschutz. Die Steuerung des Choppertransistors findet im Umrichter statt, sie schaltet ab einer Zwischenkreisspannung von 820V ein und kann nicht beeinflusst werden.

Die Anbindung des Umrichters an das Netz ist wegen der FI-Schalter galvanisch getrennt. Auf der Gleichrichterseite befindet sich der Transformator 1 (Anlage 5) und auf der Wechselrichterseite der Transformator 2 (Anlage 3), beide Transformatoren haben ein Übersetzungsverhältnis von eins. Der Schutz S1 wird nach der vollständigen Ladung des Zwischenkreises zugeschaltet. Die drei Glättungsdrosseln (Anlage 4) haben eine Induktivität von 17,2mH und einen Widerstand von 1,4Ω.

Der Stelltransformator (Anlage 2) leistet das notwendige Spannungsgefälle zwischen der Gleich- und Wechselspannung des Wechselrichters und besteht aus drei einphasigen Ringstellsparttransformatoren mit einem Nennstrom von 5A. Das Übersetzungsverhältnis des Stelltransformators ist auf 0,7 eingestellt, die dreiphasige Netzspannung an der Wechselrichterseite beträgt 160V effektiv. Der Transformator 2, die Glättungsdrosseln und der Stelltransformator sind in Serie geschaltet und werden in einer Gesamtimpedanz \underline{Z}_d zusammengefasst.

$$\underline{Z}_d = 3,6 \Omega + j\omega \cdot 22,8 \text{ mH} \quad (5.66)$$

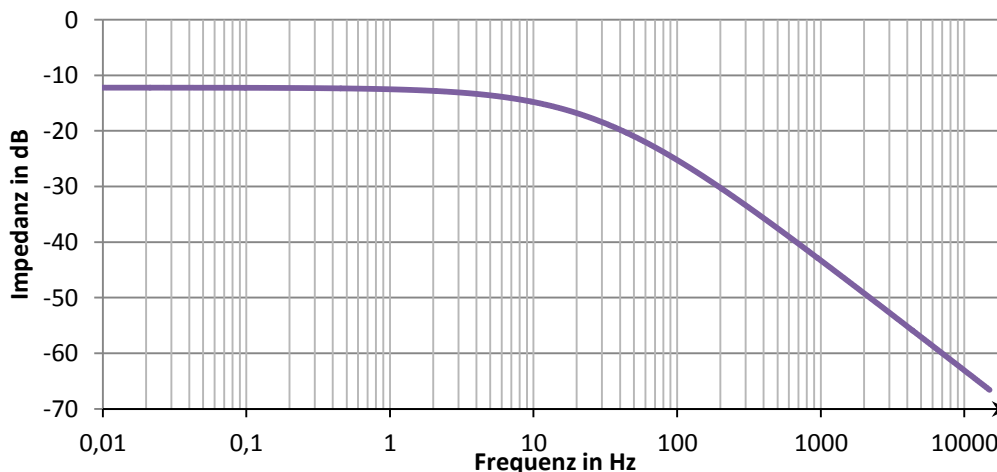


Abbildung 49: Amplitudenfrequenzgang der Impedanz \underline{Z}_d

Dieser Prototyp der Intelligenten Ladesäule ist an keine Elektrofahrzeugsbatterie angeschlossen. Die Gleichspannung einer solchen Batterie beträgt 300V und wird mit einem Hochsetzsteller auf 700V erhöht, der Gleichrichter des Umrichters bildet diese Gleichspannung nach. Die Ladung der Batterie wird nachgebildet, indem die Energie, die in dem Zwischenkreis fließt, durch den Chopperwiderstand in Wärme umgewandelt wird.

5.1.2 Strom- und Spannungswandler, A/D-Wandler

Die Messwerterfassung der dreiphasigen Netzspannung, des dreiphasigen Wechselrichterstromes sowie der einphasigen Zwischenkreisspannung erfolgt mit analogen Wandlern. Die drei Spannungswandler LEM CV 3-500 (Anlage 6) erfassen die dreiphasige Netzspannung mit einem Übersetzungsverhältnis von 50:1. Ein Spannungswandler LEM CV 3-1500 (Anlage 7) erfasst die Zwischenkreisspannung mit einem Übersetzungsverhältnis von 150:1. Das Tiefpassverhalten der Spannungswandler kann vernachlässigt werden, die Bandbreiten bei -1dB betragen 0Hz – 300kHz und 0Hz – 800kHz.

Der dreiphasige Wechselrichterstrom wird mit drei Stromwandlern LEM 200-S (Anlage 8) erfasst. Diese besitzen ein Übersetzungsverhältnis von 1000:1 und eine Bandbreite von 0Hz – 150kHz bei -1dB, das Tiefpassverhalten kann ebenfalls vernachlässigt werden. Für die Digitalisierung der vier Spannungen und der drei Ströme wird ein 16bit A/D-Wandler (Anlage 9) mit acht simultanen Sample & Hold bipolaren Eingängen (± 10 V) eingesetzt. Die Wertebereiche der analogen Strom- und Spannungsmesswerte am Eingang des A/D-Wandlers sind der Tabelle 9 zu entnehmen.

	Netzspannung	Zwischenkreisspannung	Wechselrichterstrom
physikalischer Bereich	-500V ... 500V	-1500V ... 1500V	-20A ... 20A
analoger Bereich	-10V ... 10V	-10V ... 10V	-200mA ... 200mA -1V ... 1V
digitaler Bereich	-32768 ... 32767	-32768 ... 32767	-3276 ... 3275
Auflösung	15,26mV/LSB	45,78mV/LSB	6,1mA/LSB

Tabelle 9: Wertebereich der Eingangsdaten des A/D-Wandlers

Das Übersetzungsverhältnis der Stromwandler wird durch 10-fache Wicklung der Stromleitung am Stromwandler auf 100:1 reduziert. Die Stromwandler liefern einen Strommesswert der durch einen Präzisionswiderstand von 5 Ω als Spannung gemessen werden kann.

Die Voraussetzung für die deterministische Diskretisierung der Zeit ist eine konstante Abtastfrequenz. Der A/D-Wandler ist mit dem Ausgang eines Timers direkt verbunden. Dieser Timer generiert das Wandlungsstartsignal mit einer konstanten Frequenz von 10kHz. Bei jeder Wandlungsanforderung werden alle acht Eingangsspannungen des A/D-Wandlers gleichzeitig in den Sample & Hold-Einheiten gespeichert und schließlich nacheinander digitalisiert.

Nach dem Abtasttheorem von Shannon muss die Abtastfrequenz f_a des A/D Wandlers mindestens doppelt so hoch wie die höchste Frequenzkomponente f_g des Messsignals sein.

$$f_a > 2 \cdot f_g \quad (5.67)$$

Frequenzanteile die höher als die Nyquist-Frequenz f_g sind, erzeugen in Folge von spektraler Überfaltung Messabweichungen im nützlichen Frequenzbereich des Signals. Das Beispiel des Alias-Effekts auf Abbildung 50 zeigt, dass die Unterabtastung des grauen Signals die niederfrequente rote Abweichung verursacht.

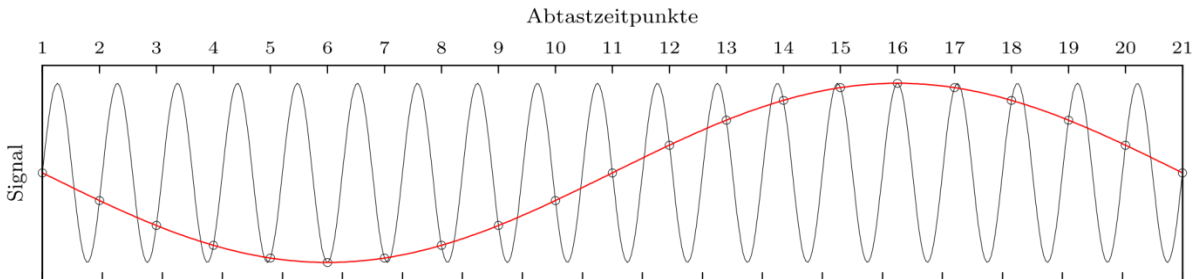


Abbildung 50: Alias-Effekt (Quelle: Wikipedia)

Am Eingang des A/D-Wandlers ist ein Tiefpassfilter 2. Ordnung, der bei einer Abtastfrequenz von 200kHz den Alias-Effekt um 35dB reduziert.

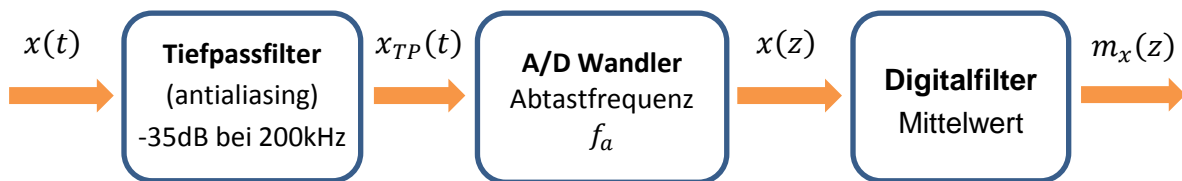


Abbildung 51: Filter-Komponente des A/D-Wandlers

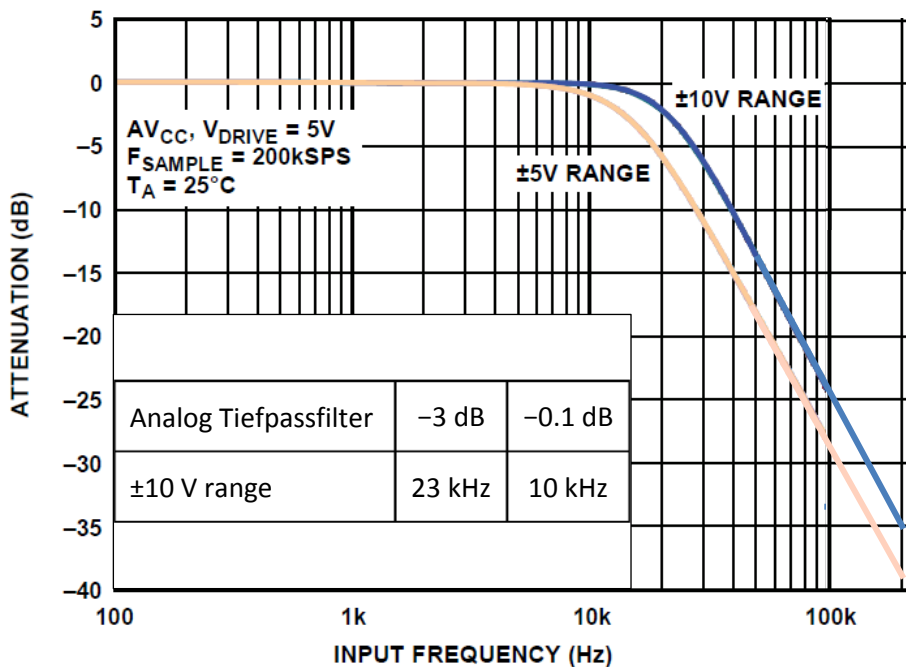


Abbildung 52: Antialiasing-Filter des A/D-Wandlers: Amplitudenfrequenzgang

Die Messung der 1. bis 40. Oberschwung (50 bis 2000Hz) wird von dem Tiefpassfilter nicht beeinflusst. Für die optimale Wirkung des Antialiasfilters muss der A/D-Wandler mit einer Abtastfrequenz von 200kHz betrieben werden. Für die Regelung des

Wechselrichters ist eine Abtastung der Ströme und Spannungen mit 10kHz jedoch ausreichend. Um die Datenmenge von 200kHz auf 10kHz zu reduzieren, besitzt der A/D-Wandler einen Digitalfilter, der einen Mittelwert der 200kHz abgetasteten Messwerte bildet. Das Zeitfenster dieses Mittelwertes ist durch das Überabtastungsverhältnis (oversampling ratio) definiert. Mit einem oversampling ratio von 16 ist die Abtastfrequenz am Ausgang des A/D-Wandlers auf 12,5kHz reduziert. Der Frequenzgang des Digitalfilters wurde bereits in Kapitel 1.2 beschrieben.

oversampling ratio		2	4	8	16	32	64
maximale Wandlungsfrequenz		100 kHz	50 kHz	25 kHz	12,5 kHz	6,25 kHz	3,125 kHz
Wandlungszeit	Min	7.87 μ s	16.05 μ s	33 μ s	66 μ s	133 μ s	257 μ s
	Max	9.1 μ s	18.8 μ s	39 μ s	78 μ s	158 μ s	315 μ s

Tabelle 10: Oversampling ratio des A/D-Wandlers

Aktuell wurde in dem Prototyp der Intelligenten Ladesäule eine Abtastfrequenz von 200kHz in Kombination mit dem Digitalfilter noch nicht implementiert. Die Abtastung der Ströme und Spannungen erfolgt mit einer Frequenz von 10kHz. Die Impedanz Z_d am Ausgang des Wechselrichters (die Netzimpedanz wird vernachlässigt) hat ein Tiefpassverhalten (Abbildung 49), die Pulsfrequenzen (≥ 10 kHz) der Wechselrichterausgangsspannungen werden mit mehr als 60dB gedämpft. Der Wechselrichterstrom enthält nach der Filterung jedoch einen Teil dieser Pulsfrequenzen. Der Zeitpunkt des Messanstoßes spielt eine entscheidende Rolle für die Eliminierung der Pulsfrequenzen und eine exakte Erfassung des Wechselrichterstromes. Der Messanstoß muss genau in der Mitte der Nullvektorzeit stattfinden, dieser korrespondiert aufgrund der Symmetrie des Transistorpulsformats immer mit dem Anfang einer Taktperiode.

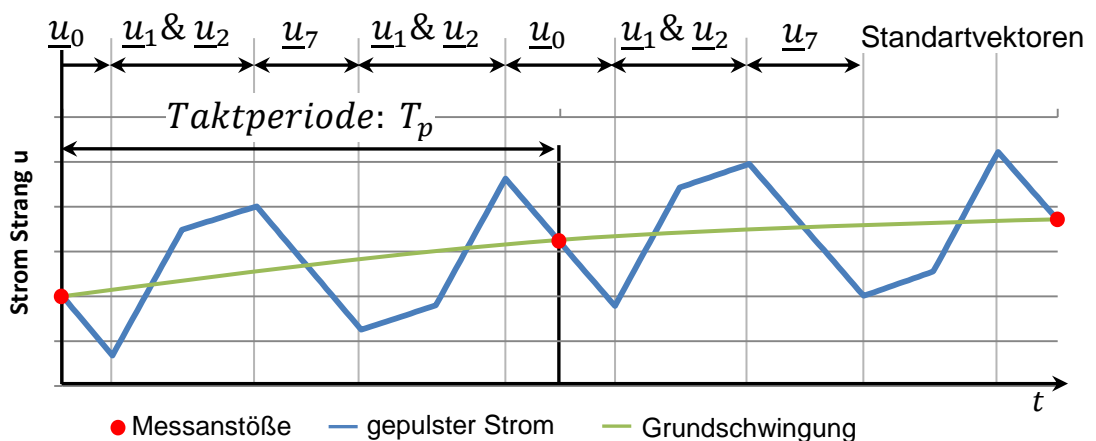


Abbildung 53: Zeitpunkt der Messanstoße zur Strommessung mittels A/D-Wandler [QUAN99]

Während eines Nullvektors sind die Wechselrichterausgangsspannungen null (Abbildung 30) und beeinflussen nicht die Messung der Netzspannung. Durch das Tiefpassverhalten der Netzimpedanz sind Störungen der Netzspannung, die eine Frequenz von 5kHz oder

höher besitzen von niedriger Amplitude und verursachen nur kleine Abweichungen (Alias-Effekt). Der Zwischenkreiskondensator sichert einen quasi konstanten Zwischenkreisspannungsmesswert innerhalb einer Taktperiode.

5.1.3 DSP

Die Softwareentwicklung wird auf den Digitalen Signal Prozessor (DSP) von Analog Devices Blackfin ADSP BF537 (Anlage 10) zugeschnitten. Es handelt sich dabei um einen 32-Bit-Prozessor mit einer Ganzzahlarithmetik. Der Prozessorkern arbeitet mit einer Taktfrequenz von 600MHz und beinhaltet zwei 16-Bit Multiplizierer, zwei 40-Bit Akkumulatoren, zwei 40-Bit arithmetische-logische-Einheiten (ALU), und einen 40-Bit Schieberegister. Jeder Multiplikationsakkumulator kann eine 16-Bit mal 16-Bit Multiplikation mit 40-Bit Akkumulation in einem Taktzyklus durchführen (Bsp.: $A' = A + B \cdot C$). Unter der Vielzahl von internen Peripheriegeräten des DSP befinden sich acht 32-Bit Timer mit externen Ausgängen für die Pulsweitenmodulation (PWM), 48 universelle Ein- und Ausgänge und 132Kbyte schnelle Arbeitsspeicher. Der Prozessor wird auf einem Development Kit betrieben und ist an 64Mbyte externe Arbeitsspeicher (SDRAM) angeschlossen. Die Portbelegung des DSP ist in Tabelle 11 beschrieben:

DSP	I/O	Peripherie	Funktion
Port G			
PG0 ... 15	I	A/D-Wandler	16-Bit Parallelport, Wandlungsdaten lesen
Port F			
PF2	O	Relais 2	Wechselrichter über Widerstand ans Netz schalten
PF3	O	Relais 1	Wechselrichter ans Netz schalten
PF4	O	Relais 3	Messung des Übersetzungsverhältnis des Stelltransformators
PF5	O	Wechselrichter	Freigabe
PF6	O	Wechselrichter	PWM Strang u
PF7	O	Wechselrichter	PWM Strang v
PF8	O	Wechselrichter	PWM Strang w
PF9	O	A/D-Wandler	Wandlungsstartsignal
PF10	O	A/D-Wandler	Parallelport aktivieren Wandlungsdaten eines Kanals selektieren
PF12	O	A/D-Wandler	Standby
PF13	I	A/D-Wandler	Wandler ist beschäftigt
PF14	O	A/D-Wandler	Spannungsbereich der Analogeingänge
PF15	O	A/D-Wandler	Reset

Tabelle 11: Portbelegung des DSP

5.2 Software

5.2.1 Zahlenformat und Variablentyp

Die Regelungsalgorithmen der Intelligenten Ladesäule werden in der Programmiersprache C implementiert. Die Standard Variablentypen, die in dem Programm angewendet werden, sind in der DSP mit folgendem Format definiert.

Variablentyp	Format	Betrag	Wertebereich
short	16 bits signed	15 bits	-32768 ... 32767
unsigned short	16 bits unsigned	16 bits	0 ... 65536
int	32 bits signed	31 bits	-2 147 483 648 ... 2 147 483 647
unsigned int	32 bits unsigned	32 bits	0 ... 4 294 967 296
long long	64 bits signed	63 bits	-2^{63} ... $2^{63} - 1$
unsigned long long	64 bits unsigned	64 bits	0 ... 2^{64}

Tabelle 12: angewendete Variablentypen in der DSP

Berechnungen mit Gleitkommazahlen sind in der DSP nicht optimiert und deshalb rechenzeitbelastend, Variablentypen float und double werden vermeiden. Der A/D-Wandler liefert für unterschiedliche physikalische Eingangsgrößen (Strom und Spannung) ganzzahlige Wandlungswerte in Typ short. Die Auflösungen dieser digitalen Ströme und Spannungen sind unterschiedlich (Tabelle 9 s.53) und erfordern eine dezimale Darstellung der physikalischen Eingangsgrößen. Es wird eine Festkommaarithmetik mit dem Qn.m Zahlenformat verwendet, dabei gibt n an wie viele Bits vor dem Komma und m wie viele Bits nach dem Komma verwendet werden. Die beiden Werte n und m berechnen sich dann wie folgt:

$$n = \text{ganzzahl}(\log_2(\text{Max}_{\text{physikalischer Bereich}})) \quad (5.68a)$$

$$m = \text{Betragsbits} - n \quad (5.68b)$$

Für die Zwischenkreisspannung können beispielsweise Spannungen von -1500V bis 1500V gemessen werden, somit ergibt sich ein Maximalwert von 1500V. Eingesetzt in obige Gleichungen (5.68) ergibt sich für den Variablentyp short das Zahlenformat Q11.4. Die maximalen und minimalen Werte einer short Variable im Zahlenformat Q11.4 sind

$$\text{Max}_{Q11.4} = 2^n - 2^{-m} = 2^{11} - 2^{-4} = 2047,9375 \quad (5.69a)$$

$$\text{Min}_{Q11.4} = -2^n = -2^{11} = -2048 \quad (5.69b)$$

und entsprechen nicht dem Spannungsbereich von -1500V bis 1500V.

Das vom A/D-Wandler gewandelte Ergebnis muss demzufolge mit einem Korrekturfaktor k_{kf} angepasst werden.

$$k_{kf} = \frac{Max_{\text{physikalischer Bereich}} \cdot 2^m}{Max_{A/D \text{ Wandler Bereich}}} \quad (5.70)$$

Die Messbereiche der Ströme und Spannungen in dem A/D-Wandler und in der DSP sind in Tabelle 13 zusammengefasst.

		Netzspannung	Zwischenkreis- spannung	Wechselrichter- strom
physikalischer Bereich		-500V ... 500V	-1500V ... 1500V	-20A ... 20A
A/D- Wandler	digitaler Bereich	-32768 ... 32767	-32768 ... 32767	-3276 ... 3275
	Auflösung	15,26mV/LSB	45,78mV/LSB	6,1mA/LSB
Korrekturfaktor		0,976592303	0,732444227	12,50687023
DSP	digitaler Bereich	-24001 ... 24000	-32001 ... 32000	-40961 ... 40960
	Variablentyp	short	short	short
	Zahlenformat	Q9.6	Q11.4	Q4.11

Tabelle 13: Zahlenformat und Wertebereich der Messwerte

Die Anzahl von Nachkommastellen m , bezeichnet die Auflösung der digitalen Messwerte. Die Umrechnung zwischen physikalischen und digitalen Messwerten ist:

$$x_{\text{digital}} = x_{\text{physikalisch}} \cdot 2^m \quad (5.71a)$$

$$\text{Bsp. Netzspannung: } 20800 = 325V \cdot 2^6 \quad (5.71b)$$

Für die Berechnung mit Festkommazahlen müssen folgende Regeln beachtet werden: Nur Variablen mit der gleichen Anzahl von Nachkommastellen m , können addiert oder subtrahiert werden, das Zahlenformat ist wie folgt anzupassen:

$$x_{Qn.m} \cdot 2^a = x_{Q(n-a).(m+a)} \quad (5.72)$$

Variablen mit unterschiedlichem Zahlenformat können miteinander multipliziert werden. Die meisten Variablen sind in dem 32-Bit integer typ deklariert. Es wird aber nur die Hälfte der 32-Bits genutzt. Die Multiplikationsakkumulatoren des Prozessorkerns können zwei 16-Bit Variablen in einem Taktzyklus multiplizieren und liefern ein 32-Bit Ergebnis. Das Zahlenformat des Ergebnisses wird nach der Multiplikation auf das Zahlenformat einer der beiden Operanden angepasst.

$$x_{Qa.b} \cdot y_{Qc.d} = z_{Q(a+c).(b+d)} \quad (5.73a)$$

$$z_{Q(a+c+d).b} = z_{Q(a+c).(b+d)} \cdot 2^{-b} \quad (5.73b)$$

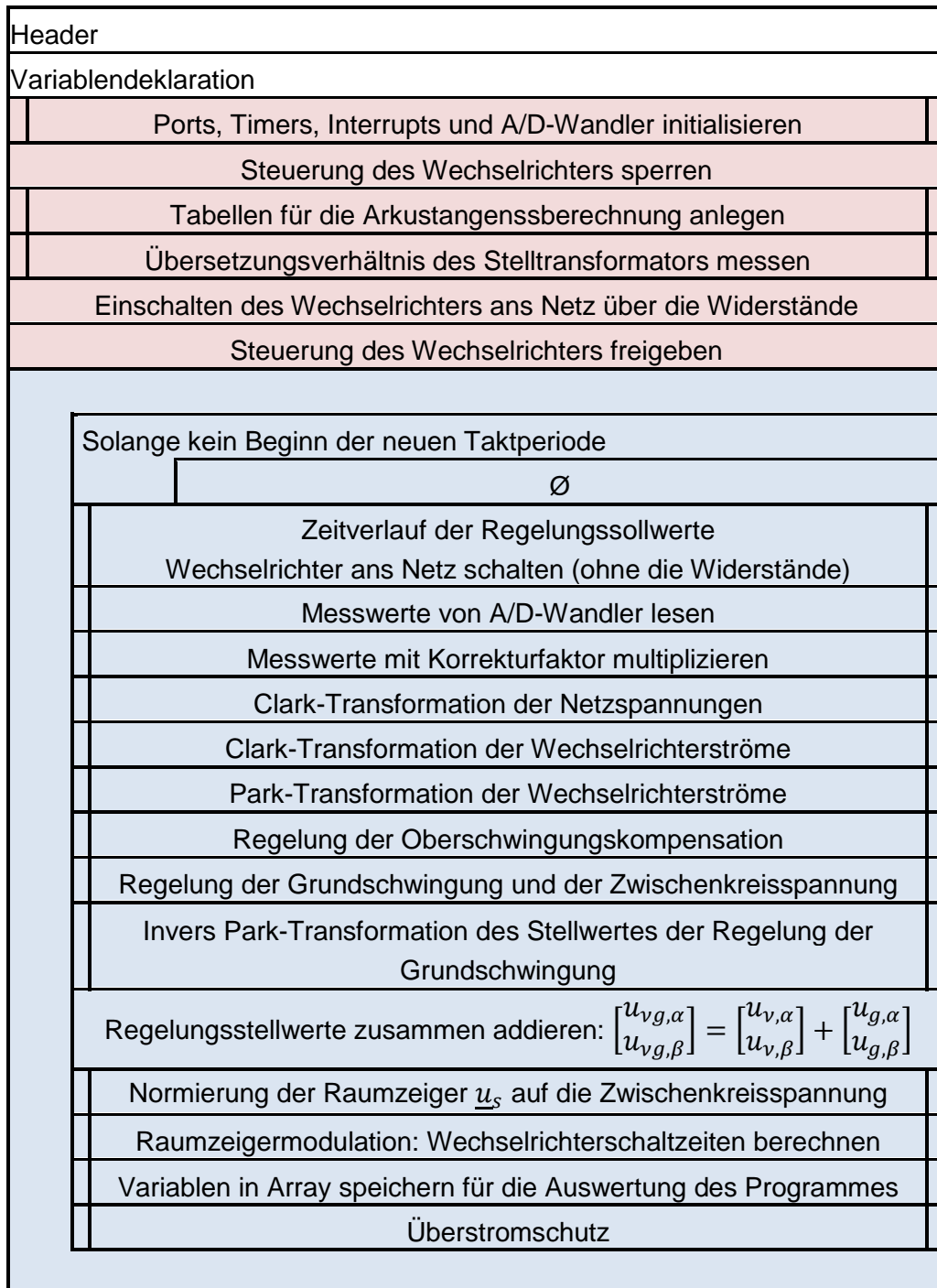
Besonders bei der Multiplikation muss die Gefahr eines Überlaufs des Ergebnisses sorgfältig überprüft werden. In dem Programm wird der Maximalwert jeder Variablen in einem Kommentar notiert. Die Verwendung von 64-Bit Variablen würde das Überlaufproblem lösen aber verlängert die Rechenzeit. Ein Mittelpunkt zwischen Auflösung und Überlauf muss gefunden werden, damit die Variablen mit einer ausreichenden Anzahl von Nachkommastellen dargestellt werden aber 16-Bit nicht überschreiten.

Andererseits ist der Überlauf für die Berechnung vom Phasenwinkel günstig. Der „Sprung“ des Phasenwinkels an der Stelle $2\pi = 0$ wird durch den Überlauf des Variablentyps unsigned short ($2^{16} = 0$) realisiert.

5.2.2 Hauptprogramm

Der Quellcode des gesamten Programms ist in Anlage 11 hinterlegt, der Programtablauf wird anhand von Struktogrammen erläutert.

Hauptprogramm:



Struktogramm 1: Hauptprogramm (main.c)

Das Hauptprogramm besteht aus Initialisierungsunterprogrammen (rosa), die nur beim Starten des Programms abgearbeitet werden, sowie aus einer Schleife (blau), die endlos durchgeführt wird. Diese endlose Schleife beinhaltet die Regelalgorithmen für die Steuerung des Energieflusses und die Blindleistungs- und Oberschwingungskompensation, die in Kapitel 4 erläutert wurden. Diese diskrete Regelung wird synchron mit der 100µs Taktperiode der PWM abgearbeitet. Der Timer0 generiert jede Halbtaktperiode (50µs) einen Softwareinterrupt und ein Hardware Wandlungsstartsignal an dem A/D-Wandler. Alle Timer werden mit 120MHz abgetastet. Sie werden während der Initialisierung synchron gestartet und arbeiten mit der gleichen Halbtaktperiode von 50µs. Der Zeitablauf der 100µs Hauptprogrammenschleife ist in Abbildung 54 dargestellt.

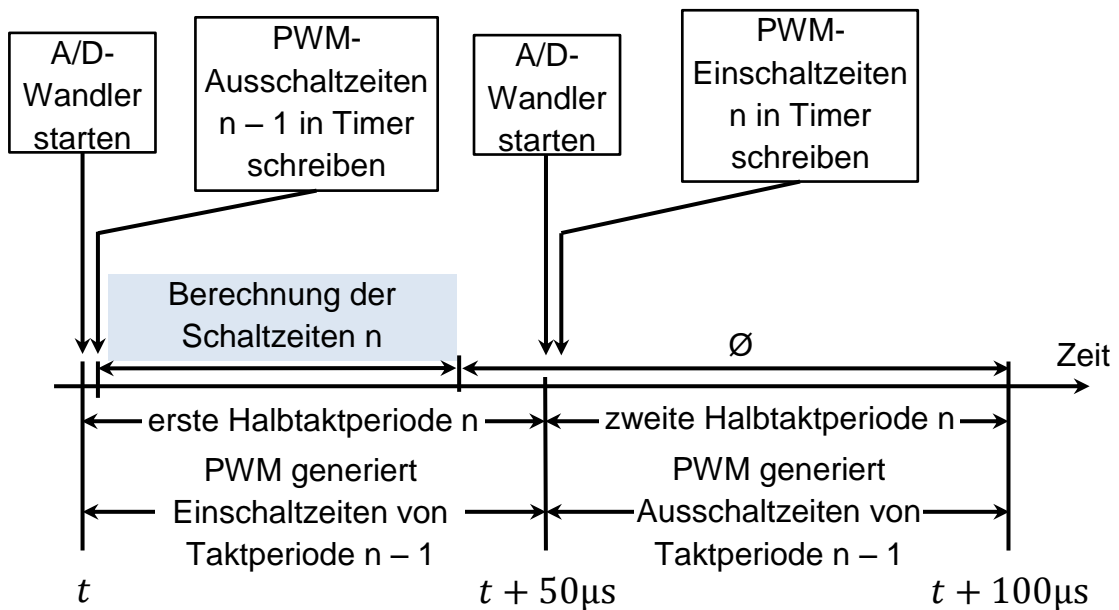


Abbildung 54: Zeitablauf einer 100µs Hauptprogrammenschleife

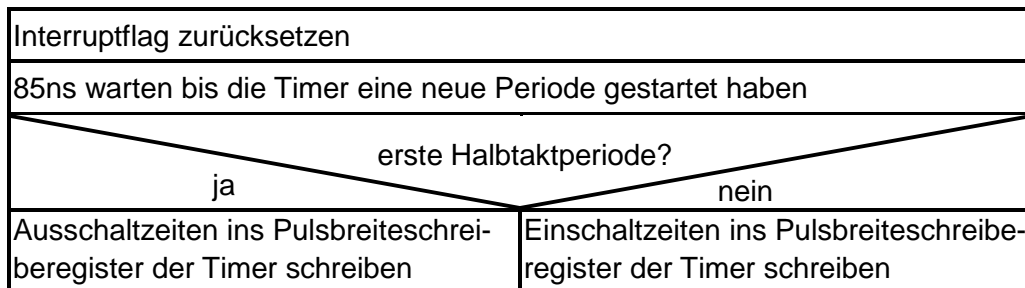
Der A/D-Wandler wird aller 50µs gestartet, die Wandlungsmesswerte werden jedoch nur am Anfang der 100µs Taktperiode gelesen.

Jeder Transistorzweig des Wechselrichters ist mit dem PWM-Ausgang eines Timers verbunden (Timer 1 bis 3). Anhand der Pulsbreite wird definiert, zu welchem Zeitpunkt innerhalb der 50µs, der PWM-Ausgang umgeschaltet wird. Diese Pulsbreite heißt Einschaltzeit, wenn der PWM von 0 auf 1 schaltet und Ausschaltzeit, wenn der PWM von 1 auf 0 schaltet. Die Ein- und Ausschaltzeiten werden aller 50µs abwechselnd in die Pulsbreiteschreiberegister der Timer 1 bis 3 geschrieben. Am Ende einer Halbtaktperiode lädt jeder Timer automatisch den Wert des Pulsbreiteschreiberegisters in das Pulsbreiteleseregister und beginnt wieder von vorn mit eins hoch zu zählen. Nur das Pulsbreiteleseregister ist für die Generierung der PWM zuständig. Das Pulsbreiteschreiberegister ist ein Puffer, in dem die Schaltzeiten für die nächste Halbtaktperiode der PWM geschrieben werden. Dies soll an einem Beispiel verdeutlicht werden:

Die Timer haben eine Ausschaltzeit generiert und kommen am das Ende der 50µs Periode, der Timer0 generiert einen Interrupt. In dem Interrupt Programm wird erst 85ns gewartet, bis die Timer ihr Pulsbreiteleseregister mit der Einschaltzeit des

Pulsbreiteschreiberegisters aktualisieren, bevor die neuen berechneten Ausschaltzeiten in die Pulsbreiteschreiberegister geschrieben werden.

Interrupt Behandlung für jede Halbtaktperiode (Timer0):



Struktogramm 2: Interrupt Behandlung

5.2.3 Optimierung der Rechenzeit

Die diskrete Regelung der Oberschwingungskompensation muss innerhalb einer 100µs Taktperiode abgearbeitet werden. Um dieses zu erreichen, werden trigonometrische Funktionen (sin, cos, arctan) mit Gleitkommazahlen, sowie die Division von Ganzzahlen durch Tabellen ersetzt. Eine gute Auflösung dieser Operationen wird durch lange Tabellen von einigen Megabyte erreicht. Dies ist möglich, da der DSP über eine 64Mbyte externe Speichereinheit verfügt.

Die Division wird durch Multiplikation ersetzt, die Divisionstabelle enthält den Kehrwert des Nenners, z.B. an der Adresse 10 der Divisionstabelle steht der Kehrwert 0,1.

Die Kehrwerte sind vom Variablentyp signed short und werden in das optimale Zahlenformat geschrieben, somit entstehen wenige Genauigkeitsverluste. Für 0,1 steht in der Tabelle die Zahl 26214 mit dem Zahlenformat Qn.18. Die Anzahl von Nachkommastellen m und der Kehrwert werden zusammen in eine integer Variable geschrieben.

32-Bit integer Variable der Divisionstabelle:



Abbildung 55: Speicherung der Kehrwerte in der Divisionstabelle

Die Umwandlung dieser integer Variable der Divisionstabelle in eine short Variable liefert den Kehrwert, und die Verschiebung dieser integer Variable um 16 Stellen nach rechts liefert die Nachkommastellen m. Die Division zweier 16-Bit Variablen ist folgende:

	real	signed short	Zahlenformat
Zähler	153,625	2458	Qn.4
Nenner	0,15625	10	Qn.6
Kehrwert	6,4	26214	Qn.12 ⇔ Qn.(18 – 6)
Ergebnis	983,2	64434012	Qn.16 ⇔ Qn.(12 + 4)

Tabelle 14: Bsp. Division durch Multiplikation ersetzen

Die trigonometrischen Funktionen Cosinus und Sinus werden durch eine Tabelle substituiert. Der Winkel (unsigned short) ist die Adresse des Ergebnisses (signed int, Q0.15), z.B. an der Adresse Pi ($\Leftrightarrow 2^{15}$) steht in der Cosinus-Tabelle eine -1 ($\Leftrightarrow -2^{15}$) und in der Sinus-Tabelle eine 0 ($\Leftrightarrow 0$).

Die trigonometrische Funktion Arkustangens kann in einer einzigen Tabelle nicht substituiert werden, 16 Tabellen mit einer geeigneten Auflösung werden benötigt.

Die Funktion Arkustangens wird im Bereich 0 bis $\pi/2$ durch Tabellen ersetzt und nur für die Berechnung des Arguments einer komplexen Zahl angewendet.

$$\arg\{a + jb\} = \arctan\left(\frac{b}{a}\right) \text{ für } a > 0 \text{ und } b > 0 \quad (5.74)$$

Der Quotient b/a wird mit der Divisionstabelle berechnet, in das Zahlenformat Q15.14 angepasst und dessen Betrag gebildet. Der Wertebereich der entstanden Variable $|b/a|$ ist $(0, 2^{29})$. In den Arkustangens-Tabellen sind die Ergebnisse der Arkustangenten an die Adresse $|b/a|$ geordnet, die Tabellen sind wie folgt verteilt:

Tabellenummer	Anzahl von Werte	$ b/a $ Auflösung	$ b/a $ Startwert
0	2^{15}	2^0	0
1	2^{14}	2^1	2^{15}
2	2^{13}	2^3	2^{16}
3	2^{12}	2^5	2^{17}
4	2^{11}	2^7	2^{18}
5	2^{10}	2^9	2^{19}
6	2^9	2^{11}	2^{20}
7	2^8	2^{13}	2^{21}
8	2^7	2^{15}	2^{22}
9	2^6	2^{17}	2^{23}
10	2^5	2^{19}	2^{24}
11	2^4	2^{21}	2^{25}
12	2^3	2^{23}	2^{26}
13	2^2	2^{25}	2^{27}
14	2^1	2^{27}	2^{28}
15	2^0	2^{29}	2^{29}

Tabelle 15: Substitution der Funktion Arkustangens durch Tabellen

Die Tabellen werden anhand des Startwerts ausgewählt, z.B. die Variable $|b/a|$ ist gleich 45126 bedeutet das Ergebnis ist in Tabelle 1. Die Adresse des Ergebnisses in der ausgewählte Tabelle ist wie folgt zu berechnen: Adresse = $(|b/a| - \text{Startwert}) / \text{Auflösung}$, was im Beispiel dem Wert $(45126 - 2^{15}) / 2^1 = 6179$ entspricht. Die Aufteilung nach Logarithmus zur Basis zwei des Startwertes und der Auflösung führt zu einer schnellen Berechnung der Adresse. Die Funktion Logarithmus zur Basis zwei für Ganzzahlen wird auch mit einer Tabelle substituiert und für die Auswahl der Arkustangens-Tabelle eingesetzt. Der Adressenbereich der Log2-Tabelle ist 0 bis 2^{15} , die Variable $|b/a|$ wird dann vorerst durch 2^{14} dividiert, was einer Verschiebung um 14 Stellen nach rechts entspricht. Die Nummer der Arkustangens-Tabelle wird dann an der Adresse $|b/a|/2^{14}$ in

der Log2-Tabelle abgelesen. Zum Schluss wird eine Korrektur des Winkels die Funktion Arkustangens auf den Bereich 0 bis 2π erweitern.

$a \geq 0, b \geq 0$	Winkel
$a < 0, b \geq 0$	2π - Winkel
$a < 0, b < 0$	π + Winkel
$a \geq 0, b < 0$	π - Winkel

Tabelle 16: Erweiterung des Arkustangens auf 2π

Die Verbesserung der Rechenzeit für jede Operation und die gesamte Regelung ist somit gelungen.

Operation	Rechenzeit	Verbesserung
Division (mit Variablentyp integer)	708ns	- 639ns/Division, 10 mal schneller
Division mit der Divisionstabelle	69ns	
Arkustangens (mit Variablentyp float)	1755ns	- 1507ns/Arkustangens, 7 mal schneller
Arkustangens mit Arkustangens-Tabellen	248ns	
bestehende diskrete Regelung der Blindleistung der vorherigen Masterarbeit	35 μ s	aufwendigere Regelungsalgorithmen und noch 75 μ s Pufferzeit pro Taktperiode
diskrete Regelung der Blindleistungs- und Oberschwingungskompensation	25 μ s	

Tabelle 17: Ergebnisse der Rechenzeitoptimierung

5.2.4 Initialisierungsunterprogramm

Der Port G und Port F des DSPs werden konfiguriert, der A/D-Wandler initialisiert, die Timer und das Interrupt werden ebenfalls konfiguriert und abschließend die Timer gestartet.

Ports, Timers, Interrupts und A/D-Wandler initialisieren:

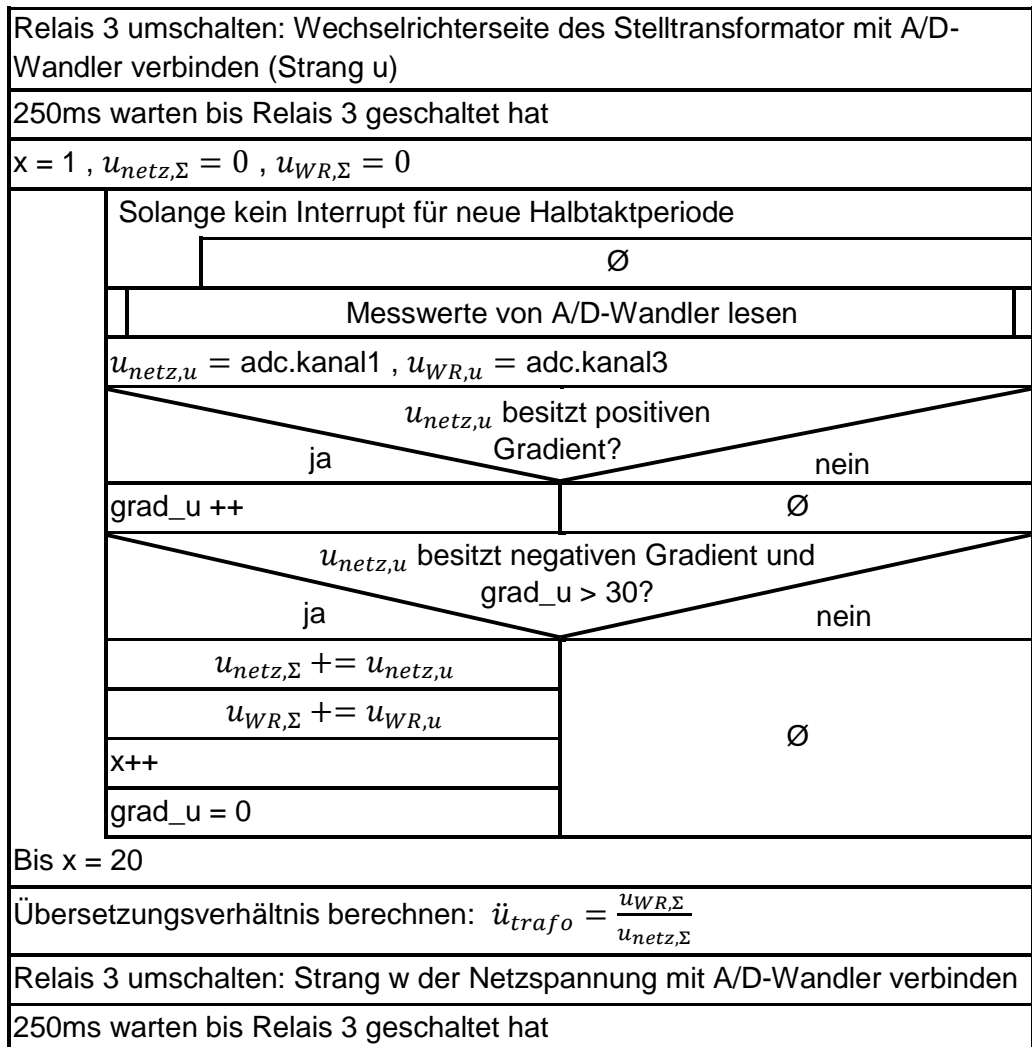
Port G als universelle Eingänge konfigurieren
Port F konfigurieren: PF 9, 8, 7, 6: als Timerausgänge aktivieren (Timer 0 bis 3) PF 2, 3, 4, 5, 10, 11, 12, 14: universelle Ausgänge PF 13: universeller Eingang
A/D-Wandler einschalten 100ms warten bis A/D-Wandler initialisiert ist Eingangsspannungsbereich des A/D-Wandlers einstellen: ± 10 V
Timer0 bis 3 konfigurieren: PWM Modus aktivieren PWM-Periode = 50 μ s Timer0: Interrupt aktivieren
Interrupt initialisieren: assign core IDs to Interrupts

assign Interrupt Service Routine to Interrupt vectors Interrupt Timer0 einschalten
Timer starten

Struktogramm 3: Initialisierung des DSP und des A/D-Wandlers

Die Tabellen für den Logarithmus zur Basis zwei, der Sinusfunktion, der Cosinusfunktion und der Division sind in Headerdateien bereits geschrieben. Die Tabellen für die Funktion Arkustangens werden mit einem Initialisierungsunterprogramm berechnet und gespeichert. Die Berechnung erfolgt mit Variablentyp float und folgt der in Tabelle 15 angegebenen Bereiche und Auflösungen. Daraufhin wird das Übersetzungsverhältnis des Stelltransformators ermittelt. An 20 sukzessiven Scheitelpunkten der sinusförmigen Netzspannung werden die Primär- und Sekundärspannung des Stelltransformators gemessen. Eine Summe jeder der zwei Messreihen wird gebildet und das Übersetzungsverhältnis berechnet.

Übersetzungsverhältnis des Stelltransformators messen:



Struktogramm 4: Messung des Übersetzungsverhältnisses des Stelltransformators

5.2.5 Unterprogramme der diskreten Regelung

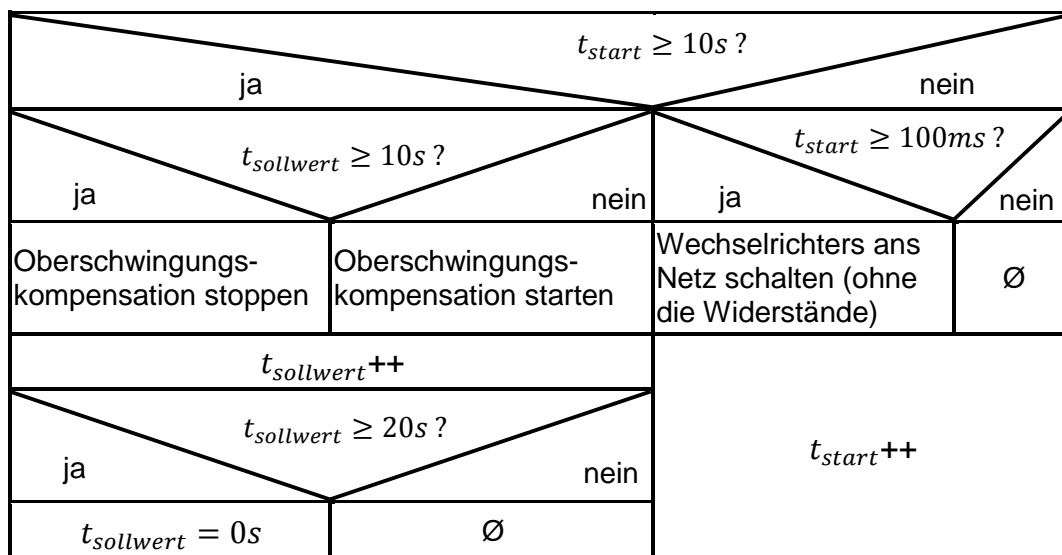
In diesem Abschnitt wird die endlose Schleife des Hauptprogramms beschrieben. Sie wird nach jeder zweiten Interrupt Behandlung (Struktogramm 2) gestartet und wird aller $100\mu\text{s}$ periodisch durchgeführt. Die Wiederholungszahl dieser Schleife wird hoch gezählt und bildet die diskrete Zeiteinheit des Programms. Die zeitliche Steuerung des Programms erfolgt an verschiedenen Stellen des Programms und kann deshalb durch ein Struktogramm nicht dargestellt werden, der Programmablauf ist in der folgenden Tabelle zusammengefasst:

Zeit t	Programmablauf und Aktivität des Wechselrichters
-	Initialisierung, Wechselrichter ist gesperrt
0	Wechselrichter ans Netz schalten mit Widerständen Freigabe des Wechselrichters netzgesteuerter Betrieb des Wechselrichter ist aktiv Regelungsstellwerte werden berechnet aber bleiben inaktiv
0,1s	Wechselrichter ans Netz schalten ohne Widerstände
0,5s	Regelungsstellwert des Grundschwingungsstromes ist aktiv
1s	Regelungsstellwert des Oberschwingungsstromes ist aktiv Sollwert des Oberschwingungsstromes ist null
10s	Sollwert des Oberschwingungsstromes = Oberschwingungskompensation

Tabelle 18: Programmablauf und Aktivität des Wechselrichters

In dem ersten Unterprogramm werden die Zeitverläufe der Regelungssollwerte und die Einschaltzeiten des Wechselrichters ans Netz gesteuert.

Zeitverlauf der Regelungssollwerte und Wechselrichter ans Netz schalten (ohne die Widerstände):



Struktogramm 5: Regelungssollwerte

Nach der Festlegung der Sollwerte werden die Istwerte der Netzspannungen und Wechselrichterströme im A/D-Wandler gelesen und in einer Struktur gespeichert.

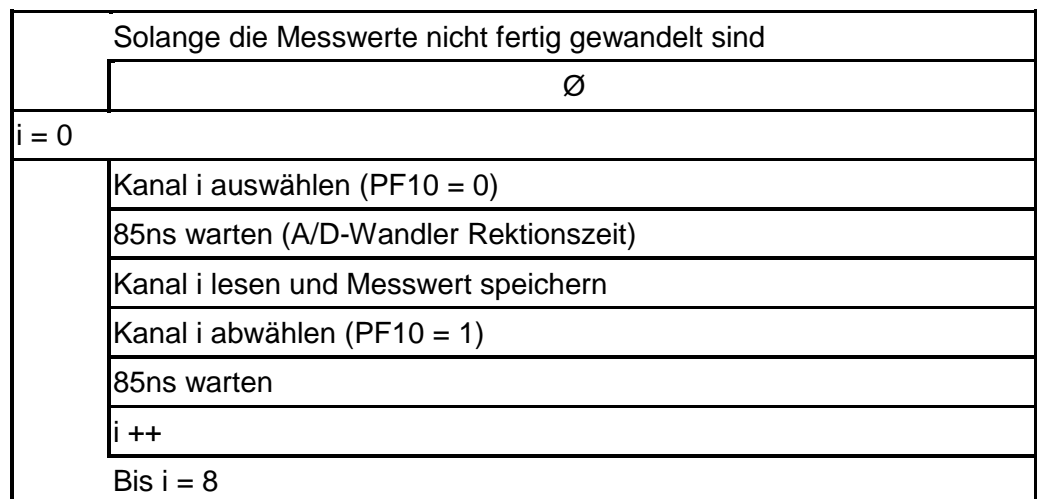
```

struct
{
    short kanal1;      /*Spannung U*/
    short kanal2;      /*Spannung V*/
    short kanal3;      /*Spannung W*/
    short kanal4;      /*noch Frei*/
    short kanal5;      /*Spannung ZK*/
    short kanal6;      /*Strom W*/
    short kanal7;      /*Strom V*/
    short kanal8;      /*Strom U*/
    short update;      /* auf 1 gesetzt nach neuer Wandlung*/
} stADCDaten;

```

Der Ausgang des Timer0 löst das Wandlungsstartsignal aus. Durch Polling wird das Ende der Wandlung getestet, die Wandlungszeit beträgt 4µs. Die Wandlungswerte jeder Kanäle werden denn auf den Parallelport des A/D-Wandlers nacheinander gelesen.

Messwerte von A/D-Wandler lesen:



Struktogramm 6: Messwerte von A/D-Wandler lesen

Die Messwerte werden normiert, wie es in Kapitel 5.2.1 (s. 57) beschreiben wurde, und in getrennte Strukturen gespeichert.

```

_stDatenInDreiAchsen      stDreiPhasenNetzSpannung;
_stDatenInDreiAchsen      stDreiPhasenNetzStrom;
_stSystemParameter        stSystemParameter;

typedef struct _stDatenInDreiAchsen
{
    int u_phas;           /* Strom oder Spannung Phase-U */
    int v_phas;           /* Strom oder Spannung Phase-V */
    int w_phas;           /* Strom oder Spannung Phase-W */
} _stDatenInDreiAchsen;

typedef struct _stSystemParameter
{
    int uebersetzungsverhaeltniss;
    int netzzeiger;
    int trafozeiger;
}

```

```

int zwischenkreisspannung;      /* Zwischenkreisspannung */
int teststrom_RMS_U;
int teststrom_RMS_V;
int teststrom_RMS_W;
int regler_alpha;
int regler_beta;
int rzm_alpha;
int rzm_beta;
} _stSystemParameter;

```

Messwerte mit Korrekturfaktor multiplizieren:

$\begin{bmatrix} u_{netz,u} \\ u_{netz,v} \\ u_{netz,w} \end{bmatrix}_{norm} = \begin{bmatrix} u_{netz,u} \\ u_{netz,v} \\ u_{netz,w} \end{bmatrix} \cdot 0,9766$
$u_{ZK,norm} = u_{ZK} \cdot 0,7324$
$\begin{bmatrix} i_{WR,u} \\ i_{WR,v} \\ i_{WR,w} \end{bmatrix}_{norm} = \begin{bmatrix} i_{WR,u} \\ i_{WR,v} \\ i_{WR,w} \end{bmatrix} \cdot 12,507$

Struktogramm 7: Normierung der Messwerte

Die Raumzeiger der Netzspannung und der Wechselrichterströme werden mit der Clark-Transformation gebildet und in zwei Strukturen gespeichert.

```

_stDatenInFesteAchsen      stAlphaBetaNetzSpannung;
_stDatenInFesteAchsen      stAlphaBetaNetzStrom;

typedef struct _stDatenInFesteAchsen
{
  int alpha; /* Realteil des rotierenden Raumzeigers */
  int beta;  /* Imaginärteil des rotierenden Raumzeigers */
  int theta; /* Phasenwinkel des rotierenden Raumzeigers */
} _stDatenInFesteAchsen;

```

Clark-Transformation der Netzspannungen:

$\begin{bmatrix} u_{netz,\alpha} \\ u_{netz,\beta} \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \end{bmatrix} \cdot \begin{bmatrix} u_{netz,u} \\ u_{netz,v} \\ u_{netz,w} \end{bmatrix}_{norm}$
Phasenwinkel θ_u mit der Funktion Arkustangens berechnen

Struktogramm 8: Clark-Transformation der Netzspannungen

Clark-Transformation der Wechselrichterströme:

$\begin{bmatrix} i_{WR,\alpha} \\ i_{WR,\beta} \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \end{bmatrix} \cdot \begin{bmatrix} i_{WR,u} \\ i_{WR,v} \\ i_{WR,w} \end{bmatrix}_{norm}$

Struktogramm 9: Clark-Transformation der Wechselrichterströme

Der Raumzeiger der Wechselrichterströme wird in das d,q rotierende Koordinatensystem transformiert und in einer Struktur gespeichert.

```

_stDatenInRotierendeAchsen  stParkKomponenteStrom

typedef struct _stDatenInRotierendeAchsen
{
    int d_part;      /* Realteil des ruhenden Raumzeigers */
    int q_part;      /* Imaginärteil des ruhenden Raumzeigers*/
} _stDatenInRotierendeAchsen;

```

Park-Transformation der Wechselrichterströme:

$$\begin{bmatrix} i_{WR,d} \\ i_{WR,q} \end{bmatrix} = \begin{bmatrix} \cos \theta_u & \sin \theta_u \\ -\sin \theta_u & \cos \theta_u \end{bmatrix} \cdot \begin{bmatrix} i_{WR,\alpha} \\ i_{WR,\beta} \end{bmatrix}$$

Struktogramm 10: Park-Transformation der Wechselrichterströme

Die Bearbeitung der Istwerte für die Regelung des Grundschwingungsstromes ist fertig. Es wird jetzt die Regelung der Oberschwingungskompensation behandelt. Zuerst wird der Zeitverlauf jedes normierten Messwertes in einem Array gespeichert, dann erfolgen die Regelung der Stromüberschwingung und zuletzt die Auswertung der Spannungsüberschwingung.

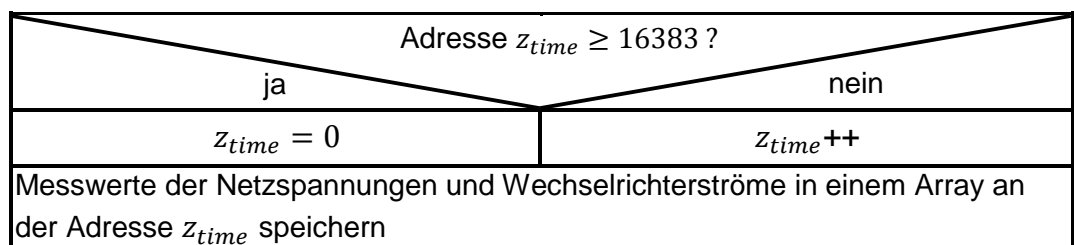
Regelung der Oberschwingungskompensation:

	Speicherung der Strom und Spannungsmesswerte in einem Array	
	Auswertung und Regelung der Stromüberschwingung (Leiter u)	
	Auswertung der Oberschwingungen der Netzspannung (Leiter u,v,w)	

Struktogramm 11: Regelung der Oberschwingungskompensation

Die Speicheradresse z_{time} wird jedes Mal inkrementiert und zeigt auf den nächsten Speicherplatz des Arrays. Jeder Messwert wird einem 2^{14} Groß Array zugeordnet, was einer Speicherzeit von 1,6384 Sekunden entspricht.

Speicherung der Strom und Spannungsmesswerte in einem Array:



Struktogramm 12: Speicherung des Zeitverlaufs der Messwerte

Die Auswertung des Oberschwingungsstromes erfolgt mit einem 50Hz DFT. Die Zeit innerhalb des 20ms Zeitfensters der DFT wird mit der Variablen $z_{time50Hz}$ gezählt. Aus dieser diskreten Zeitvariablen wird der Phasenwinkel eines $\nu \cdot 50\text{Hz}$ rotierenden Zeigers

gebildet und anschließend der Real- und Imaginärteil dieses Zeigers berechnet. Anhand dieses Zeigers und der gespeicherten Messwerte wird die v . Stromoberschwingung des Leiters u mit der gleitenden DFT ausgewertet. Je nach der symmetrischen Komponente der Stromoberschwingung wird die Rotationsrichtung des entstandenen Zeigers angepasst und eine PI-Regelung der Stromoberschwingung durchgeführt. Der Stellwert des PI-Reglers wird schließlich anhand der Park-Transformation in das α, β Koordinatensystem rücktransformiert.

Auswertung und Regelung der Stromoberschwingung:

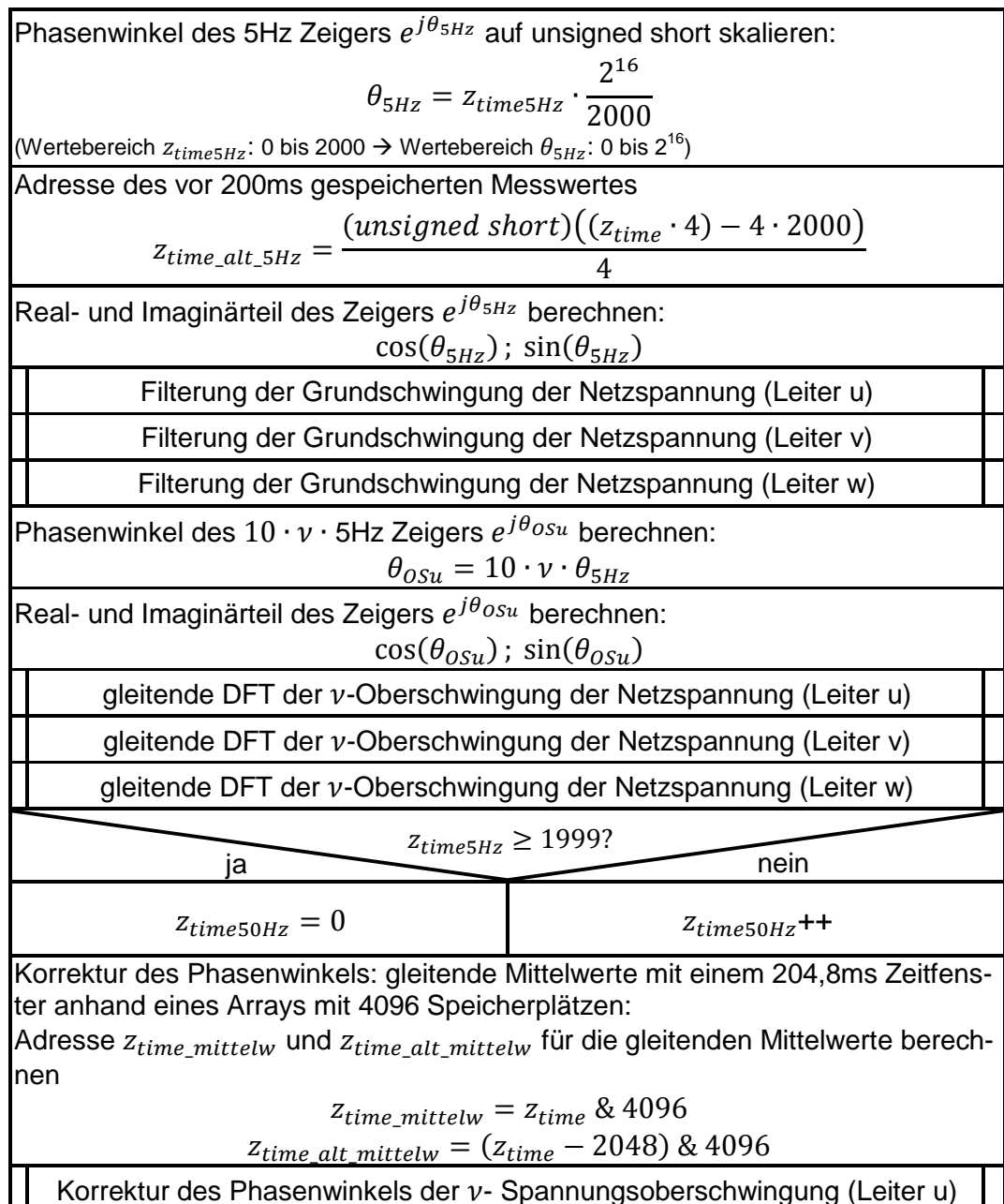
Phasenwinkel des 50Hz Zeigers $e^{j\theta_{50Hz}}$ auf unsigned short skalieren: $\theta_{50Hz} = z_{time50Hz} \cdot \frac{2^{16}}{200}$ (Wertebereich $z_{time50Hz}$: 0 bis 200 → Wertebereich θ_{50Hz} : 0 bis 2^{16})	
Adresse des vor 20ms gespeicherten Messwertes $z_{time_alt_50Hz} = \frac{(unsigned\ short)((z_{time} \cdot 4) - 4 \cdot 200)}{4}$	
Phasenwinkel des $v \cdot 50Hz$ Zeigers $e^{j\theta_{0Si}}$ berechnen: $\theta_{0Si} = v \cdot \theta_{50Hz}$	
Real- und Imaginärteil des Zeigers $e^{j\theta_{0Si}}$ berechnen: $\cos(\theta_{0Si}) ; \sin(\theta_{0Si})$	
Gleitende DFT der v -Oberschwingung des Stromes $i_{WR,u}$ berechnen: $i_{WRv,d\Sigma} += i_{WR,u}(z_{time}) \cdot \sin(\theta_{0Si}) - i_{WR,u}(z_{time_alt_50Hz}) \cdot \sin(\theta_{0Si})$ $i_{WRv,q\Sigma} += i_{WR,u}(z_{time}) \cdot \cos(\theta_{0Si}) - i_{WR,u}(z_{time_alt_50Hz}) \cdot \cos(\theta_{0Si})$	
Mitsystem Oberschwingung?	
ja	nein
Istwert der Stromoberschwingung auf 20ms skalieren: $i_{WRv,d} = \frac{i_{WRv,d\Sigma}}{200} ; i_{WRv,q} = \frac{i_{WRv,q\Sigma}}{200}$	Istwert der Stromoberschwingung auf 20ms skalieren: $i_{WRv,d} = \frac{i_{WRv,d\Sigma}}{200} ; i_{WRv,q} = -\frac{i_{WRv,q\Sigma}}{200}$
PI-Regelung der Stromoberschwingung	PI-Regelung der Stromoberschwingung
Stellwert der Stromoberschwingung in α, β Koordinatensystem transformieren: $\begin{bmatrix} u_{v,\alpha} \\ u_{v,\beta} \end{bmatrix} = \begin{bmatrix} \cos \theta_{0Si} & -\sin \theta_{0Si} \\ \sin \theta_{0Si} & \cos \theta_{0Si} \end{bmatrix} \cdot \begin{bmatrix} u_{v,d} \\ u_{v,q} \end{bmatrix}$	Stellwert der Stromoberschwingung in α, β Koordinatensystem transformieren: $\begin{bmatrix} u_{v,\alpha} \\ u_{v,\beta} \end{bmatrix} = \begin{bmatrix} \cos \theta_{0Si} & \sin \theta_{0Si} \\ -\sin \theta_{0Si} & \cos \theta_{0Si} \end{bmatrix} \cdot \begin{bmatrix} u_{v,d} \\ u_{v,q} \end{bmatrix}$
$z_{time50Hz} \geq 199?$	
ja	nein
$z_{time50Hz} = 0$	$z_{time50Hz} ++$

Struktogramm 13: Regelung der Stromoberschwingung

Die Oberschwingungskompensation basiert auf der Messung der Oberschwingung der Netzspannung des Leiters u . Die Kompensationswirkung wird anhand der Oberschwingungsmesswerte der drei Leitern u, v, w ausgewertet. Die Auswertung der Oberschwingungsspannung erfolgt mit einer 5Hz DFT. Die Zeit innerhalb des 200ms

Zeitfensters der DFT wird mit der Variablen $z_{time5Hz}$ gezählt. Aus dieser diskreten Zeitvariablen wird der Phasenwinkel eines $\nu \cdot 10 \cdot 5\text{Hz}$ rotierenden Zeigers gebildet und anschließend der Real- und Imaginärteil dieses Zeigers berechnet. Vor der Auswertung der Oberschwingung mit der gleitenden DFT wird die Grundschiwingung der Netzspannung gefiltert. Der Phasenwinkel der Spannungsüberschwingung des Leiters u wird einer Totzeitkorrektur angepasst.

Auswertung der Oberschwingungen der Netzspannung:



Struktogramm 14: Auswertung der Oberschwingungen der Netzspannung

Der 50Hz Filter für die Grundschiwingung der Netzspannung ist, wie in Kapitel 4.4.1 (s.43) aufgebaut und wird für alle drei Netzspannungen u,v,w durchgeführt.

Filterung der Grundschiwingung der Netzspannung:

gleitende 5Hz DFT der Grundschiwingung der Netzspannung berechnen: $u_{netz,g,d\Sigma} += u_{netz}(z_{time}) \cdot \sin(\theta_{50Hz}) - u_{netz}(z_{time_alt_5Hz}) \cdot \sin(\theta_{50Hz})$ $u_{netz,g,q\Sigma} += u_{netz}(z_{time}) \cdot \cos(\theta_{50Hz}) - u_{netz}(z_{time_alt_5Hz}) \cdot \cos(\theta_{50Hz})$
Grundschiwingung auf 200ms skalieren: $u_{netz,g,d} = \frac{u_{netz,g,d\Sigma}}{2000} ; u_{netz,g,q} = \frac{u_{netz,g,q\Sigma}}{2000}$
Grundschiwingung nachbilden $u_{netz,g} = u_{netz,g,d} \cdot \sin(\theta_{50Hz}) + u_{netz,g,q} \cdot \cos(\theta_{50Hz})$
Grundschiwingung von dem Netzspannungsmesswert abziehen $u_{netz,OS} = u_{netz} - u_{netz,g}$

Struktogramm 15: Filterung der Grundschiwingung der Netzspannung

Die gleitende DFT liefert den Real- und Imaginärteil des Oberschwingungszeigers. Für die Auswertung der Oberschwingungskompensation wird der Betrag in einem passenden Zahlenformat für die kleine Amplitude der Spannungsoberschwingung gebildet, der Phasenwinkel der Oberschwingungsspannung des Leiters u wird anhand der Arkustangens-Tabellen berechnet.

gleitende DFT der v-Oberschwingung der Netzspannung:

gleitende 5Hz DFT der v-Oberschwingung der Netzspannung berechnen: $u_{v,d\Sigma} += u_{netz,OS}(z_{time}) \cdot \sin(\theta_{OSu}) - u_{netz,OS}(z_{time_alt_5Hz}) \cdot \sin(\theta_{OSu})$ $u_{v,q\Sigma} += u_{netz,OS}(z_{time}) \cdot \cos(\theta_{OSu}) - u_{netz,OS}(z_{time_alt_5Hz}) \cdot \cos(\theta_{OSu})$
v-Oberschwingung auf 200ms und Zahlenformat Q3.12 skalieren: $u_{v,d} = u_{v,d\Sigma} \cdot \frac{2^6}{2000} ; u_{v,q} = u_{v,q\Sigma} \cdot \frac{2^6}{2000}$
Amplitude der v-Oberschwingung berechnen $\hat{u}_v = \sqrt{u_{v,d}^2 + u_{v,q}^2}$
Phasenwinkel der v-Oberschwingung mit der Funktion Arcustangens

Struktogramm 16: gleitende DFT der v-Oberschwingung der Netzspannung

Die Totzeitkorrektur und Filterung des Phasenwinkels erfolgt wie in Kapitel 4.4.3 (s.46). Der entstandene korrigierte Phasenwinkel wird als Sollwert an die Regelung des Oberschwingungsstromes weitergegeben (Struktogramm 5).

Korrektur des Phasenwinkels der v. Oberschwingungsspannung:

Kreisfrequenz der v-Oberschwingung der Netzspannung berechnen: $\omega = \varphi(z_{time}) - \varphi(z_{time} - 1)$
gleitenden Mittelwert der Kreisfrequenz berechnen: $m_{\omega\Sigma} += \omega(z_{time_mittelw}) - \omega(z_{time_alt_mittelw})$

gemittelte Kreisfrequenz skalieren und integrieren → gemittelter Phasenwinkel
$m_{\varphi} += \frac{m_{\omega\Sigma}}{2048}$
Phasenwinkel der Totzeit der 5Hz DFT berechnen
$\varphi_{tot} = \frac{m_{\omega\Sigma} \cdot 200ms}{2048}$
Korrektur der Phasenverschiebung
$\Delta\varphi = \varphi + \varphi_{tot} - m_{\varphi}$
gleitenden Mittelwert der Phasenverschiebung berechnen:
$m_{\Delta\varphi\Sigma} += \Delta\varphi(z_{time_mittelw}) - \Delta\varphi(z_{time_alt_mittelw})$
gemittelte Phasenverschiebung und gemittelten Phasenwinkel addieren
$\bar{\varphi}_{OS} = m_{\varphi} + \frac{m_{\Delta\varphi}}{2048}$

Struktogramm 17: Korrektur des Phasenwinkels der v. Spannungsüberschwingung

Die Regelung der Oberschwingungskompensation ist jetzt abgeschlossen. Die Regelung der Grundschwingung, der Zwischenkreisspannung und der Stromüberschwingung erfolgt anhand PI-Regler. Die Real- und Imaginärteile der komplexen Zeigern werden separat geregelt, das Unterprogramm des PI-Reglers wird insgesamt 5-mal aufgerufen und folgender Strukturen übergeben.

```

_stPID_Regler    stPID_Regler_D    /*Realteil Grundschwingung */
_stPID_Regler    stPID_Regler_Q    /*Imaginärteil Grundschwingung */
_stPID_Regler    stPID_Regler_U_zk /*Zwischenkreisspannung */
_stPID_Regler    stPID_Regler_OS1_D /*Realteil Oberschwingung */
_stPID_Regler    stPID_Regler_OS1_Q /*Imaginärteil Oberschwingung */

typedef struct _stPID_Regler{
    int e_k_minus_1;    /*Regler Abweichung */
    int p_anteil;    /*Proportionaler Anteil des Reglers */
    int i_anteil;    /*Integraler Anteil des Reglers */
    int d_anteil;    /*Differentialanteil */
    int abtastzeit;    /*Aufruf Zeitabstand*/
    int kp;    /*Proportionalverstärkung */
    int ki;    /*Integralverstärkung */
    int kd;    /*Differentialverstärkung */
    int sollwert;    /*Sollwert*/
    int istwert;    /*Istwert*/
    int stellwert;    /*Ausgabewert des */
    int modus;    /*1=P, 2=PI, 3=PID */
} _stPID_Regler;

```

Die Zwischenkreisspannung wird auf 680V (10800 in Q11.4) geregelt. Der Stellwert der Regelung der Zwischenkreisspannung wird als Sollwert an die Regelung des Wirkanteils des Grundschwingungsstromes weitergegeben. Die PI-Regler kompensieren die Reaktionszeiten der Strecke, die aus den Glättungsdröseln und der DFT bestehen. Wegen des kleinen Nennstromwertes des Stelltransformators wird eine Überschreitung des Sollwertstromes vermieden. Dieses geschieht, indem die PI-Regler mit einer äquivalenten oder langsameren Reaktionszeit als die Strecke eingestellt werden. Die Proportional- und Integralverstärkungen der PI-Regler sind folgende:

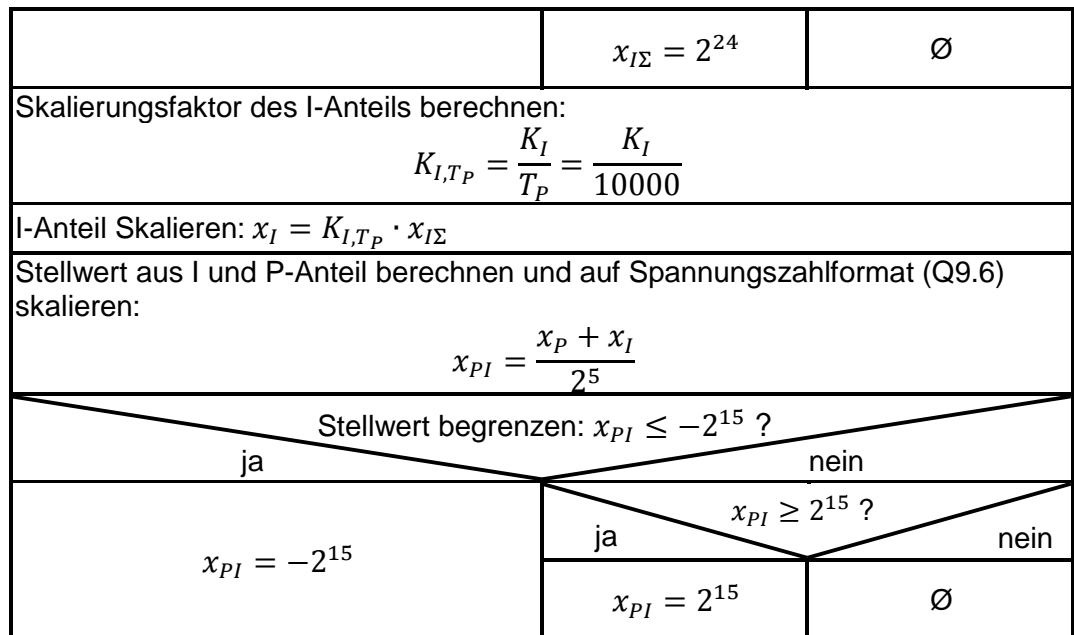
	Grundschwingungsstrom	Zwischenkreisspannung	Stromüberschwingung
Strecke	Impedanz \underline{Z}_d $R_d = 4,1 \Omega$ $L_d = 22,6 \text{ mH}$ (s. Gl. (5.66))	Regelung des Grundschwingungsstromes	Totzeit der 50Hz DFT und Impedanz \underline{Z}_d
Zeitkonstant der Strecke	$\tau_d = \frac{L_d}{R_d} = 5,51 \text{ ms}$		$\tau_s \approx 25 \text{ ms}$
Ki	393	30	150
Kp	2	3,9	1
Zeitkonstant des PI-Reglers	$\tau_{PI} = \frac{K_P}{K_I} = 5,07 \text{ ms}$	$\tau_{PI} = \frac{K_P}{K_I} = 130 \text{ ms}$	$\tau_{PI} = \frac{K_P}{K_I} = 67 \text{ ms}$

Tabelle 19: Proportional- und Integralverstärkungen der PI-Regler

Die Regelabweichung eines diskreten Reglers wird aus der Differenz zwischen dem aktuellen Messwert und dem Messwert der vorherigen Taktperiode berechnet. Die Regelabweichung, der Proportionalanteil, und der Integralanteil werden durch Minimal- und Maximalwert begrenzt.

PI-Regler:

Regelabweichung berechnen: $e = x_{soll} - x_{ist}$			
Regelabweichung begrenzen: $e \leq -2^{15}$?			
ja		nein	
$e = -2^{15}$	ja		nein
	$e = 2^{15}$	\emptyset	
P-Anteil des Reglers berechnen: $x_p = K_P \cdot e$			
P-Anteil begrenzen: $x_p \leq -6 \cdot 2^{15}$?			
ja		nein	
$x_p = -6 \cdot 2^{15}$	ja		nein
	$x_p = 6 \cdot 2^{15}$	\emptyset	
I-Anteil des Reglers berechnen: $x_{I\Sigma} = x_{I\Sigma} + e$			
I-Anteil begrenzen: $x_{I\Sigma} \leq -2^{24}$?			
ja		nein	
$x_{I\Sigma} = -2^{24}$	ja		nein
	$x_{I\Sigma} \geq 2^{24}$?		



Struktogramm 18: diskrete PI-Regler

Invers Park-Transformation des Stellwerts der Regelung der Grundschwingung:

$\begin{bmatrix} u_{g,\alpha} \\ u_{g,\beta} \end{bmatrix} = \begin{bmatrix} \cos \theta_u & -\sin \theta_u \\ \sin \theta_u & \cos \theta_u \end{bmatrix} \cdot \begin{bmatrix} u_{g,d} \\ u_{g,q} \end{bmatrix}$
--

Struktogramm 19: Invers Park-Transformation

Für die Netzsteuerung des Wechselrichters wird der Raumzeiger der Netzspannung mit dem Übersetzungsverhältnis des Stelltransformators skaliert. Die Netzsteuerung und der Stellwert der Regelungen werden zusammen addiert und auf die Zwischenkreisspannung normiert.

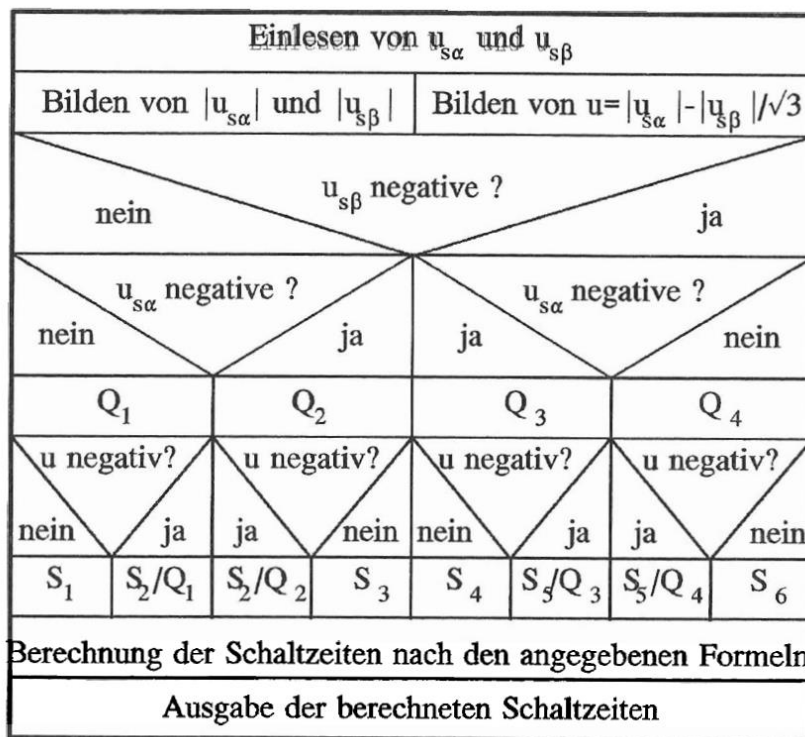
Normierung der Raumzeiger u_s auf die Zwischenkreisspannung:

$\begin{bmatrix} u_{s,\alpha} \\ u_{s,\beta} \end{bmatrix} = \frac{\sqrt{3}}{U_{ZK}} \cdot \left(\begin{bmatrix} u_{vg,\alpha} \\ u_{vg,\beta} \end{bmatrix} + \frac{\sqrt{3}}{2} \cdot \ddot{u} \cdot \begin{bmatrix} u_{netz,\alpha} \\ u_{netz,\beta} \end{bmatrix} \right)$
--

Struktogramm 20: Normierung

Aus dem normierten Raumzeiger werden die Ein- und Ausschaltzeiten der drei Transistorzweige für eine Taktperiode berechnet.

Raumzeigermodulation: Wechselrichterschaltzeiten berechnen:



Struktogramm 21: Schaltzeitenberechnung [QUAN99]

Sektor 1:

$$T_{uein} = T_{waus} = \frac{T}{2} \left(1 - |u_{s\alpha}| - \frac{|u_{s\beta}|}{\sqrt{3}} \right)$$

$$T_{uaus} = T_{wein} = \frac{T}{2} \left(1 + |u_{s\alpha}| + \frac{|u_{s\beta}|}{\sqrt{3}} \right)$$

$$T_{vein} = \frac{T}{2} (1 + |u_{s\alpha}| - \sqrt{3}|u_{s\beta}|); \quad T_{vaus} = \frac{T}{2} (1 - |u_{s\alpha}| + \sqrt{3}|u_{s\beta}|)$$

Sektor 4:

$$T_{uein} = T_{waus} = \frac{T}{2} \left(1 + |u_{s\alpha}| + \frac{|u_{s\beta}|}{\sqrt{3}} \right)$$

$$T_{uaus} = T_{wein} = \frac{T}{2} \left(1 - |u_{s\alpha}| - \frac{|u_{s\beta}|}{\sqrt{3}} \right)$$

$$T_{vein} = \frac{T}{2} (1 - |u_{s\alpha}| + \sqrt{3}|u_{s\beta}|); \quad T_{vaus} = \frac{T}{2} (1 + |u_{s\alpha}| - \sqrt{3}|u_{s\beta}|)$$

Sektor 2:

Quadrant 1: $T_{uein} = \frac{T}{2} (1 - 2|u_{s\alpha}|); \quad T_{uaus} = \frac{T}{2} (1 + 2|u_{s\alpha}|)$

Quadrant 2: $T_{uein} = \frac{T}{2} (1 + 2|u_{s\alpha}|); \quad T_{uaus} = \frac{T}{2} (1 - 2|u_{s\alpha}|)$

$$T_{vein} = T_{waus} = \frac{T}{2} \left(1 - \frac{2|u_{s\beta}|}{\sqrt{3}} \right); \quad T_{vaus} = T_{wein} = \frac{T}{2} \left(1 + \frac{2|u_{s\beta}|}{\sqrt{3}} \right)$$

Sektor 5:

Quadrant 3: $T_{uein} = \frac{T}{2} (1 + 2|u_{s\alpha}|); \quad T_{uaus} = \frac{T}{2} (1 - 2|u_{s\alpha}|)$

Quadrant 4: $T_{uein} = \frac{T}{2} (1 - 2|u_{s\alpha}|); \quad T_{uaus} = \frac{T}{2} (1 + 2|u_{s\alpha}|)$

$$T_{vein} = T_{waus} = \frac{T}{2} \left(1 + \frac{2|u_{s\beta}|}{\sqrt{3}} \right); \quad T_{vaus} = T_{wein} = \frac{T}{2} \left(1 - \frac{2|u_{s\beta}|}{\sqrt{3}} \right)$$

Sektor 3:

$$T_{uein} = T_{vaus} = \frac{T}{2} \left(1 + |u_{s\alpha}| + \frac{|u_{s\beta}|}{\sqrt{3}} \right)$$

$$T_{uaus} = T_{vein} = \frac{T}{2} \left(1 - |u_{s\alpha}| - \frac{|u_{s\beta}|}{\sqrt{3}} \right)$$

$$T_{vein} = \frac{T}{2} (1 - |u_{s\alpha}| + \sqrt{3}|u_{s\beta}|); \quad T_{waus} = \frac{T}{2} (1 + |u_{s\alpha}| - \sqrt{3}|u_{s\beta}|)$$

Sektor 6:

$$T_{uein} = T_{vaus} = \frac{T}{2} \left(1 - |u_{s\alpha}| - \frac{|u_{s\beta}|}{\sqrt{3}} \right)$$

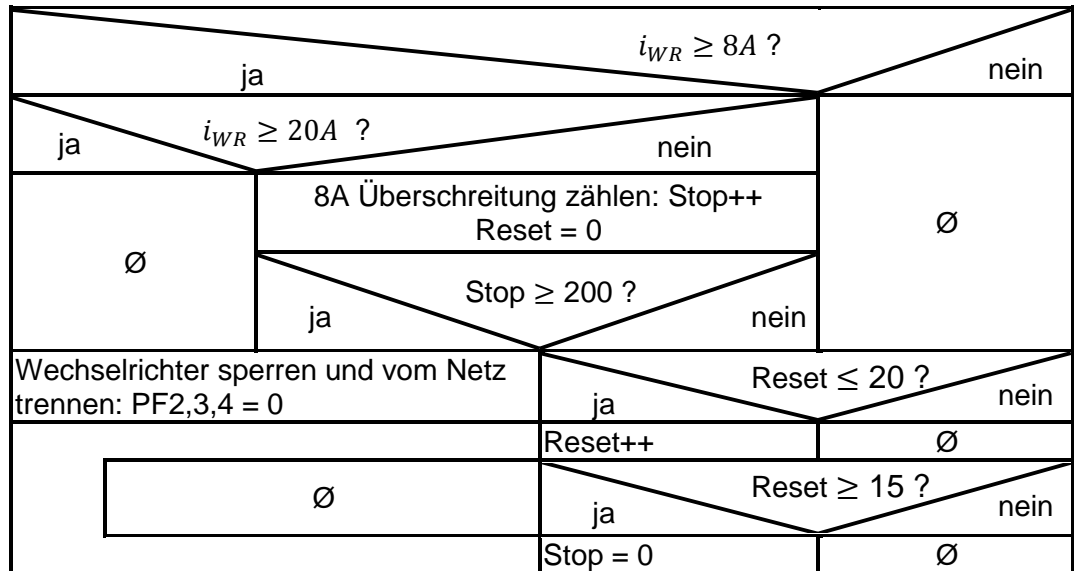
$$T_{uaus} = T_{vein} = \frac{T}{2} \left(1 + |u_{s\alpha}| + \frac{|u_{s\beta}|}{\sqrt{3}} \right)$$

$$T_{vein} = \frac{T}{2} (1 + |u_{s\alpha}| - \sqrt{3}|u_{s\beta}|); \quad T_{waus} = \frac{T}{2} (1 - |u_{s\alpha}| + \sqrt{3}|u_{s\beta}|)$$

Gln. 5.75 Schaltzeitenberechnung [QUAN99]

Eine Überwachung des Wechselrichterstromes sichert die Sperrung und Abtrennung des Wechselrichters im Fehlerfall. Ab 20A wird der Wechselrichter sofort getrennt, ab 8A werden die Überschreitungen gezählt, nach 200-maliger Wiederholung (20ms) erfolgt die Trennung.

Überstromschutz:



Struktogramm 22: Überstromschutz

Die diskrete Regelung der Blindleistungs- und Oberschwingungskompensation ist zu Ende. Sie wird in 25 μ s in der DSP abgearbeitet, bis zum Anfang der neuen Taktperiode wartet der DSP 75 μ s.

5.3 Messergebnisse

Der Prototyp der Intelligenten Ladesäule ist an das 0,4kV Netz des Labors der Hochschule angeschlossen. Die Oberschwingungskompensation wurde auf die 5. Oberschwingung der Netzspannung eingestellt. Für eine bessere Darstellung der Messergebnisse wird die Oberschwingungskompensation 10 Sekunden periodisch ein- und ausgeschaltet. Auf **Abbildung 56** sind 30 Sekunden dargestellt, von 0 bis 10 Sekunden ist die Kompensation eingeschaltet, von 10 bis 20 Sekunden ist sie aus und von 20 bis 30 Sekunden ist sie wieder an. Das **Diagramm a)** stellt die Auswertung der Phasenwinkel der 5. Spannungsoberschwingung der drei Leitern u,v,w anhand der gleitenden DFT des DSP dar. Der Phasenversatz von 120° zwischen den drei Leitern ist erkennbar und deutet auf eine Gegensystemoberschwingung.

$$\varphi_{OSu} = \varphi_{OSv} + 120^\circ = \varphi_{OSw} - 120^\circ \quad (5.76)$$

Die abfallende Sägezahnfunktion des Phasenwinkels deutet auf eine leichte Abweichung von etwa -50mHz zwischen der Frequenz der DFT und der Netzfrequenz.

Aus dem Phasenwinkel φ_{OSu} wird der in **Diagramm b)** dargestellte Sollwert der Regelung der Stromoberschwingung gebildet. Die Amplitude dieses Sollwerts ist auf 2A eingestellt, in der Zeit von 10 bis 20 Sekunden ist der Sollwert auf null gesetzt, die Kompensation ist aus.

Die Nachstellzeit der Regelung der Stromoberschwingung ist anhand der Stellwerte auf **Diagramm c)** erkennbar, die sprunghaftigen Änderungen des Sollwerts beim Ein- und Ausschalten ruft einen exponentiell ablaufenden Vorgang des Stellwerts hervor.

Während der Kompensation sind die langsameren Änderungen des Sollwerts vollständig in dem Stellwert nachgebildet.

Der Istwert auf **Diagramm d)** stellt die Auswertung der 5. Stromoberschwingung des Leiters u anhand der gleitenden DFT des DSP dar. Der Sollwert und der Istwert sind identisch, dies deutet auf eine erfolgreiche Regelung der Stromoberschwingung hin.

Die positive Wirkung der Oberschwingungskompensation ist auf dem **Diagramm e)** zu erkennen, die Amplitude der 5. Spannungsoberschwingung verringert sich um 0,3V wenn die Kompensation eingeschaltet ist. Der Betrag der Netzimpedanz bei 250Hz kann demzufolge auf 150mΩ geschätzt werden.

Auf dem **Diagramm f)** ist die Zwischenkreisspannung des Umrichters dargestellt, die Regelung auf 680V ist erkennbar, aber noch nicht optimal eingestellt. Das Ein- und Ausschalten der Kompensation führt zu Abweichungen der Zwischenkreisspannung.

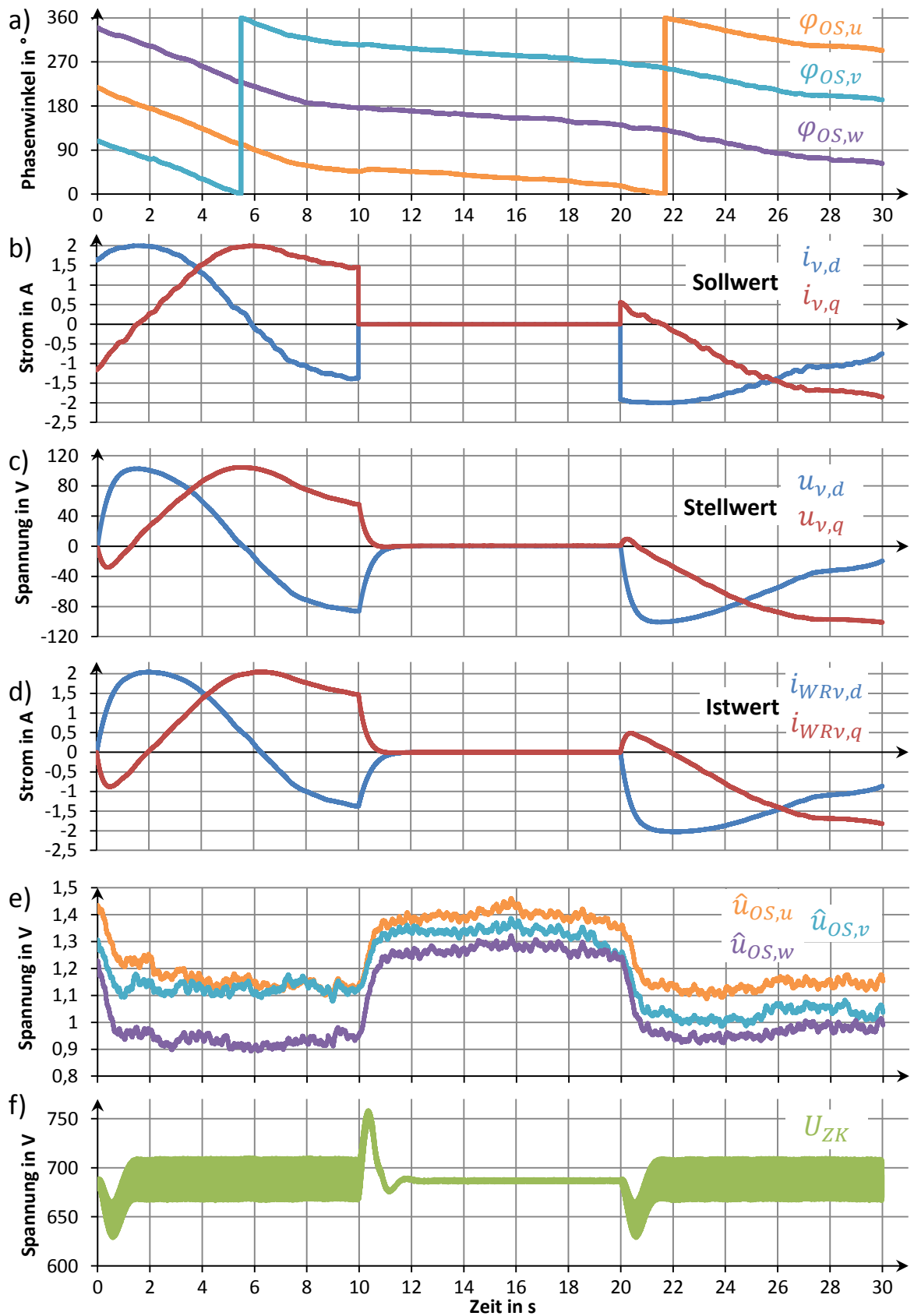


Abbildung 56: Kompensation der 5. Spannungsüberschwingung des 0,4kV Netzes

- a) Phasenwinkel der Spannungsüberschwingung
- b) Sollwert des Kompensationsstromes
- c) Stellwert des Kompensationsstromes
- d) Istwert des Kompensationsstromes
- e) Amplitude der Spannungsüberschwingung
- f) Zwischenkreisspannung des Umrichters

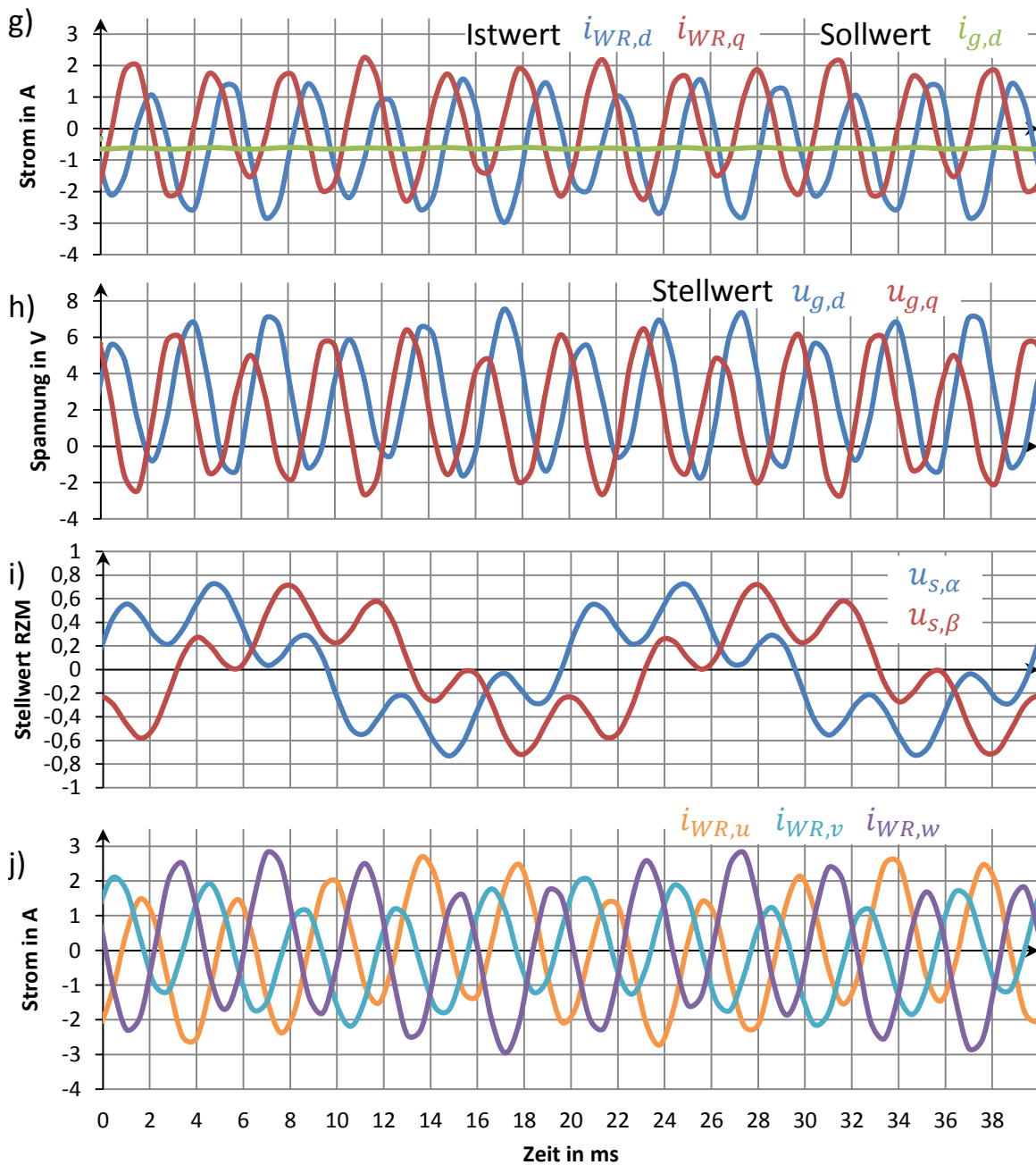


Abbildung 57: Kompensation der 5. Oberschwingungsspannung des 0,4kV Netzes

g) Istwert und Sollwert des Grundschwingungsstromes

h) Stellwert des Grundschwingungsstromes

i) normierter Stellwert u_s

j) dreiphasiger Wechselrichterstrom

Die **Abbildung 57** stellt einen 40ms Zeitabschnitt der Oberschwingungsbehafteten Messwerte bei eingeschalteter Kompensation dar. Die zwei d,q Komponenten der Istwerte der Grundschwingung auf **Diagramm g)** beinhalten einen Gleichanteil und einen Wechselanteil. Der Gleichanteil der d-Komponente ist negativ (\rightarrow Wirkleistungsverbrauch) und der Gleichanteil der q-Komponente ist null (\rightarrow keine Blindleistung). Der Wechselanteil ist die 5.Oberschwingung, diese Gegensystem Oberschwingung besitzt

nach der Park-Transformation eine Frequenz von 300Hz. Die d-Komponente des Grundschiwungssollwerts auf Diagramm g) wird von der Regelung der Zwischenkreisspannung beliefert und bezeichnet den eigenen Wirkleistungsverbrauch der Intelligenten Ladesäule (Messung im Erzeugerpeilsystem).

Auf **Diagramm h)** ist der Stellwert der Grundschiwung dargestellt. Die 5. Oberschiwung wird von dem P-Regler durchgelassen und ist deshalb immer noch präsent, der Gleichanteil des Stellwertes deckt den Wirkleistungsverbrauch und die Korrektur des netzgesteuerten Betriebs.

Der gesamte Stellwert \underline{u}_s auf **Diagramm i)** ist auf der Zwischenkreisspannung normiert und deshalb einheitslos. Der Stellwert \underline{u}_s ist die Summe der Oberschiwungsregelung und der Grundschiwungsregelung, sowie der Netzsteuerung. Die Überlagerung der 50Hz und 250Hz Schwiwungen sind gut erkennbar.

Auf **Diagramm j)** ist der dreiphasige Wechselrichterstrom dargestellt, der ebenfalls die Grundschiwung und die 5. Oberschiwung beinhaltet. Die Unsymmetrie des Wechselrichterstroms wurde nicht genau untersucht und ist sehr wahrscheinlich auf die Unsymmetrie des Transformators 2, der Glättungsdröseln und des Stelltransformators zurückzuführen.

Alle Messwerte die bis jetzt erläutert wurden, sind mit einer Zeitauföslung von 100 μ s mit der DSP erfasst worden. Aufgrund der niedrigen Zeitauföslung ist auf den Diagrammen g) bis j) eine Sinusformabweichung der 5. Oberschiwung zu sehen.

Die Oszilloskopbilder auf **Abbildung 58** zeigen eine 50ms Aufnahme des dreiphasigen Wechselrichterstromes mit einer hohen Zeitauföslung von 8 μ s.

Auf dem oberen Oszilloskopbild ist die Kompensation eingeschaltet, der Wechselrichterstrom (rot: i_{WRu} , blau: i_{WRv} , lila: i_{WRw}) besteht aus der Grundschiwung, der 5. Oberschiwung und der 10kHz PWM des Wechselrichters.

Die 5. Oberschiwung besitzt eine ordentliche Sinusform und ist im Gegensystem.

Auf dem unteren Teil der Oszilloskopbilder ist das Amplitudenspektrum in dBV der Netzspannung (Leiter u) dargestellt, die Frequenzauföslung beträgt 5Hz.

Dabei wurde die 5. Oberschiwung markiert, ihre Amplitude beträgt während der Kompensation 1V und 1,3V ohne Kompensation, die positive Wirkung der Oberschiwungskompensation ist damit bestätigt.

Auf dem unteren Oszilloskopbild ist die Kompensation ausgeschaltet, ein Grundschiwungsstrom von 0,5A Amplitude versorgt den Wechselrichter, die Messung erfolgte im Erzeugerpeilsystem.

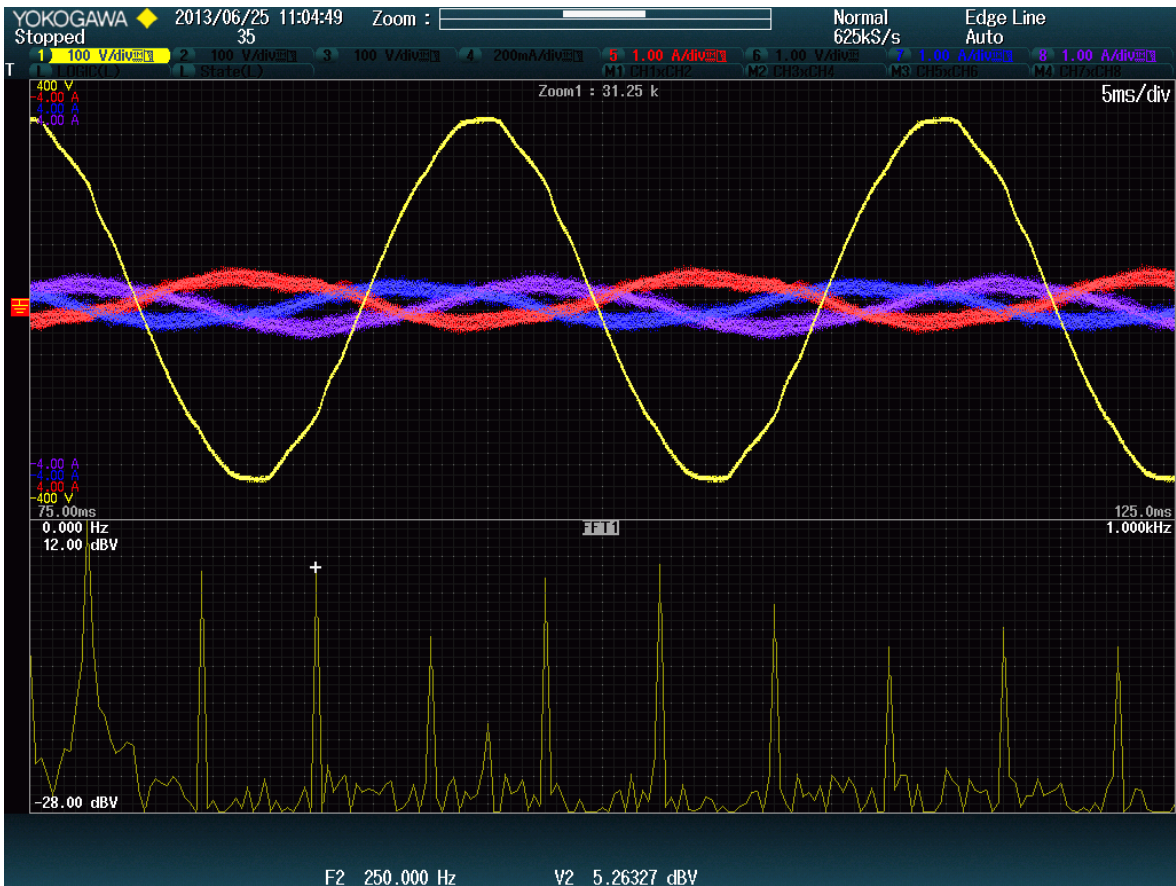
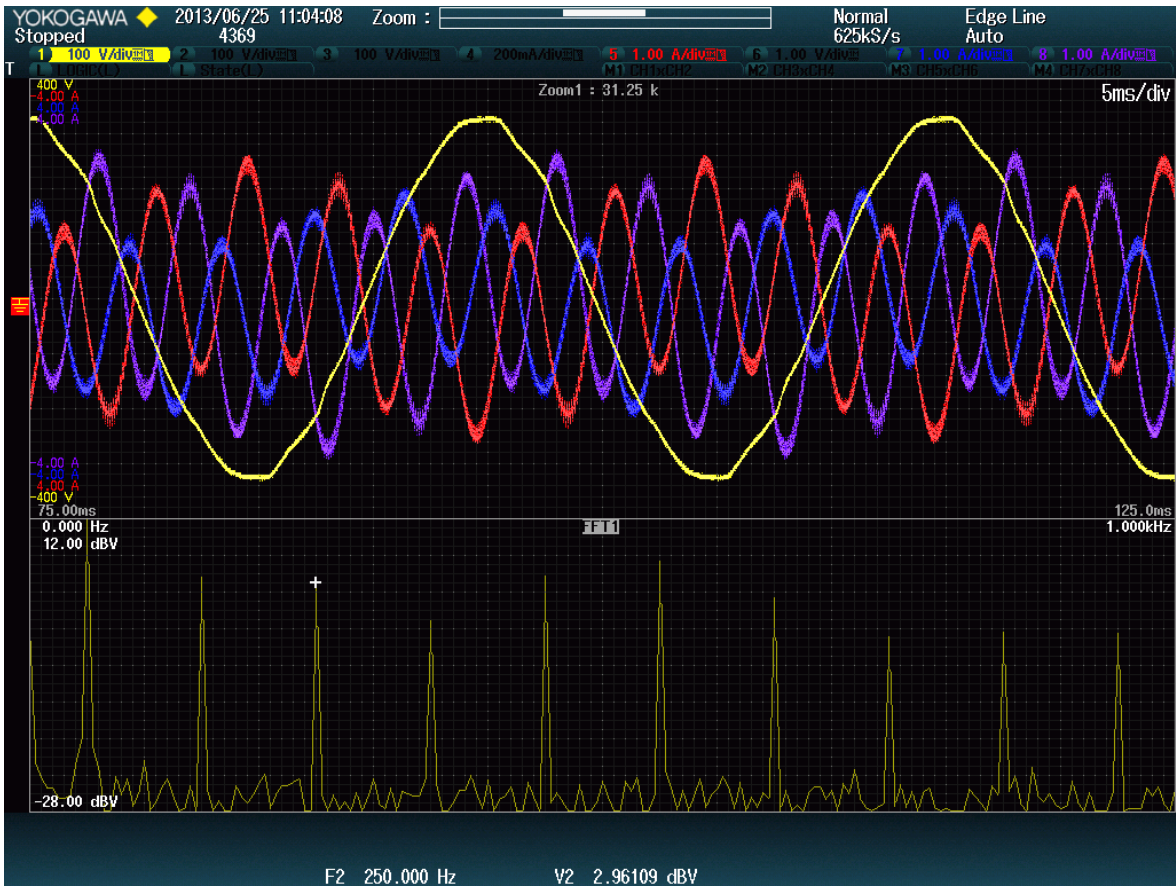


Abbildung 58: Kompensation der 5. Oberschwungungsspannung des 0,4kV Netzes

rot: i_{WRu} , blau: i_{WRv} , lila: i_{WRw} , gelb: $u_{netz,u}$, FFT: $u_{netz,u}$

Abschließend wurde der dreiphasige Wechselrichterstrom mit und ohne Kompensation für 1s auf dem Oszilloskop aufgenommen. Mit der Software Matlab wurden daraus zwei Oberschwingungsmessungen nach DIN 61000-4-7 berechnet, die auf der **Abbildung 59** dargestellt sind.

Die Amplituden der Oberschwingungsströme liegen um das 10-fache unter den gesetzlichen Grenzwerten, die in Tabelle 4 (Seite 19) angegeben sind.

Der Oberschwingungsgehalt des Wechselrichterstroms ist abhängig von der betriebenen Leistung der Intelligenten Ladesäule und würde bei einer Leistung von 10kW steigen.

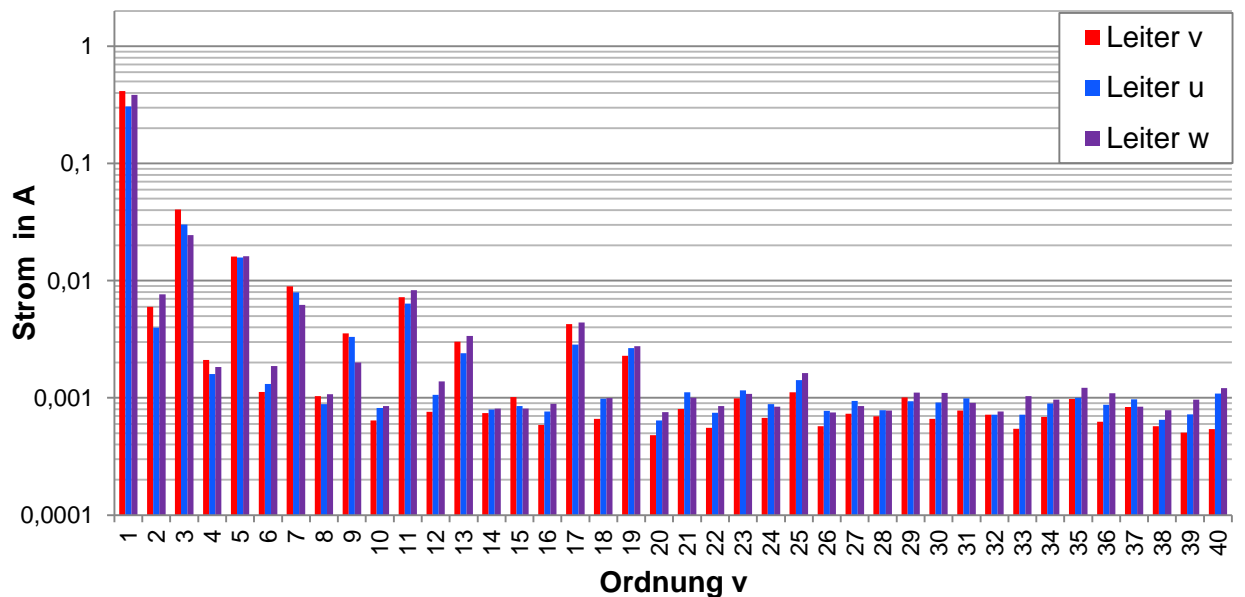
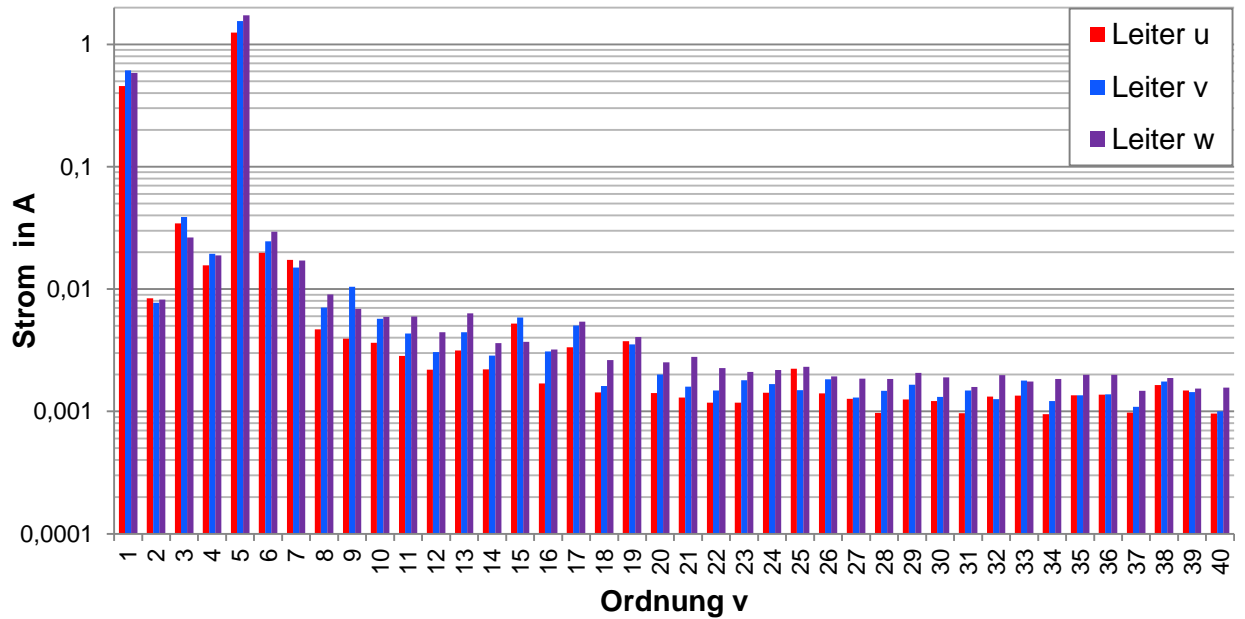


Abbildung 59: Oberschwingungsgehalt des Wechselrichterstroms

5.4 Zusammenfassung

Der Prototyp der Intelligenten Ladesäule war vor dem Beginn dieser Arbeit bereits aufgebaut und mit einem Blindleistungsregler ausgerüstet. Die zentralen Komponenten dieses Prototyps sind ein 11kW Wechselrichter und ein Digitaler Signalprozessor (DSP). Der Wechselrichterausgang ist an das dreiphasige 0,4kV Netz angeschlossen, die Gleichspannung einer Fahrzeugbatterie wird durch einen Kondensator im Zwischenkreis ersetzt, der von einer dreiphasigen Netzgleichrichtung gespeist wird. Die Einspeisung ins Netz erfolgt über den Wechselrichterausgang, gesteuert von einer Pulsweitenmodulation (PWM) mit einer Taktfrequenz von 10kHz, was eine maximale Rechenzeit von 100µs zulässt. Demzufolge war vor der Implementierung des diskreten Reglers für die Oberschwingungskompensation eine Rechenzeitoptimierung der bestehenden Software zur Blindleistungsbereitstellung nötig. Die Substitution von trigonometrischen Funktionen (Cosinus-, Sinus- und Arkustangens) durch Tabellen und die Vermeidung von rechenintensiven Operationen wie der Division, führte zu einer Reduzierung der bisherigen Rechenzeit und ermöglichte die Erweiterung der Software.

Die Kompensation einer Netzoerschwingung wird durch das Einprägen einer Stromerschwingung synchron zur ausgewählten Spannungsoerschwingung erzielt. Für die Messung dieser Oberschwingungen wurde die gleitende diskrete Fourier Transformation und ein 50Hz Filter implementiert. Um die Stromerschwingung mit der Spannungsoerschwingung zu synchronisieren, wird deren Phasenverschiebung ausgewertet und geregelt. Für eine stabile Synchronisierung wurde der gemessene Phasenwinkel der Spannungsoerschwingung gefiltert und mit einer Totzeitkorrektur angepasst. Die Regelung der Stromerschwingung wurde anhand des bestehenden PI-Reglers realisiert und für Mitsystem- und Gegensystemerschwingungen erweitert. Mit dem implementierten Programm wurde im Labor gezielt eine phasennegierte Stromerschwingung ins Netz eingepreßt und damit eine ausgewählte Spannungsoerschwingung reduziert.

6 Zusammenfassung und Ausblick

Im abschließenden Kapitel werden die Ergebnisse zusammengefasst und eine Bewertung der Leistung aus Sicht des Autors vorgenommen. Ein Ausblick zeigt die geplante Weiterentwicklung des Projektes auf.

6.1 Ergebnisse

Die Intelligente Ladesäule ist ein Forschungsprojekt der Professur Antriebs- und Regelungstechnik an der Hochschule Mittweida. Die Kernaufgabe einer Intelligenten Ladesäule ist das sichere und schnelle Laden eines E-Fahrzeuges. Zusatzfunktionen der Ladesäule zielen auf die Verbesserung der Netzspannungsqualität, so kann die Ladesäule zu jeder Zeit und unabhängig von der Ladung eines Elektrofahrzeuges die Blindleistung bereitstellen und die Spannungsüberschwingungen kompensieren. Das Thema „Entwicklung eines echtzeitfähigen Algorithmus zur Ermittlung und Kompensation der Netzüberschwingungen und Implementierung im Rechner der Intelligenten Ladesäule“ ergab sich als Teilgebiet der Entwicklung im Forschungsprojekt und war damit Aufgabenstellung dieser Masterarbeit.

Zuerst wurden die mathematischen Grundlagen wie die gleitende diskrete Fourier Transformation zur Oberschwingungsmessung und die Raumzeigerdarstellung des dreiphasigen Netzes als Grundlage der gewählten Reglerstruktur des Wechselrichters geschildert.

Aus den Normen des Niederspannungsnetzes wurden die Oberschwingungsgrenzwerte ermittelt und daraus die Oberschwingungsmessverfahren für Strom und Spannung abgeleitet.

Auf Basis einer theoretischen Vorbetrachtung konnte der Verbrauch eines Kompensationsstroms, bereitgestellt durch eine ideale Spannungsquelle, zur Reduzierung der Oberschwingungen im Netz simuliert werden. Die erfolgreiche Simulation war die Basis für die Entwicklung der Oberschwingungskompensation, die es mit Hilfe der Intelligenten Ladesäule umzusetzen galt. Im Ergebnis der Entwicklung wurden Oberschwingungsströme phasennegiert ins Netz eingeprägt, die zur Verbesserung der Oberschwingungsbelastung führten.

Vor dem Beginn dieser Arbeit war der Prototyp der Intelligenten Ladesäule bereits aufgebaut. Eine Beschreibung der Hardwarekomponenten des Prototyps wurde durchgeführt. Die Software für die Regelung des Energieflusses und der Blindleistungskompensation wurde optimiert, und um eine Software für die Regelung der

Oberschwingungskompensation erweitert, so dass die Ströme der Grund- und Oberschwingung separat geregelt werden können. Abschließend wurde die erfolgreiche Kompensation einer Oberschwingung im dreiphasigen Netz gezeigt und dokumentiert.

6.2 Bewertung der Arbeit

Alle Anforderungen und Ziele dieser Masterarbeit wurden erfüllt. Sie bildet damit die Grundlage zur weiteren Entwicklung der Oberschwingungskompensation auf Basis des Prototyps der Intelligenten Ladesäule.

Ausgehend von Simulationen und erfolgreichen Tests konnte bewiesen werden, dass die Intelligente Ladesäule durch die Reduzierung der Netzoberschwingungen, eine wesentliche Rolle zur Verbesserung der Netzspannungsqualität beitragen kann.

Die aus der vorangegangenen Bachelorarbeit gewonnenen Erkenntnisse konnten mit Hilfe dieser Arbeit erweitert und im Prototyp der Intelligenten Ladesäule praktisch umgesetzt werden. Der quantitative Nachweis der Oberschwingungskompensation konnte im Labor der Hochschule erbracht werden.

Die Blindleistungsbereitstellung und die Oberschwingungskompensationsregelung können während, aber auch außerhalb der Ladung eines E-Fahrzeuges erfolgen, so dass der Netzbetreiber dezentral über die „Intelligente Ladesäule“ die Netzspannung regeln kann.

6.3 Ausblick

Im Rahmen des Forschungsprojekts „CO₂- neutrale E-Mobilität im Smart Grid“ wird ein Prototyp der Intelligenten Ladesäule entwickelt.

Die Entwicklungen dieser Masterarbeit münden in eine Promotion, in der die Oberschwingungskompensation weiter optimiert werden soll. Speziell die Anpassung an variable Netzimpedanzen und den im Tagesverlauf wechselnden Oberschwingungsgehalt stellt eine Herausforderung dar, für die es Lösungen zu entwickeln gilt. Außerdem müssen Untersuchungen stattfinden in wieweit es möglich ist, mehrere Oberschwingungen der Netzspannung zu kompensieren und welche Hardwarekonfiguration dafür geeignet ist.

Anlagen

Anlage 1: Standardvektoren und Ausgangsspannungen des Wechselrichters	88
Anlage 2: Ringstellpartransformator	89
Anlage 3: Transformator 2.....	91
Anlage 4: Glättungsrosseln.....	91
Anlage 5: Transformator 1	92
Anlage 6: Spannungswandler LEM CV 3-500	93
Anlage 7: Spannungswandler CV 3-1500	94
Anlage 8: Stromwandler LEM IT 200-S	95
Anlage 9: A/D-Wandler AD7606.....	96
Anlage 10: DSP Blackfin ADSP BF537 von Analog Devices	98
Anlage 11: Quellecode des gesamten Programms.....	100

Anlage 1: Standardvektoren und Ausgangsspannungen des Wechselrichters

Standard-vektoren	TR1 TR4	TR2 TR5	TR3 TR6	u_{TR14}	u_{TR25}	u_{TR36}	Schaltung des Spannungsteilers
\underline{u}_0	0	0	0	0	0	0	
\underline{u}_1	1	0	0	$\frac{2}{3}$	$-\frac{1}{3}$	$-\frac{1}{3}$	
\underline{u}_2	1	1	0	$\frac{1}{3}$	$\frac{1}{3}$	$-\frac{2}{3}$	
\underline{u}_3	0	1	0	$-\frac{1}{3}$	$\frac{2}{3}$	$-\frac{1}{3}$	
\underline{u}_4	0	1	1	$-\frac{2}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	
\underline{u}_5	0	0	1	$-\frac{1}{3}$	$-\frac{1}{3}$	$\frac{2}{3}$	
\underline{u}_6	1	0	1	$\frac{1}{3}$	$-\frac{2}{3}$	$\frac{1}{3}$	
\underline{u}_7	1	1	1	0	0	0	

Anlage 2: Ringstellsparttransformator

Datenblatt

Produktdetail

Bezeichnung: 115DF 230V 50-400Hz / 0-230/115V 5,0A

Artikelnummer: 61-200115-005

Einphasen-Ringstellsparttransformator nach VDE0552

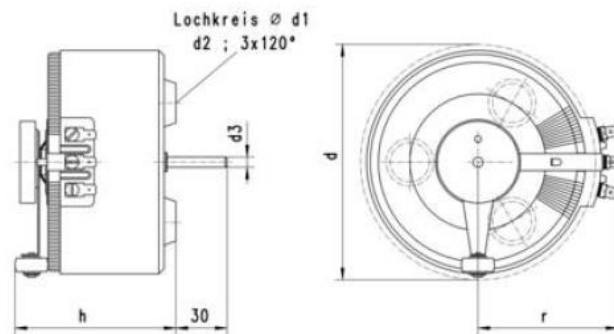
- Mit Mittelanzapfung (Summenstrom=Nennstrom)
- Dreipunktbefestigung nach Serie 684
- Handantrieb fußregelnd, Welle D6mm 30mm lang ab Auflage
- Flachsteckanschluss 6,3x0,8mm

Spannungs-Eingänge

1x 230V 5.50A 50-60Hz

Spannungs-Ausgänge

1x 0-230/115V 5.00A



Abmessung

(alle Werte in mm)

d	d1	d2	d3	h	r
135,00	80,00	6,00	6,00	95,00	83,00

Zubehör:

[Drehknopf K60-6](#)

[Drehknopf K90-6](#)

Medien-Downloads:

keine Dateien vorhanden

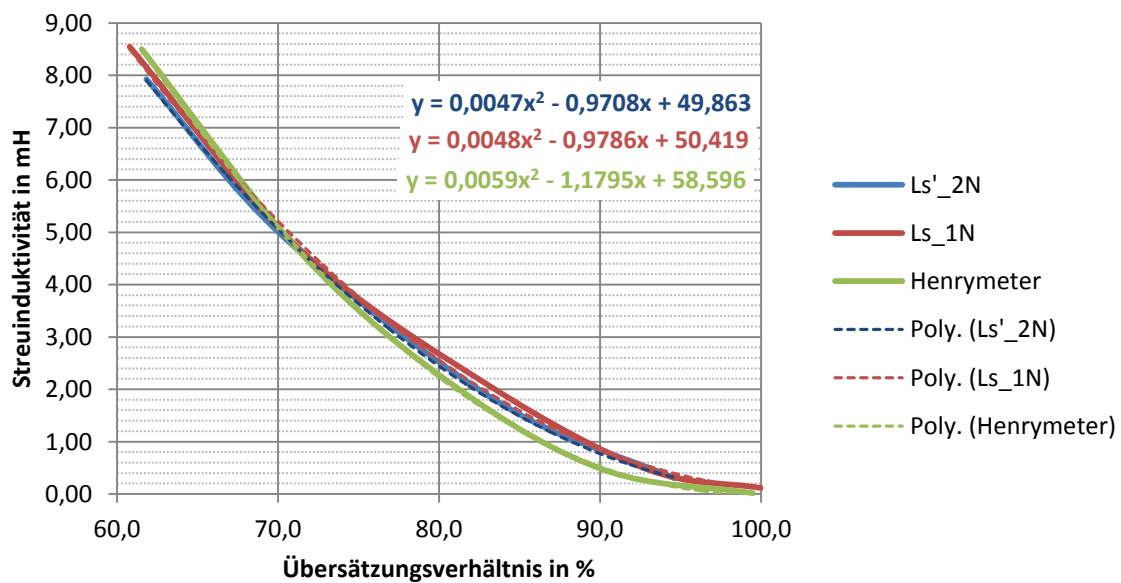
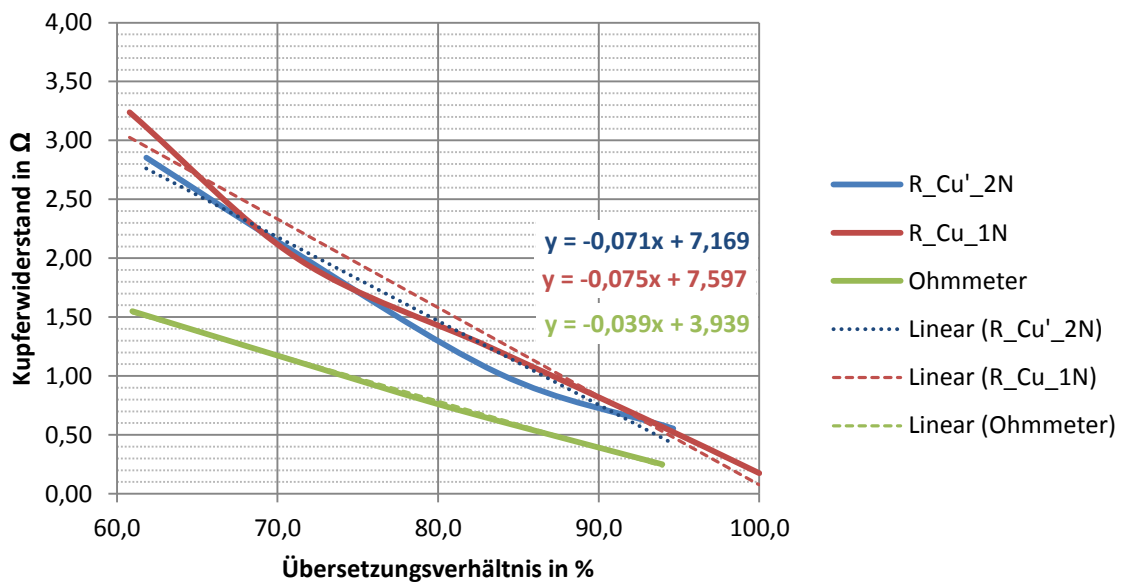
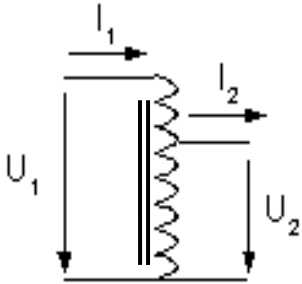


Artikelnummer:	61-200115-005
Spannungs-Eingänge	1x 230V 5.50A 50-60Hz
Spannungs-Ausgänge	1x 0-230/115V 5.00A
Bauleistung:	1.150 VA
Schaltgruppe:	Ia0
Schutzklasse:	I
Isolationsklasse:	T40B
Schutzart:	IP00
Umgebungstemperatur:	min 0 °C / max 40 °C
MTBF:	30.000 h
Kupfergewicht:	0,37 kg
Gewicht:	4,20 kg
Baugröße:	115DF
Produktserie:	Serie 684
Lieferzeit:	4 AT
Bruttopreis:	182,80 EUR +CuTZ +MwSt

Möller-Preussler GmbH, Gutenbergring 55a, 22848 Norderstedt
Tel. 0049/40/534 20-0 / Fax 0049/40/534 20-40
www.moeller-preussler.de / info@moellerpreussler.de

Messung der Ersatzschaltbildelemente des Stelltransformators

Die Kurzschlussmessung wurde an beiden Seite des Transformators sukzessive durchgeführt, die „2N“ seitige gemessenen Messwerte sind auf die „1N“ Seite transformiert → Strichgrößen: 2N´.



Kupferwiderstand: $R_{Cu} = f(\ddot{u}) = -0,0745 \cdot \ddot{u} + 7,5$

$$R_{Cu1} = \frac{R_{Cu}}{2} \qquad R_{Cu2} = \frac{R_{Cu} \cdot \ddot{u}^2}{2}$$

Streuinduktivität: $L_S = f(\ddot{u}) = \frac{0,00513 \cdot \ddot{u}^2 - 1,043 \cdot \ddot{u} + 52,96}{1000}$

$$L_{S1} = \frac{L_S}{2} \qquad L_{S2} = \frac{L_S \cdot \ddot{u}^2}{2}$$

Hauptinduktivität: $L_h = 16,1 \text{ H}$

Eisenverluste: $R_{Fe} = 7300\Omega$

Anlage 3: Transformator 2

Nennleistung: 2,77kVA
 Nennspannung Primär: 3 x 230 – 400V
 Nennspannung Sekundär: 3 x 24 – 127 – 230 – 400V
 Nennstrom: 6 – 8 – 6 – 4A

Messergebnisse der Sprungantwort des kurzgeschlossenen Transformators

Kupferwiderstand: $R_{Cu} = 0,082 \Omega$

Streuinduktivität: $L_S = 0,58 \text{ mH}$

Anlage 4: Glättungsdrosseln

Die drei Glättungsdrosseln des Herstellers Siemens (Typ: 4ET3011-3AA00) sind identisch, sie bestehen aus zwei Wicklungskreisen und einem Magnetkreis. Durch die gegengerichtete Parallelschaltung der zwei Wicklungskreise wurden folgende Induktivitäten und Widerstände gemessen.

Glättungsdrosseln	1	2	3
Spannung in V	2,78	4,49	4,94
Strom in A	0,509	0,823	0,897
Wirkleistung in W	0,4	0,9	1,13
Blindleistung in Var	1,4	3,7	4,42
Induktivität in mH	17,2	17,39	17,49
Widerstand in Ω	1,54	1,33	1,40

Der Nennstrom in Parallelschaltung beträgt 7A.

Anlage 5: Transformator 1

Typ:	DRUE 2400
Nennleistung:	2,4kVA
Nennspannung Primär:	3 x 400V
Nennstrom Primär:	3,7A
Nennspannung Sekundär:	3 x 400V + N
Nennstrom Sekundär:	3,46A
Schaltung:	Dyn5



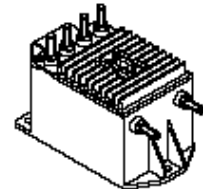
Spannungswandler CV 3-500

$$V_{PN} = 350 \text{ V}$$

Für die elektronische Messung : DC, AC, Impuls...,
mit galvanischer Trennung zwischen dem Primärkreis
(Hochspannung) und dem Sekundärkreis (elektronischer Kreis).



0633



Elektrische Daten

V_{PN}	Primärnennspannung, effektiv	350	V
V_P	Primärspannung, Messbereich	0 .. ± 500	V
V_S	Sekundärspannung @ V_{Pmax}	10	V
K_N	Übersetzungsverhältnis	500 V / 10 V	
R_L	Eingangsimpedanz des Messkreises	≥ 1	kΩ
C_L	Kapazitive Belastung	≤ 5	nF
V_C	Versorgungsspannung (± 5 %)	± 15	V
I_C	Stromaufnahme	$32 + V_S / R_L$	mA
V_d	Prüfspannung, effektiv, 50 Hz, 1 mn	6	kV
V_o	Glimmaussetzspannung, effektiv @ 10 pC	2	kV

Eigenschaften

- Elektronischer Kompensationswandler
- Gehäuse aus isolierendem selbstlöschendem Material UL 94-V0
- Patent angemeldet.

Vorteile

- Hervorragende Messgenauigkeit
- Sehr gute Linearität
- Geringe Temperaturdrift
- Kurze Ansprechzeit
- Weiter Frequenzbereich
- Grosse Bandbreite
- Geringe Störanfälligkeit gegenüber Fremdfeldern
- Geringe Gleichtaktstörung.

Genauigkeit - Dynamisches Verhalten

			Typ	Max	
X_G	Globale Genauigkeit @ V_{Pmax}	$T_A = 25^\circ\text{C}$ - 40°C .. + 85°C		± 0.2	%
V_O	Offsetspannung @ $V_P = 0$	$T_A = 25^\circ\text{C}$ - 40°C .. + 85°C		± 5.0	mV
t_r	Ansprechzeit ¹⁾ @ 90 % von V_{PN}		0.3		µs
dv/dt	dv/dt bei optimaler Kopplung		500		V/µs
f	Frequenzbereich (- 1 dB) @ V_{PN}		DC .. 300		kHz

Anwendungen

- Drehstrom- und Servoantriebe, Generatoren
- Stromrichter für Gleichstromantriebe
- Unterbrechungsfreie Stromversorgungen (USV)
- Stromversorgungen für Schweissanlagen.

Allgemeine Daten

T_A	Umgebungstemperatur	- 40 .. + 85	°C
T_S	Lagertemperatur	- 45 .. + 90	°C
P	Primärverlustleistung, dauernd	3.1	W
R_1	Primärwiderstand	40	kΩ
m	Masse	560	g
	Normen ²⁾	EN 50155	

Anmerkungen : ¹⁾ Mit einem dv/dt von 500 V/µs

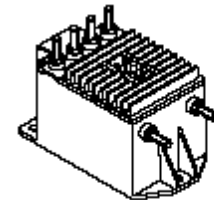
²⁾ Die Liste der durchgeführten Versuche ist auf Anfrage erhältlich.



Voltage Transducer CV 3-1500

$$V_{PN} = 1000 \text{ V}$$

For the electronic measurement of voltages : DC, AC, pulsed..., with a galvanic isolation between the primary circuit (high voltage) and the secondary circuit (electronic circuit).



Electrical data

V_{PN}	Primary nominal r.m.s. voltage	1000	V
V_P	Primary voltage, measuring range	0 .. ± 1500	V
V_S	Secondary analog voltage @ V_{Pmax}	10	V
K_N	Conversion ratio	1500 V/10 V	
R_L	Load resistance	≥ 1	k Ω
C_L	Capacitive loading	≤ 5	nF
V_C	Supply voltage ($\pm 5\%$)	± 15	V
I_C	Current consumption	$32 + V_S/R_L$	mA
V_d	R.m.s. voltage for AC isolation test, 50 Hz, 1 mn	6	kV
V_e	R.m.s. voltage for partial discharge extinction @ 10 pC	2	kV

Accuracy - Dynamic performance data

			Typ	Max	
X_G	Overall accuracy @ V_{Pmax}	$T_A = 25^\circ\text{C}$		± 0.2	%
		-40 $^\circ\text{C}$.. +85 $^\circ\text{C}$		± 0.6	%
V_O	Offset voltage @ $V_P = 0$	$T_A = 25^\circ\text{C}$		± 5.0	mV
		-40 $^\circ\text{C}$.. +85 $^\circ\text{C}$		± 13.0	mV
t_r	Response time ¹⁾ @ 90 % of V_{Pmax}		0.4		μs
dv/dt	dv/dt accurately followed		900		V/ μs
f	Frequency bandwidth (-1 dB) @ 33 % of V_{PN}		DC .. 800		kHz

General data

T_A	Ambient operating temperature	-40 .. +85	$^\circ\text{C}$
T_S	Ambient storage temperature	-45 .. +90	$^\circ\text{C}$
P	Total primary power loss	2.8	W
R_1	Primary resistance	360	k Ω
m	Mass	560	g
	Standards	EN 50155	

Features

- Closed loop (compensated) voltage transducer
- Insulated plastic case recognized according to UL 94-V0
- Patent pending.

Advantages

- Excellent accuracy
- Very good linearity
- Low thermal drift
- Low response time
- High bandwidth
- High immunity to external interference
- Low disturbance in common mode.

Applications

- AC variable speed drives and servo motor drives
- Static converters for DC motor drives
- Uninterruptible Power Supplies (UPS)
- Power supplies for welding applications
- Railway overhead line voltage measurement.

Note: ¹⁾ With a dv/dt of 900 V/ μs

Anlage 8: Stromwandler LEM IT 200-S



High Performance Current Transducer IT 200-S ULTRASTAB

$I_{PM} = 0 \dots 200 \text{ A}$

For the electronic measurement of currents: DC, AC, pulsed..., with galvanic isolation between the primary circuit and the secondary circuit.



Electrical data			
I_{PN}	Primary nominal current DC	200	A
I_{PN}	Primary nominal current rms	141	A
I_{PM}	Primary current, measuring range	0 .. ± 200	A
I_p	Max overload capability 100 ms ¹⁾	± 1000	A
R_M	Measuring resistance	R_{Mmin} R_{Mmax}	
	Over operating current, temperature and supply voltage range	0 30	Ω
I_S	Secondary current	0 .. ± 200	mA
I_{SN}	Secondary nominal current rms	141	mA
K_N	Conversion ratio	1 : 1000	
V_C	Supply voltage ($\pm 5\%$)	± 15	V
I_C	Current consumption $\pm 15 \text{ V}$	$\leq 80 + I_S$	mA

Accuracy - Dynamic performance data			
ϵ_L	Linearity error ²⁾	≤ 3	ppm
I_{OE}	Electrical offset current + self magnetization + effect of earth magnetic field @ $T_A = 25^\circ\text{C}$ ²⁾	< 80	ppm
ΔI_{OE}	Offset stability (no load) ²⁾	< 1	ppm/month
TCI_{OE}	Temperature coefficient of I_{OE} ($10^\circ\text{C} \dots 50^\circ\text{C}$) ²⁾	< 2	ppm/K
	Offset vs. power supply stability @ $T_A = 25^\circ\text{C}$ ²⁾ @ $V_C = \pm 15 \text{ V} \pm 5\%$	< 6	ppm/% of $V_C = \pm 15 \text{ V}$

Dynamic performance data			
BW	Frequency bandwidth for small signal 0.5 %, of I_{PN} (DC) ($\pm 1 \text{ dB}$)	DC .. 150	kHz
	($\pm 3 \text{ dB}$)	DC .. > 500	kHz
di/dt	di/dt accurately followed	> 100	A/ μs
t_r	Response time ¹⁾ to 90 % of I_{PN} step	< 1	μs

Note: ¹⁾ With a di/dt of 100 A/ μs .

Features

- Closed loop (compensated) current transducer using an extremely accurate zero flux detector
- Electrostatic shield between primary and secondary circuit.

Special features

- 9-pin D-Sub male secondary connector
- Output indicates the transducer state
- LED indicators confirms normal operation.

Advantages

- Very high accuracy
- Excellent linearity
- Extremely low temperature drift
- Wide frequency bandwidth
- High immunity to external electrostatic and magnetic fields interference
- No insertion losses
- High resolution
- Low noise on output signal
- Low noise reflected back onto primary conductor.

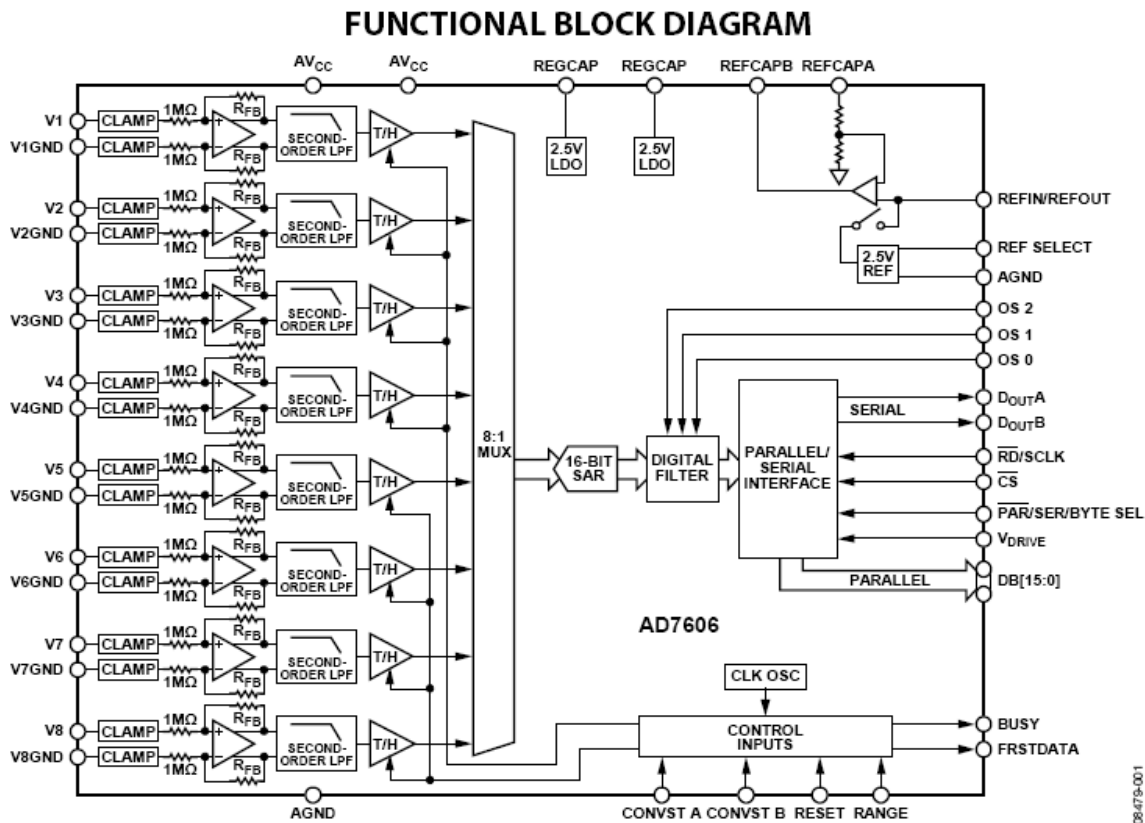
Applications

- Feed back element in high performance gradient amplifiers for MRI
- Feed back element in precision current regulated devices (power supplies...)
- Calibration unit
- Precision and high-stability inverters
- Energy measurement
- Medical equipment.

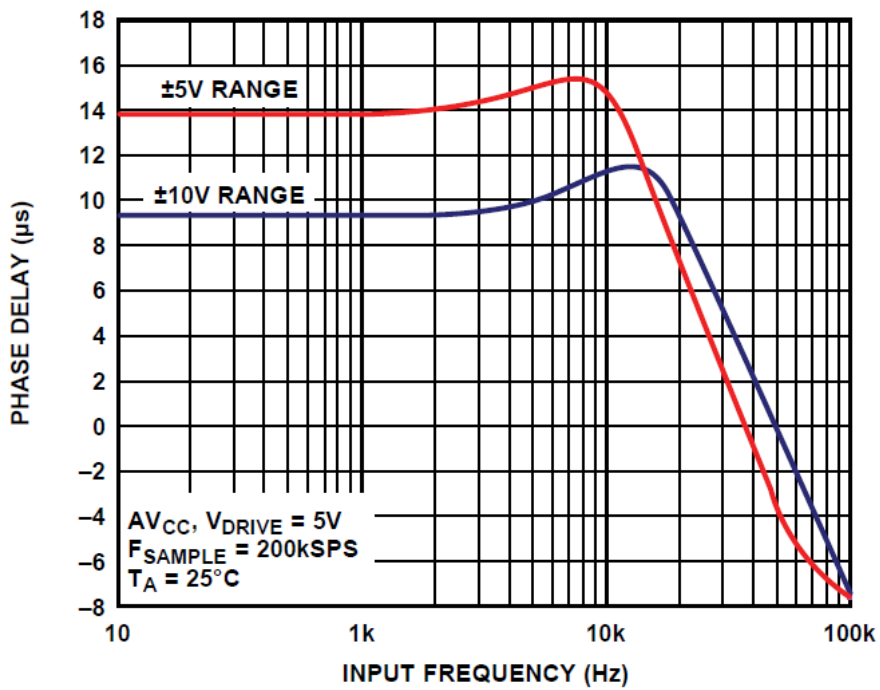
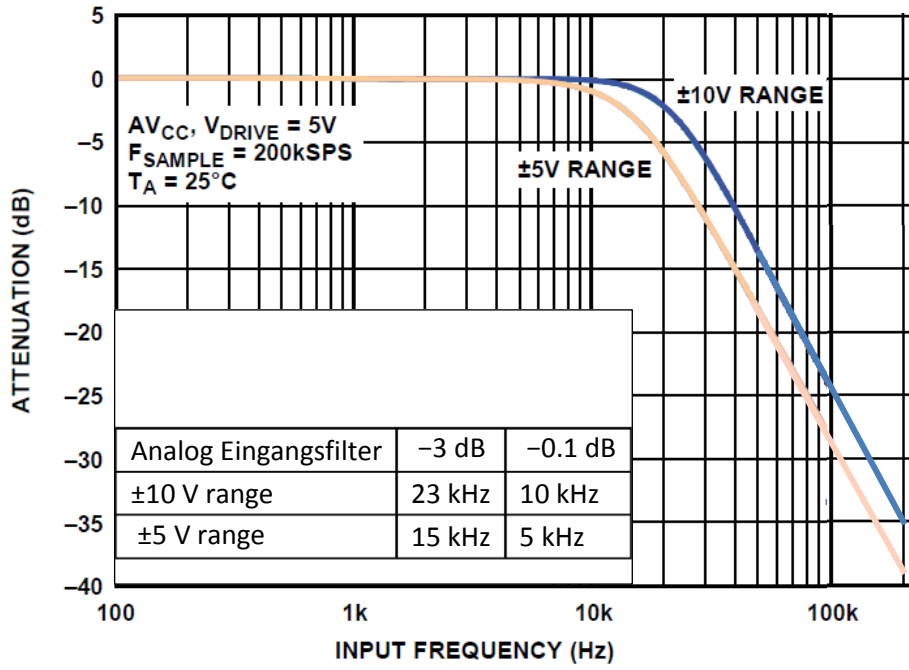
Application domain

- Industrial and Medical.

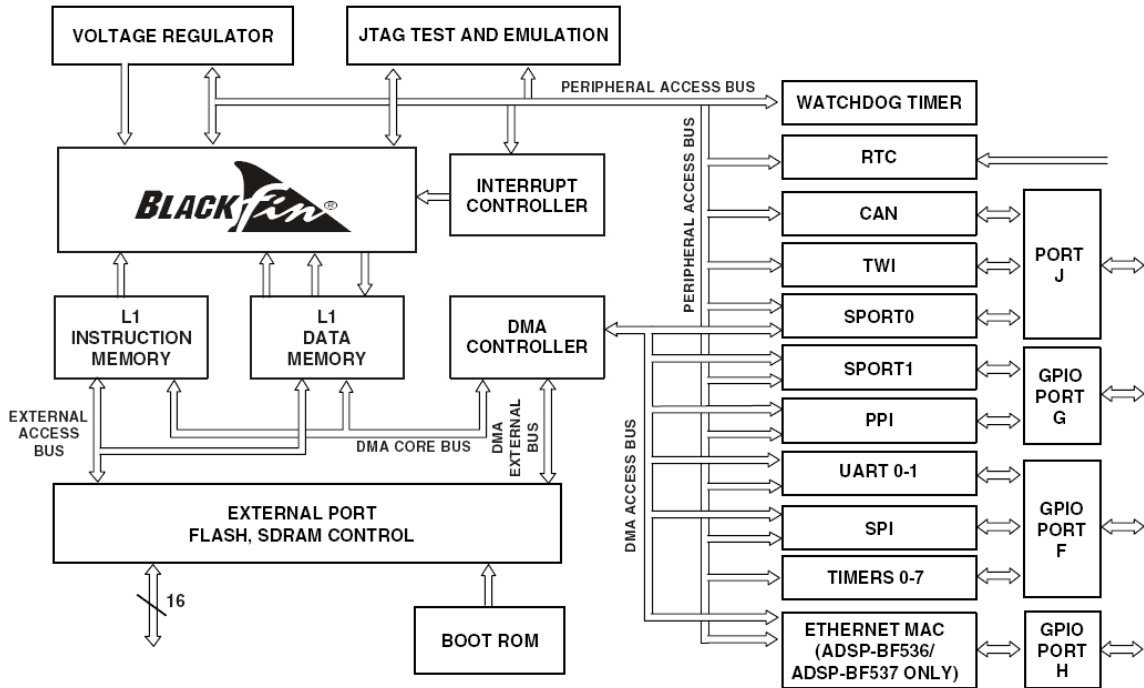
Anlage 9: A/D-Wandler AD7606



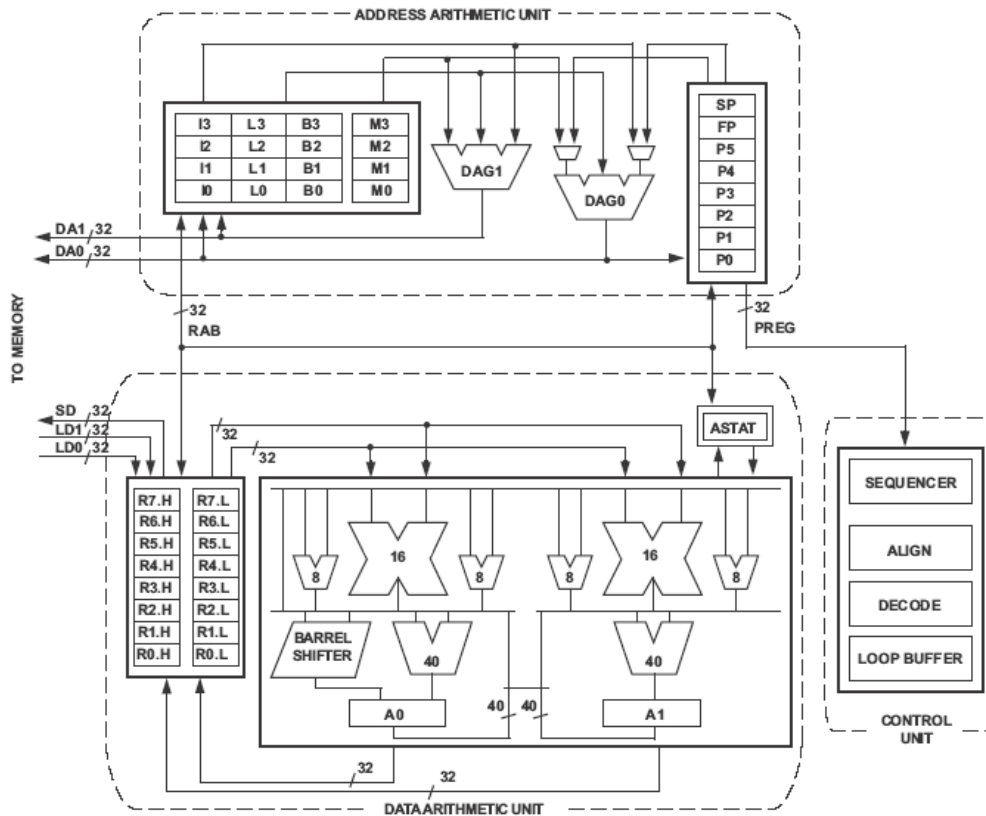
The AD7606/AD7606-6/AD7606-4 operate from a single 5 V supply and can accommodate ± 10 V and ± 5 V true bipolar input signals while sampling at throughput rates up to 200 kSPS for all channels. The input clamp protection circuitry can tolerate voltages up to ± 16.5 V. The AD7606 has 1 M Ω analog input impedance regardless of sampling frequency. The single supply operation, on-chip filtering, and high input impedance eliminate the need for driver op amps and external bipolar supplies. The AD7606/AD7606-6/AD7606-4 antialiasing filter has a 3 dB cutoff frequency of 22 kHz and provides 40 dB antialias rejection when sampling at 200 kSPS. The flexible digital filter is pin driven, yields improvements in SNR, and reduces the 3 dB bandwidth.



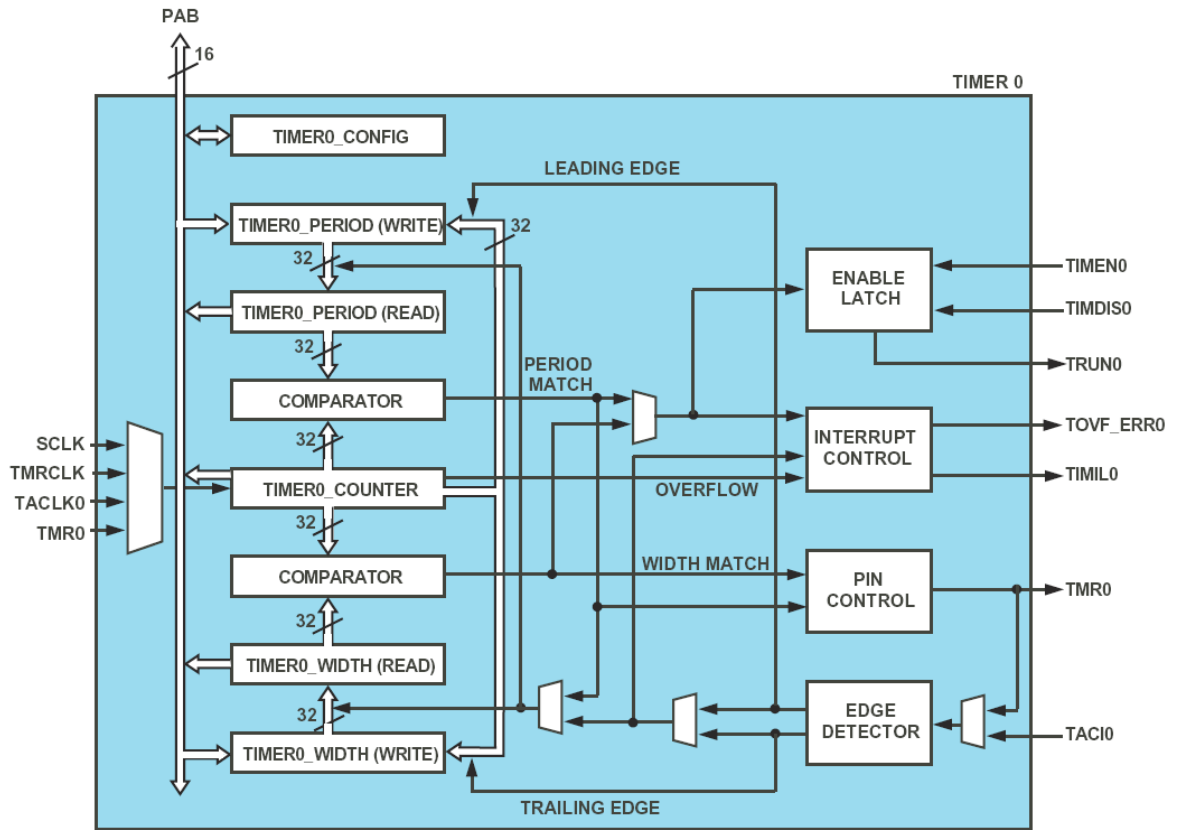
Anlage 10: DSP Blackfin ADSP BF537 von Analog Devices



Blockschaltbild des ADSP BF357



Architektur des Prozessorkerns



Interne Struktur eines Timers

Anlage 11: Quellecode des gesamten Programms

arctan.h

```
// **** Funktions-Prototypen ****
void arctan_tabelle_anlegen(void);           // Initialisierungs-Routinen
unsigned short trigo_arctan(signed short, signed short); // Zahler und Nenner

// **** Arctan Tabellen ****
static unsigned short arctan_tabelle_1[32768];
static unsigned short arctan_tabelle_2[16384];
static unsigned short arctan_tabelle_3[8192];
static unsigned short arctan_tabelle_4[4096];
static unsigned short arctan_tabelle_5[2048];
static unsigned short arctan_tabelle_6[1024];
static unsigned short arctan_tabelle_7[512];
static unsigned short arctan_tabelle_8[256];
static unsigned short arctan_tabelle_9[128];
static unsigned short arctan_tabelle_10[64];
static unsigned short arctan_tabelle_11[32];
static unsigned short arctan_tabelle_12[16];
static unsigned short arctan_tabelle_13[8];
```

arctan.c

```
/*
 * arctan.c
 */
#include <math.h>
#include "div_tabelle.h"
#include "log2_tabelle.h"
#include "arctan.h"

// schnelle Berechnung des Phasenwinkels einer komplexen Zahl
// Eingang: (signed short) Realanteil (hier nenner) und Imaginaeranteil (hier
// zahler)
// Ausgang: (unsigned short) Phasenwinkel in Q1.15: 2pi <=> 65535
// Berechnung der Division und Arcustangent anhand Tabellen

unsigned short trigo_arctan(signed short zahler, signed short nenner)
{
    int  ergebn_div, abs_ergebn_div, nenner_temp;
    unsigned short arctan_tabelle_nummer, korr2pi, winkel, winkel2pi, tab_adress,
    log2_adress;

    // div_liste ablesen --> nenner_temp = 1/nenner , format: bit 31 - 16,
    // 1/nenner: bit 15 - 0
    nenner_temp = div_liste[(unsigned short)nenner];

    // ergebn_div in Q15.14, Multiplikation mit 1/nenner = Division, format ein-
    // stellen = verschieben
    ergebn_div = (zahler * (signed short)nenner_temp) >> ((nenner_temp >> 16) -
    14);

    // Betrag weil arctan_tabelle von 0 bis +unendlich
    abs_ergebn_div = abs(ergebn_div);

    // adress für log2_tabelle einstellen: 0 bis 2^15 <=> 0, 2^15 bis 2^16 <=> 1,
    // 2^16 bis 2^17 <=> 2 ....
    log2_adress = abs_ergebn_div >> 15;

    // log2_liste ablesen --> nummer der arctan Tabelle
    arctan_tabelle_nummer = log2_liste [log2_adress];

    // winkel in Q1.15 ablesen, 0 bis pi/2 <=> 0 bis 16384
```

```

switch (arctan_tabelle_nummer)
{
    case 0:    winkel = arctan_tabelle_1[abs_ergeb_div]; break;
    case 1:    tab_adress = (abs_ergeb_div >> 1) - 16384; winkel = ar-
ctan_tabelle_2[tab_adress]; break;
    case 2:    tab_adress = (abs_ergeb_div >> 3) - 8192; winkel = ar-
ctan_tabelle_3[tab_adress]; break;
    case 3:    tab_adress = (abs_ergeb_div >> 5) - 4096; winkel = ar-
ctan_tabelle_4[tab_adress]; break;
    case 4:    tab_adress = (abs_ergeb_div >> 7) - 2048; winkel = ar-
ctan_tabelle_5[tab_adress]; break;
    case 5:    tab_adress = (abs_ergeb_div >> 9) - 1024; winkel = ar-
ctan_tabelle_6[tab_adress]; break;
    case 6:    tab_adress = (abs_ergeb_div >> 11) - 512; winkel = ar-
ctan_tabelle_7[tab_adress]; break;
    case 7:    tab_adress = (abs_ergeb_div >> 13) - 256; winkel = ar-
ctan_tabelle_8[tab_adress]; break;
    case 8:    tab_adress = (abs_ergeb_div >> 15) - 128; winkel = ar-
ctan_tabelle_9[tab_adress]; break;
    case 9:    tab_adress = (abs_ergeb_div >> 17) - 64; winkel = ar-
ctan_tabelle_10[tab_adress]; break;
    case 10:   tab_adress = (abs_ergeb_div >> 19) - 32; winkel = ar-
ctan_tabelle_11[tab_adress]; break;
    case 11:   tab_adress = (abs_ergeb_div >> 21) - 16; winkel = ar-
ctan_tabelle_12[tab_adress]; break;
    case 12:   tab_adress = (abs_ergeb_div >> 23) - 8; winkel = ar-
ctan_tabelle_13[tab_adress]; break;
    case 13:   winkel = 16383; break;
    case 14:   winkel = 16383; break;
    case 15:   winkel = 16384; break;
    default:   winkel=0;
}

// -*** korrektur: winkel: 0 bis pi/2 --> winkel2pi: 0 bis 2pi ***
korr2pi = (nenner<0) + (zahler<0) + (nenner>=0 && zahler<0)*2;

switch (korr2pi)
{
    case 1:    winkel2pi = 32768 - winkel; break;
    case 2:    winkel2pi = 32768 + winkel; break;
    case 3:    winkel2pi = 65536 - winkel; break;
    default:   winkel2pi = winkel;
}

return winkel2pi;
}

void arctan_tabelle_anlegen (void)
{
    int j;
    float tab_start=32768;

    // -**** arctan_tabelle_1 anlegen ****
    for (j=0; j<=32767; j++)
    {
        arctan_tabelle_1[j] = (unsigned short)((atan((float)j/16384)) * 32768 /
3.141592653589793238462643383279 + 0.5);
    }

    // -**** arctan_tabelle_2 anlegen ****

```

```

for (j=0; j<=16383; j++)
{
    arctan_tabelle_2[j] = (unsigned
short)((atan(((float)j*2+tab_start)/16384)) * 32768 /
3.141592653589793238462643383279 + 0.5);
}

// -**** arctan_tabelle_3 anlegen ****
for (j=0; j<=8191; j++)
{
    arctan_tabelle_3[j] = (unsigned
short)((atan(((float)j*8+tab_start*2)/16384)) * 32768 /
3.141592653589793238462643383279 + 0.5);
}

// -**** arctan_tabelle_4 anlegen ****
for (j=0; j<=4095; j++)
{
    arctan_tabelle_4[j] = (unsigned
short)((atan(((float)j*32+tab_start*4)/16384)) * 32768 /
3.141592653589793238462643383279 + 0.5);
}

// -**** arctan_tabelle_5 anlegen ****
for (j=0; j<=2047; j++)
{
    arctan_tabelle_5[j] = (unsigned
short)((atan(((float)j*128+tab_start*8)/16384)) * 32768 /
3.141592653589793238462643383279 + 0.5);
}

// -**** arctan_tabelle_6 anlegen ****
for (j=0; j<=1023; j++)
{
    arctan_tabelle_6[j] = (unsigned
short)((atan(((float)j*512+tab_start*16)/16384)) * 32768 /
3.141592653589793238462643383279 + 0.5);
}

// -**** arctan_tabelle_7 anlegen ****
for (j=0; j<=511; j++)
{
    arctan_tabelle_7[j] = (unsigned
short)((atan(((float)j*2048+tab_start*32)/16384)) * 32768 /
3.141592653589793238462643383279 + 0.5);
}

// -**** arctan_tabelle_8 anlegen ****
for (j=0; j<=255; j++)
{
    arctan_tabelle_8[j] = (unsigned
short)((atan(((float)j*8192+tab_start*64)/16384)) * 32768 /
3.141592653589793238462643383279 + 0.5);
}

// -**** arctan_tabelle_9 anlegen ****
for (j=0; j<=127; j++)
{
    arctan_tabelle_9[j] = (unsigned
short)((atan(((float)j*32768+tab_start*128)/16384)) * 32768 /
3.141592653589793238462643383279 + 0.5);
}

```



```

    }

// -**** arctan_tabelle_10 anlegen ****
for (j=0; j<=63; j++)
{
    arctan_tabelle_10[j] = (unsigned
short) ((atan(((float)j*131072+tab_start*256)/16384)) * 32768 /
3.141592653589793238462643383279 + 0.5);

}

// -**** arctan_tabelle_11 anlegen ****
for (j=0; j<=31; j++)
{
    arctan_tabelle_11[j] = (unsigned
short) ((atan(((float)j*524288+tab_start*512)/16384)) * 32768 /
3.141592653589793238462643383279 + 0.5);

}

// -**** arctan_tabelle_12 anlegen ****
for (j=0; j<=15; j++)
{
    arctan_tabelle_12[j] = (unsigned
short) ((atan(((float)j*2097152+tab_start*1024)/16384)) * 32768 /
3.141592653589793238462643383279 + 0.5);

}

// -**** arctan_tabelle_13 anlegen ****
for (j=0; j<=7; j++)
{
    arctan_tabelle_13[j] = (unsigned
short) ((atan(((float)j*8388608+tab_start*2056)/16384)) * 32768 /
3.141592653589793238462643383279 + 0.5);

}

}

```

clarktransformation.h

```

#ifndef _CLARKTRANSFORMATION_H_
#define _CLARKTRANSFORMATION_H_

void clarktransformation_unsymmetrisch(_stDatenInDreiAchsen
*, _stDatenInFesteAchsen *);
void clarktransformation_unsymmetrisch_ohne_theta(_stDatenInDreiAchsen *,
_stDatenInFesteAchsen *);

#endif

```

clarktransformation.c

```

/*****
Funtionsname Name: Clarktransformation
C-File-Name: clarktransformation.c
*****/

#include "arctan.h"
#include "main.h"

void clarktransformation_unsymmetrisch(_stDatenInDreiAchsen *eingabewerte,
_stDatenInFesteAchsen *ausgabewerte)
{
    ausgabewerte->alpha = ((eingabewerte->u_phas * ZWEIDRITTEL) >> 15)

```

```

        - ((eingabewerte->v_phas * EINDRITTEL ) >> 16)
        - ((eingabewerte->w_phas * EINDRITTEL ) >> 16);

    if (ausgabewerte->alpha > 32767){ausgabewerte->alpha = 32767;}           //
    Begrenzung fuer arctan berechnung
    else {if (ausgabewerte->alpha < -32768) ausgabewerte->alpha = -32768;}

    ausgabewerte->beta = ((eingabewerte->v_phas * EINS DURCHSQRT3) >> 15)
        - ((eingabewerte->w_phas * EINS DURCHSQRT3) >> 15);

    if (ausgabewerte->beta > 32767){ausgabewerte->beta = 32767;}           //
    Begrenzung fuer arctan berechnung
    else {if (ausgabewerte->beta < -32768) ausgabewerte->beta = -32768;}

    ausgabewerte->theta = trigo_arctan(ausgabewerte->beta, ausgabewerte->alpha);
}

void clarktransformation_unsymmetrisch_ohne_theta(_stDatenInDreiAchsen
*eingabewerte, _stDatenInFesteAchsen *ausgabewerte)
{
    ausgabewerte->alpha = ((eingabewerte->u_phas * ZWEIDRITTEL) >> 15)
        - ((eingabewerte->v_phas * EINDRITTEL ) >> 16)
        - ((eingabewerte->w_phas * EINDRITTEL ) >> 16);

    ausgabewerte->beta = ((eingabewerte->v_phas * EINS DURCHSQRT3) >> 15)
        - ((eingabewerte->w_phas * EINS DURCHSQRT3) >> 15);
}

/** EOF */

```

DFT.h

```

#ifndef _DFT_H_
#define _DFT_H_

// ***** Funktions-Prototypen *****
void DFT (void);
void Filter50Hz (int *, int *, int *, _stOS *, int *, int *);
void DFT_5Hz_RIAP (int *, int *, _stOS *, int *, int *);
void Phasenwinkel_Korrektur (    short *,
                                short *,
                                int *,
                                unsigned short *,
                                unsigned short *,
                                unsigned int *,
                                short *,
                                short *,
                                int *,
                                int *,
                                unsigned short *);

#endif

```

DFT.c

```

#include "main.h"
#include "cos_tabelle.h"
#include "sin_tabelle.h"
#include "DFT.h"
#include "arctan.h"
#include "PID_Regler.h"

extern unsigned int I_soll_start;
extern unsigned short ordnung ; //Ordnung der Oberschwingung
extern int OS_system; // Mitsystem = 1, Gegensystem = -1

//*****
// Messwert Speicherung

```

```

extern _stDatenInDreiAchsen      stDreiPhasenNetzSpannung;
extern _stDatenInDreiAchsen      stDreiPhasenNetzStrom;
extern int Strom_U [16384], Strom_V [16384], Strom_W [16384];
extern int Spg_U [16384], Spg_V [16384], Spg_W [16384];
extern int Spg_U_ohne_50Hz[16384], Spg_V_ohne_50Hz[16384], Spg_W_ohne_50Hz[16384];
extern unsigned short ztime;

//*****
//50Hz DFT
extern _stOS OS_Strom_U_50Hz;
extern unsigned short ztime50Hz;
extern int cos_winkelOS1p, sin_winkelOS1p, cos_winkelOS1, sin_winkelOS1;
extern unsigned short winkelOS1, winkelOS1p;
extern int Regler_OS1_alpha, Regler_OS1_beta;
extern _stPID_Regler      stPID_Regler_OS1_D;
extern _stPID_Regler      stPID_Regler_OS1_Q;

//*****
// 5Hz DFT
extern _stOS OS_Spg_U_5Hz[2], OS_Spg_V_5Hz[2], OS_Spg_W_5Hz[2];;
extern unsigned short ztime5Hz;

//*****
// Phasenwinkel Korrektur
extern int Totzeiten;
extern unsigned short ztime_mittelw, ztime_alt_mittelw;
extern short Kreisfrequenz_U[4096], Kreisfrequenz_V[4096], Kreisfrequenz_W[4096];
extern int Kreisfrequenz_U_sum, Kreisfrequenz_V_sum, Kreisfrequenz_W_sum;
extern unsigned short Phase_U_alt, Phase_V_alt, Phase_W_alt;
extern unsigned int Phase_U_mittelw, Phase_V_mittelw, Phase_W_mittelw;
extern short delta_Phase_U[4096], delta_Phase_V[4096], delta_Phase_W[4096];
extern int delta_Phase_U_sum, delta_Phase_V_sum, delta_Phase_W_sum, Phasenwinkel_totzeit;
extern unsigned short Phase_U_ok, Phase_V_ok, Phase_W_ok;

//*****
void DFT (void)
{
// ztime = Zeiger für die Messwert Speicherung
  if( ztime >= 16383 ) { ztime=0; }
  else { ztime++; }

// *** Messwert Speicherung ***
  Strom_U [ztime] = stDreiPhasenNetzStrom.u_phas;
  Strom_V [ztime] = stDreiPhasenNetzStrom.v_phas;
  Strom_W [ztime] = stDreiPhasenNetzStrom.w_phas;
  Spg_U [ztime] = stDreiPhasenNetzSpannung.u_phas;
  Spg_V [ztime] = stDreiPhasenNetzSpannung.v_phas;
  Spg_W [ztime] = stDreiPhasenNetzSpannung.w_phas;

// *****
// *          DFT 50Hz          *
// *****

  unsigned short zwinkel50Hz = (ztime50Hz * SKAL_WINKEL50Hz) >> 15 ;
  unsigned short ztime_alt_50Hz = (unsigned short)((ztime << 2) - 800) >> 2;

// Regelung Strom OS

  winkelOS1 = ordnung * zwinkel50Hz;

  cos_winkelOS1 = cos_liste[winkelOS1];
  sin_winkelOS1 = sin_liste[winkelOS1];

  OS_Strom_U_50Hz.Re += ((Strom_U[ztime] * sin_winkelOS1) >> 15)

```

```

        - ((Strom_U[ztime_alt_50Hz] * sin_winkelOS1) >> 15);
OS_Strom_U_50Hz.Im += ((Strom_U[ztime] * cos_winkelOS1) >> 15)
        - ((Strom_U[ztime_alt_50Hz] * cos_winkelOS1) >> 15);

if (I_soll_start > 10000)
{
    if(OS_system >= 0)
    {
        stPID_Regler_OS1_D.istwert = ((OS_Strom_U_50Hz.Re >> 4) * SKAL_ReIm50Hz) >>
17;
        stPID_Regler_OS1_Q.istwert = ((OS_Strom_U_50Hz.Im >> 4) * SKAL_ReIm50Hz) >>
17;

        PID_regler(&stPID_Regler_OS1_D);
        PID_regler(&stPID_Regler_OS1_Q);

        Regler_OS1_alpha = ((cos_winkelOS1 * stPID_Regler_OS1_D.stellwert) >> 15)
            - ((sin_winkelOS1 * stPID_Regler_OS1_Q.stellwert) >> 15);

        Regler_OS1_beta = ((cos_winkelOS1 * stPID_Regler_OS1_Q.stellwert) >> 15)
            + ((sin_winkelOS1 * stPID_Regler_OS1_D.stellwert) >> 15);
    }
    else
    {
        stPID_Regler_OS1_D.istwert = ((OS_Strom_U_50Hz.Re >> 4) * SKAL_ReIm50Hz) >>
17;
        stPID_Regler_OS1_Q.istwert = -((OS_Strom_U_50Hz.Im >> 4) * SKAL_ReIm50Hz) >>
17;

        PID_regler(&stPID_Regler_OS1_D);
        PID_regler(&stPID_Regler_OS1_Q);

        Regler_OS1_alpha = ((cos_winkelOS1 * stPID_Regler_OS1_D.stellwert) >> 15)
            + ((sin_winkelOS1 * stPID_Regler_OS1_Q.stellwert) >> 15);

        Regler_OS1_beta = ((cos_winkelOS1 * stPID_Regler_OS1_Q.stellwert) >> 15)
            - ((sin_winkelOS1 * stPID_Regler_OS1_D.stellwert) >> 15);
    }
}

if (ztime50Hz >= 199) { ztime50Hz = 0;}
else { ztime50Hz ++ ;}

// *****
// *          DFT 5Hz          *
// *****

// je 200ms: ztime5Hz = 0 ... 1999 --> zwinkel5Hz = 0 ... 65535
unsigned short zwinkel5Hz = (ztime5Hz * SKAL_WINKEL5Hz) >> 15;

// adress vor 200ms in Speicher mit 16384 Werte
unsigned short ztime_alt_5Hz = (unsigned short)(ztime << 2) - 8000) >> 2;

// *** 50Hz Filterung ***

int cos_winkel50Hz = cos_liste[zwinkel50Hz];
int sin_winkel50Hz = sin_liste[zwinkel50Hz];

    Filter50Hz (&Spg_U_ohne_50Hz[ztime], &Spg_U[ztime], &Spg_U[ztime_alt_5Hz],
&OS_Spg_U_5Hz[0], &cos_winkel50Hz, &sin_winkel50Hz);
    Filter50Hz (&Spg_V_ohne_50Hz[ztime], &Spg_V[ztime], &Spg_V[ztime_alt_5Hz],
&OS_Spg_V_5Hz[0], &cos_winkel50Hz, &sin_winkel50Hz);

```

```

    Filter50Hz (&Spg_W_ohne_50Hz[ztime], &Spg_W[ztime], &Spg_W[ztime_alt_5Hz],
&OS_Spg_W_5Hz[0], &cos_winkel50Hz, &sin_winkel50Hz);

// *** Messung OS Netzspannung ***

    unsigned short winkel_OS_spg = ordnung * 10 * zwinkel5Hz;

    int cos_winkel_OS_spg = cos_liste[winkel_OS_spg];
    int sin_winkel_OS_spg = sin_liste[winkel_OS_spg];

    DFT_5Hz_RIAP (&Spg_U_ohne_50Hz[ztime], &Spg_U_ohne_50Hz[ztime_alt_5Hz],
&OS_Spg_U_5Hz[1], &cos_winkel_OS_spg, &sin_winkel_OS_spg);
    DFT_5Hz_RIAP (&Spg_V_ohne_50Hz[ztime], &Spg_V_ohne_50Hz[ztime_alt_5Hz],
&OS_Spg_V_5Hz[1], &cos_winkel_OS_spg, &sin_winkel_OS_spg);
    DFT_5Hz_RIAP (&Spg_W_ohne_50Hz[ztime], &Spg_W_ohne_50Hz[ztime_alt_5Hz],
&OS_Spg_W_5Hz[1], &cos_winkel_OS_spg, &sin_winkel_OS_spg);

    if( ztime5Hz >= 1999 ) { ztime5Hz=0; }
    else { ztime5Hz++; }

// Korrektur der Phasenwinkel

ztime_mittelw = (ztime & 0xFFF); //ztime_mittelw: von 0 bis 4096
ztime_alt_mittelw = ((ztime - 2048) & 0xFFF);

    Phasenwinkel_Korrektur ( &Kreisfrequenz_U[ztime_mittelw],
&Kreisfrequenz_U[ztime_alt_mittelw],
&Kreisfrequenz_U_sum,
&OS_Spg_U_5Hz[1].phas,
&Phase_U_alt,
&Phase_U_mittelw,
&delta_Phase_U[ztime_mittelw],
&delta_Phase_U[ztime_alt_mittelw],
&delta_Phase_U_sum,
&Totzeiten,
&Phase_U_ok);

Phase_U_alt = OS_Spg_U_5Hz[1].phas;
}

//*****
// Funktion 5Hz DFT mit Amplitude und Phase Berechnung

void DFT_5Hz_RIAP (int *Spg_ztime, int *Spg_ztimealt, _stOS *Complex5Hz, int
*cos_winkel, int *sin_winkel)
{
    Complex5Hz->Re += ((*Spg_ztime * *sin_winkel) >> 15)
- ((*Spg_ztimealt * *sin_winkel) >> 15);

    Complex5Hz->Im += ((*Spg_ztime * *cos_winkel) >> 15)
- ((*Spg_ztimealt * *cos_winkel) >> 15);

    // Wertebereich OS Netzspg nach Messung = 2,5V ... 0,05V
    // Max OS = 8V Qn.6<=>Re_skal und Spg Q10.6 Spg_OS in Q4.12 => 8V <=>
32768
    int Re_skal = ((Complex5Hz->Re >> 4) * SKAL_ReIm5Hz) >> 15; //
Qn.6<=>Re_skal = (OS.Re * 2) / 2000
    int Im_skal = ((Complex5Hz->Im >> 4) * SKAL_ReIm5Hz) >> 15; //
Qn.6<=>Im_skal = (OS.Im * 2) / 2000

    Complex5Hz->ampl = (sqrt((long long)Re_skal*Re_skal + (long
long)Im_skal*Im_skal));
    Complex5Hz->phas = trigo_arctan(Im_skal, Re_skal);
}

//*****
// Funktion Filterung der 50Hz Anteil der Netzspannung mit 5Hz DFT

```

```

void Filter50Hz (int *Spg_filter, int *Spg_ztime, int *Spg_ztimealt, _stOS
*Complex5Hz, int *cos_winkel, int *sin_winkel)
{
    Complex5Hz->Re += ((*Spg_ztime * *sin_winkel) >> 15)
                    - ((*Spg_ztimealt * *sin_winkel) >> 15);

    Complex5Hz->Im += ((*Spg_ztime * *cos_winkel) >> 15)
                    - ((*Spg_ztimealt * *cos_winkel) >> 15);

    int Re_skal = ((Complex5Hz->Re >> 9) * SKAL_ReIm5Hz) >> 15;    // Re_skal =
(OS.Re * 2) / 2000
    int Im_skal = ((Complex5Hz->Im >> 9) * SKAL_ReIm5Hz) >> 15;    // Im_skal =
(OS.Im * 2) / 2000

    int Spg50Hz = ((Im_skal * *cos_winkel) >> 15)
                  + ((Re_skal * *sin_winkel) >> 15);

    *Spg_filter = *Spg_ztime - Spg50Hz;    // Auslöschen der 50Hz Grund-
schwingung
}

//*****
// Funktion Phasenwinkel Totzeiten und Filterung

void Phasenwinkel_Korrektur (    short *Kreisfreq,
                                short *Kreisfreq_alt,
                                int *Kreisfreq_sum,
                                unsigned short *Phasenwinkel,
                                unsigned short *Phasenwinkel_alt,
                                unsigned int *Phasenwinkel_mittelw,
                                short *delta_Phasenwinkel,
                                short *delta_Phasenwinkel_alt,
                                int *delta_Phasenwinkel_sum,
                                int *totzeit,
                                unsigned short *Phasenwinkel_super)
{
    *Kreisfreq = *Phasenwinkel - *Phasenwinkel_alt;
    *Kreisfreq_sum += *Kreisfreq - *Kreisfreq_alt;
    *Phasenwinkel_mittelw += (*Kreisfreq_sum << 5);
    Phasenwinkel_totzeit = (*Kreisfreq_sum * *totzeit) >> 11;
    *delta_Phasenwinkel = *Phasenwinkel + Phasenwinkel_totzeit -
(*Phasenwinkel_mittelw >> 16);
    *delta_Phasenwinkel_sum += *delta_Phasenwinkel - *delta_Phasenwinkel_alt;
    *Phasenwinkel_super = (*delta_Phasenwinkel_sum >> 11) +
(*Phasenwinkel_mittelw >> 16);
}

```

inbetriebnahme.h

```

#ifndef _INBETRIEBNAHME_H_
#define _INBETRIEBNAHME_H_

int trafouebersetzung(_stSystemParameter *);

void inbetriebnehmen(void);

#endif

/* EOF */

```

inbetriebnahme.c

```

#define UZK_MINIMUM 1
#define LEITERSTROEME_MAX

#include "main.h"
#include <math.h>

```

```

extern      struct
{
    short kanal1; /*Rohdaten Spannung U*/
    short kanal2; /*Rohdaten Spannung V*/
    short kanal3; /*Rohdaten Spannung W*/
    short kanal4; /*noch Frei*/
    short kanal5; /*Rohdaten Spannung ZK*/
    short kanal6; /*Rohdaten Strom W*/
    short kanal7; /*Rohdaten Strom V*/
    short kanal8; /*Rohdaten Strom U*/
    short update; /*Daten werden Verarbeitet und in eigene Struktur geschrieben.
                                                    Wert wird auf 0 Gesetzt wenn dies
                                                    geschehen ist. Nach neuer
                                                    Wandlung wird wieder auf 1 gesetzt*/
} stADCCDaten;

extern _stDatenInDreiAchsen      stDreiPhasenNetzSpannung;
extern _stDatenInFesteAchsen     stAlphaBetaNetzSpannung;
extern _stDatenInDreiAchsen      stDreiPhasenNetzStrom;
extern _stDatenInFesteAchsen     stAlphaBetaNetzStrom;
extern _stSystemParameter        stSystemParameter;

/*****
Eingabewerte werden aus der Messwerttabelle entnommen
Messwerttabelle ist Global
Auswertung von Spannungswandler 1 und 3
Rückgabewert -1 = Fehler; 0 und Größer ist das
Übersetzungsverhältnis
*****/

int trafoubersetzung(_stSystemParameter *eingabewerte){

    unsigned int x=1; /*Variable für whileschleife x = 0 whileschleife Ende*/
    unsigned int y=1; /*Schleifenkontrollvariable; Wenn schleife zu oft durchlaufen abbruch*/
    unsigned int z=0, zz=0; /*Werte für Arrayposition zum Speichern der Messwerte*/
    unsigned int trafoSchreiben=0, wrSchreiben=0, z_increment=0, ruecksetzfreigabe=0; /*Kontrollvariablen*/
    int trafomesswerte[20][2]; /*[Anzahl der Messwerte] [1 Trafo und 2 Wechselrichter]*/
    int trafonetz_k1=0, trafonetz_k=0; /*Vorgängerwert Netzspannung, neuer Wert Netzspannung*/
    int trafoWR_k1=0, trafoWR_k=0; /*Vorgängerwert Wechselrichterseite, neuer Wert Wechselrichterseite*/
    int rueckgabewert=0; /*Rückgabewert der Funktion -1 Fehler, und Werte >= 0 Übersetzungsverhältniss*/

    int i;
    unsigned int netzzeiger=0, wrzeiger=0;

    /*-----BEGINN SCHLEIFE MESSWERTE EINLESEN-----*/

    while(x){
        ADC_read();
        if(stADCCDaten.update==1) /*Wenn neue Messwerte vom Wandler vorliegen globale Variable auslesen*/
        {
            stADCCDaten.update =0;
            /*Globale Variable Messwerte wieder auf 0 setzen*/
            trafonetz_k1 = trafonetz_k; /*Alte Messwerte sichern*/
            trafoWR_k1 = trafoWR_k;
            /*-----*/
            /*Neue Messwerte einlesen*/

```

```

trafonetz_k = stADCDaten.kanal1 /*Globale Variable U Kanal
1*/;
trafoWR_k = stADCDaten.kanal3 /*Globale Variable U Kanal
3*/;
/*-----*/

/*****Scheitelwerte abspeichern*****/

/* Scheitelwert der Netzspannung speichern*/
if(trafonetz_k1>trafonetz_k && trafoSchreiben == 1)
{
    if (trafonetz_k1 > 0)
    {
        trafoSchreiben = 0;
        z_increment = 1;
        trafomesswerte[z][0]=trafonetz_k1;
    }
}

/* Scheitelwert hinter dem Trafo auf Wechselrichterseite spei-
chern*/
if(trafoWR_k1>trafoWR_k && wrSchreiben == 1)
{
    if (trafoWR_k1 > 0)
    {
        wrSchreiben = 0;
        trafomesswerte[z][1]=trafoWR_k1;
    }
}

/*Wurden beide Scheitelwerte erfasst Array auf nächstes Element
zeigen lassen*/
if(trafoSchreiben==0 && wrSchreiben==0 && z_increment == 1)
{
    z_increment = 0;
    z++;
    zz++; /* Insgesamt 25 Durchläufe, die ersten 5 Werte
werden verworfen*/
    if (z==20) z=0;
}

/*Wenn der Sinus wieder eine positiven gradienden Aufweist für
neue Messwerterfassung freigeben*/
if(trafonetz_k>trafonetz_k1 && trafoWR_k>trafoWR_k1)
{
    if(ruecksetzfreigabe==30)
    {
        trafoSchreiben = 1;
        wrSchreiben = 1;
        ruecksetzfreigabe=0;
    }
    ruecksetzfreigabe++;
}

}

/*Wenn 25 Messungen durchgeführt wurden und das Array gefüllt ist while-
Schleife beenden*/
if(zz==24) x=0;

/*Beenden bei Zeitüberschreitung*/
if(y++==467295)
{
    x = 0; /*Schleifenabbruch wenn Zeitüberschreitung
bei Messung*/
    rueckgabewert = -1; /*Fehler mit den Timern. Messung wurde nicht
durchgeführt*/
}

```



```

    }
    /*-----ENDE SCHLEIFE MESSWERTE EINSAMMELN-----*/

    /**Auswertung der Messergebnisse und berechnen des Übersetzungsverhältnis-
    ses***/
    if(rueckgabewert != -1) /*Wenn nicht wegen Zeitüberschreitung abgebrochen wurde*/
    {
        for (i=0;i<20;i++)
        {
            netzzeiger += trafomesswerte[i][0]; /*kein Überlauf möglich Ma-
            ximalwert 1310720*/
            wrzeiger += trafomesswerte[i][1]; /*kein Überlauf möglich Ma-
            ximalwert 1310720*/
        }
        netzzeiger = netzzeiger/20;
        wrzeiger = wrzeiger/20;

        rueckgabewert = (wrzeiger << 15) / netzzeiger; /*Idealwert 22937 = 0,7
    Q1.15*/

        eingabewerte->netzzeiger = netzzeiger;
        eingabewerte->trafozeiger = wrzeiger;
        eingabewerte->uebersetzungsverhaeltniss = rueckgabewert;
    }

    return rueckgabewert;
}

/*****
Funktion inbetriebnehmen()
Alle einstellungen werden hier überprüft
*****/

void inbetriebnehmen(void){
    int i, fehlertest;
    int summe_strom_u=0, summe_strom_v=0, summe_strom_w=0; /*Maximalwert =
    819175 -> kein Überlauf*/
    /*Relay für Trafoübersetzung Zuschalten und*/
    /*warten bis dieses sicher geschlossen ist.*/
    *pPORTFIO_SET = RELAI3;
    delay_us(250000); // 250 ms warten, bis Relai
    geschalten

    trafoubersetzung(&stSystemParameter);

    /*Relay wieder auf Messstelle am Netz umschalten*/
    /*und warten bis dieses sicher geschlossen/geöffnet ist.*/
    *pPORTFIO_CLEAR = RELAI3;
    delay_us(250000); // 250 ms warten, bis Relai ge-
    schalten

    /*Auswertung des Übersetzungsverhältnisses wenn kleiner 0,685 oder größer
    0,715 dann Fehler*/
    //stSystemParameter.uebersetzungsverhaeltniss = 22705;
    if(stSystemParameter.uebersetzungsverhaeltniss < 22446 || stSystemParame-
    ter.uebersetzungsverhaeltniss > 23429)
    {
        /*Fehlerbehandlung weil Übersetzungsverhältnis nicht richtig einge-
        stellt ist.*/
        while(1);
        //asm("nop;");
    }
}

```

```

        /*Relay zum Zuschalten des Lastwiderstandes einschalten*/
        /*Warten bis dieses sicher geschlossen ist*/

        delay_us(250000);                // 250 ms warten, bis Relai
geschalten
        delay_us(250000);                // warten ohne besonderen Grund
    }

```

main.h

```

#include <sys\exception.h>
#include <cdefBF537.h>
#include <math.h>

// ***** Definitionen für RZM *****
#define WURZEL3      56756 // (/2^15)
#define WURZEL3_REZ 37837 // (/2^16)
#define T_HALBE     3000 // 3000
#define ZWEI_T      6000 // 6000

#define DELAY_MUL 100
#define LED1      PF6      // W
#define LED2      PF7      // V
#define LED3      PF8      // U
#define LED6      PF11
#define FREIGABE  PF5      // Taster PB4

// ADC-Definitionen
#define RESET      PF15
#define CS_RD      PF10
#define CONVSTA_B  PF9
#define BUSY       PF13    // Eingang!
#define RANGE      PF14
#define STBY       PF12

// DAC-Definitionen
#define SYNCR      TFS0
#define DeviceAddr 0x7
#define Bitlaenge  23      // Datenbits - 1
#define Teiler     2//61   // ergibt 400 kHz Taktrate
#define TWI_Length 3      // 3 Bytes
#define KANAL_A    0x00180000
#define KANAL_B    0x00190000

// ***** Funktions-Prototypen *****
void init(void);                // Initialisierungs-Routinen
void delay_us(unsigned long int zeit); // variable Verzögerungszeit n µs
void delay_85ns(void);         // feste Verzögerungszeit 85
ns
void RZM(int alpha, int beta); // Raumzeigermodulation
void alphabeta(int f);        // Erzeugung von Us_alpha und
Us_beta
void ADC_read(void);          // ADC auslesen
void set_ADC_range(unsigned char v_range); // Eingangsspannungsbereich wählen
void DAC_A(int);             // Daten an DAC-Kanal A senden
void DAC_B(int);             // Daten an DAC-Kanal B senden

// ***** Variablen *****
extern volatile char flag, ein, t_ein; // flag für Zeit-Interrupt 100 µs, "ein"
für
// Betriebszu-

stand
extern int i;

extern volatile int alpha, beta;
extern volatile int U_ein, V_ein, W_ein;

```

```

/***** N I C K *****/

#define SCHIEBEN 14          /* Realzahlen sind mit SCHIEBEN nach links ge-
schoben */
#define SQRT3 28378         /* Wurzel 3 in Qn.14 */
#define EINSHALB 8192      /* 1/2 in Qn.14 */
#define DREIHALBE 24576    /* 3/2 in Qn.14 */
#define EINDRITTEL 21845   /* 1/3 in Qn.16 */
#define ZWEIDRITTEL 21845  /* 2/3 in Qn.15 */
#define SQRT3HALBE 14189   /* Wurzel 3 halbe in Qn.14 */
#define EINS DURCH SQRT3 18919 /* Eins durch Wurzel 3 in Qn.15 */
#define ZWEI DURCH SQRT3 18919 /* Zwei durch Wurzel 3 in Qn.14 */
#define WURZEL2 DURCH 3 13377 /* Wurzel(2 / 3) in Qn.14 */
#define WURZEL3 DURCH 3 9459 /* Wurzel(3) / 3 in Qn.14 */

/***** Korrekturfaktoren & Stromgrenzwerte *****/
#define SPANNUNG_UVW 32001  /* Korrekturfaktor für die Netzspnngung 500*2^6
*/
#define SPANNUNG_UZK 24001  /* Korrekturfaktor für die Zwischenkreisspannung
=1500*2^4 */
#define STROM_UVW 25601     /* Korrekturfaktor für die Ströme 200*2^7 */
#define ITESTMAX 2000      /* Maximal möglicher Teststrom bei korrektem
Anschluss der Phasen Format Q5.11*/
#define ITESTMIN 100       /* Minimaler Strom der da sein muss um zu sagen:
Ja Wandler ist angeschlossen Q5.11*/
#define I_AUSGABE_MAX 40963 /* Strombegrenzung bei 10A Format Q5.11*/

/*****
**
Werte für Reglereinstellungen und Funktionsstruktur
*****/
#define STROM_KP 8192       /* Proportionalverstärkung in Qn.12, Kp = 1 *
4096 */
#define STROM_KI 6300      /* Integralverstärkung in Qn.4* MAXIMUM 25000
Ki = 150 * 16 */
#define STROM_KD 1         /* Differentialanteil nicht in Benutzung und
nicht programmiert*/
#define REG_ABTASTZEIT 1678 /* Abtastzeit der Regelung im Format (Q8.24) */
#define KI_mal_ABTASTZEIT 983 // [Ki] * [Tpp] = Vs/A = Henry (Induktivität) in
Qn.16 => Ki * Tpp * 2^16
#define SG_MAX 183040      /* Maximal mögliche Stellgröße = 3,75A Q5.11
7680*/
#define PA_MAX 196608      /* Begrenzung P Anteil = 6 x 2^15 */
#define IA_MAX 16777216    /* Begrenzung I Anteil = 2^24 */
#define P_REG_Y_MAX 32767 /* */
#define I_REG_Y_MAX 32767 /* */

/*****
Definition der Portausgänge der Relais, Stromüberwachung
*****/
#define RELAI1 PF3         /* Ausgang Schaltet Relay 1 EIN / AUS*/
#define RELAI2 PF2         /* Ausgang Schaltet Relay 2 EIN / AUS*/
#define RELAI3 PF4         /* Ausgang Schaltet Relay 3 EIN / AUS*/
#define MAX_STROM 1300     //1300 => 8A (ADC Strommessung: 1A =>
163,8)
#define MAX_STROM2 3300    //3300 => 20A (ADC Strommessung: 1A =>
163,8)

/*****
Definition DFT
*****/
#define SKAL_WINKEL50Hz 10737418 // = 65536/200 in Qn.15: ztime50Hz = 0
... 199 --> zwinkel50Hz = 0 ... 65535
#define SKAL_WINKEL5Hz 1073742 // = 65536/2000 in Qn.15: ztime5Hz = 0

```

```

... 1999 --> zwinkel5Hz = 0 ... 65535
#define SKAL_WINKEL1Hz 214748 // = 65536/10000 in Qn.15: ztime1Hz = 0
... 9999 --> zwinkel1Hz = 0 ... 65535
#define MESSDAUER 100000
#define SKAL_ReIm50Hz 20972 //1/100 in Qn.21 = (2^4)/100 in Qn.17:
bsp:((OS.Re >> 4) * SKAL_ReIm50Hz) >> 17
#define SKAL_ReIm5Hz 16777 //1/1000 in Qn.24 = (2^9)/1000 in Qn.15:
bsp:((OS.Re >> 9) * SKAL_ReIm5Hz) >> 15

typedef struct _stPID_Regler
{
    int e_k_minus_1; /* Vorhergehende Reglerabweichung */
    int p_anteil; /* Proportionaler Anteil des Reglers */
    int i_anteil; /* Integraler Anteil des Reglers */
    int d_anteil; /* Differentialanteil */
    int abtastzeit; /* in welchen Zeitabtaenden der Regler kon-
tinuierlich aufgerufen wird ACHTUNG! Angabe in Q38.24*/
    int kp; /* Proportionalverstaerkung */
    int ki; /* Integralverstaerkung */
    int kd; /* Differentialverstaerung */
    int sollwert; /* Sollwert wird in anderer Funktion hierher
Uebergeben*/
    int istwert; /* Muss aus Struktur fuer d und Q geholt wer-
den.*/
    int stellwert; /* Ausgabewert des Reglers im Eingabeformat*/
    int modus; /* 1=P, 2=PI, 3=PID */
} _stPID_Regler;

/* volatile int thetaSpannung; Drehzeiger der Netzspannung sollte von jeder Funk-
tion gelesen werden kÄnnten*/

/*ClarkTransformation, Eingangssignal und Ausgabe in Alpha- Beta-Koordinaten*/
typedef struct _stDatenInDreiAchsen
{
    int u_phas; /* Phase-U variable*/
    int v_phas; /* Phase-V variable*/
    int w_phas; /* Phase-W variable*/
} _stDatenInDreiAchsen;

typedef struct _stDatenInFesteAchsen
{
    int alpha; /* statische d-achsenvariable */
    int beta; /* statische q-achsenvariable */
    int gamma; /* Summe der 3 Phasen wenn Unsymmetrie */
    int theta; /* Drehwinkel des Koordinatensystems */
} _stDatenInFesteAchsen;

typedef struct _stDatenInRotierendeAchsen
{
    int d_part; /* statische d-achsenvariable */
    int q_part; /* statische q-achsenvariable */
    int null; /* Summe der 3 Phasen wenn Unsymmetrie */
    int theta; /* Winkel muss mit Uebertragen werden fuer die Rueck-
transformation */
} _stDatenInRotierendeAchsen;
/*----ENDE Strukturdefinition fuer die Funktion Clarktransformation*/

typedef struct _stSystemParameter
{
    int uebersetzungsverhaeltniss; /*Übersetzungsverhältnis des Stelltrans-
formators Format: Dezimal in % 000.*/
    int netzzeiger; /*Spitzenwert der Netzspannung
230V ist 325V */
    int trafozeiger; /*Zeiger der heruntertransfor-

```

```

mierten Netzspannung am Stelltrafo */
    int zwischenkreisspannung; /*Bezogen auf Wandler für Netz (500V)
als Integer Datenformat ist größer als 2^16 Bsp: 500V/(2^16-1)*37365=570.16V*/
    int teststrom_RMS_U;
    int teststrom_RMS_V;
    int teststrom_RMS_W;
    int regler_alpha; /*Gesamtteil der Stromregelung inklusive
Oberwellen*/
    int regler_beta; /*Gesamtteil der Stromregelung inklusive
Oberwellen*/
    int rzm_alpha; /*Eingabewert für die RZM*/
    int rzm_beta; /*Eingabewert für die RZM*/
} _stSystemParameter;

```

```

typedef struct _stOS
{
    int Re;
    int Im;
    unsigned int ampl;
    unsigned short phas;
} _stOS;

```

main.c

```

/*****
*
* Wechselrichter OS Strom Regelung
*
* Nick Meyer, Fabian Merkel, Jeremie Foulquier
*
*****/
#include <cycle_count.h>
#include "main.h"
#include "inbetriebnahme.h"
#include "clarktransformation.h"
#include "parktransformation.h"
#include "normierung.h"
#include "PID_Regler.h"
#include "arctan.h"
#include "DFT.h"
#include "cos_tabelle.h"
#include "sin_tabelle.h"

// *****
// * VARIABLE DECLARATION *
// *****

// Grundsicherungsregelung, RZM, ADC

volatile char flag = 1, ein = 0, t_ein = 1; // flag für Zeit-Interrupt 100 µs

int i; // Temp-Variable

//volatile int alpha, beta, betrag_alpha, betrag_beta, u; //RZM

volatile int U_ein = 1000, V_ein = 3000, W_ein = 5000; //Transistor Schaltzeiten
unsigned int Tr_schaltzeit_over = 0; // zaehlt die ueber-
schreitung der Transistorschaltzeiten
volatile short adc[8]; //Array zu Speicherung der Rohdaten des AD-Wandlers

//Strukturen für Strom und Spannung in uvw, ab, dq, KDS und U_zk, ü_stelltrafo
...
_stDatenInDreiAchsen stDreiPhasenNetzSpannung = {0, 0, 0};
_stDatenInDreiAchsen stDreiPhasenNetzStrom = {0, 0, 0};
_stDatenInFesteAchsen stAlphaBetaNetzSpannung = {0, 0, 0, 0};
_stDatenInFesteAchsen stAlphaBetaNetzStrom = {0, 0, 0, 0};
_stDatenInRotierendeAchsen stParkKomponenteStrom = {0, 0, 0};
_stSystemParameter stSystemParameter = {0, 0, 0, 0, 0, 0, 0, 0, 0};

```

```

// Strukturen für die Regelung der 50Hz Strom
_stPID_Regler      stPID_Regler_D = { 0,          /* Vorhergehende Regler-
abweichung */
                                0,          /* Proportionaler Anteil
des Reglers */
                                0,          /* Integraler Anteil des
Reglers */
                                0,          /* Differentialanteil */
                                REG_ABTASTZEIT, /* in welchen Zeitabsta-
enden der Regler kontinuierlich aufgerufen wird ACHTUNG! Angabe in Q38.24*/
                                STROM_KP,    /* Proportionalverstaer-
kung */
                                STROM_KI,    /* Integralverstaerkung
*/
                                0,          /* Differentialverstaee-
rung */
                                0,          /* Sollwert wird in ande-
rer Funktion hierher uebergeben*/
                                0,          /* Muss aus Struktur fuer
d und Q geholt werden.*/
                                0,          /* Ausgabewert des Reg-
lers im Eingabeformat*/
                                2};        /* 1=P, 2=PI, 3=PID */

_stPID_Regler      stPID_Regler_Q = { 0,          /* Vorhergehende Regler-
abweichung */
                                0,          /* Proportionaler Anteil
des Reglers */
                                0,          /* Integraler Anteil des
Reglers */
                                0,          /* Differentialanteil */
                                REG_ABTASTZEIT, /* in welchen Zeitabsta-
enden der Regler kontinuierlich aufgerufen wird ACHTUNG! Angabe in Q38.24*/
                                STROM_KP,    /* Proportionalverstaer-
kung */
                                STROM_KI,    /* Integralverstaerkung
*/
                                0,          /* Differentialverstaee-
rung */
                                0,          /* Sollwert wird in ande-
rer Funktion hierher uebergeben*/
                                0,          /* Muss aus Struktur fuer
d und Q geholt werden.*/
                                0,          /* Ausgabewert des Reg-
lers im Eingabeformat*/
                                2};        /* 1=P, 2=PI, 3=PID */

// Struktur fuer die Regelung der Zwischenkreisspannung
_stPID_Regler      stPID_Regler_U_zk = { 0,          /* Vorhergehende Regler-
abweichung */
                                0,          /* Proportionaler Anteil
des Reglers */
                                0,          /* Integraler Anteil des
Reglers */
                                0,          /* Differentialanteil */
                                REG_ABTASTZEIT, /* in welchen Zeitabsta-
enden der Regler kontinuierlich aufgerufen wird ACHTUNG! Angabe in Q38.24*/
                                16000,      /* Proportionalverstaer-
kung */
                                480,        /* Integralverstaerkung */
                                0,          /* Differentialverstaee-
rung */
                                15000,     /* Sollwert wird in ande-
rer Funktion hierher uebergeben*/
                                0,          /* Muss aus Struktur fuer
d und Q geholt werden.*/
                                0,          /* Ausgabewert des Reg-
lers im Eingabeformat*/
                                2};        /* 1=P, 2=PI, 3=PID */

```

```

struct
{
    short kanal1; /*Spannung U*/
    short kanal2; /*Spannung V*/
    short kanal3; /*Spannung W*/
    short kanal4; /*noch Frei*/
    short kanal5; /*Spannung ZK*/
    short kanal6; /*Strom W*/
    short kanal7; /*Strom V*/
    short kanal8; /*Strom U*/
    short update;      /*Daten werden Verarbeitet und in eigene Struktur geschrieben.
                                                                Wert wird auf 0 Gesetzt wenn dies
                                                                geschehen ist. Nach neuer
                                                                Wandlung wird wieder auf 1 gesetzt*/
} stADCDaten;

//*****
// VARIABLE DFT

//Einstellung der Oberschwingungskompensation

unsigned short ordnung = 7; //Ordnung der Oberschwingung
int OS_system = 1;          // Mitsystem = 1, Gegensystem = -1
                                //OS Mitsystem: 4. 7. 10. 13. 16. 19.
                                //OS Gegensystem:2. 5. 8. 11. 14. 17. 20.

// Messwert Speicherung
int Strom_U [16384], Strom_V [16384],Strom_W [16384];
int Spg_U [16384], Spg_V [16384], Spg_W [16384];
int Spg_U_ohne_50Hz[16384],Spg_V_ohne_50Hz[16384],Spg_W_ohne_50Hz[16384];
unsigned short ztime = 0;

//*****
//50Hz DFT
_stOS OS_Strom_U_50Hz;
unsigned short ztime50Hz = 0;
int cos_winkelOS1p, sin_winkelOS1p, cos_winkelOS1, sin_winkelOS1;
unsigned short winkelOS1, winkelOS1p;
int Regler_OS1_alpha, Regler_OS1_beta;
_stPID_Regler      stPID_Regler_OS1_D = { 0,          /* Vorhergehende Reglerabweichung */
0,          /* Proportionaler Anteil des Reglers */
0,          /* Integraler Anteil des Reglers */
0,          /* Differentialanteil */
REG_ABTASTZEIT, /* in welchen Zeitabstaenden der Regler kontinuierlich aufgerufen wird ACHTUNG! Angabe in Q38.24*/
4096,      /* Proportionalverstaerkung */
2400,      /* Integralverstaerkung */
0,          /* Differentialverstaerung */
0,          /* Sollwert wird in anderer Funktion hierher Uebergeben*/
0,          /* istwert Muss aus Struktur fuer d und Q geholt werden.*/
0,          /* Stellwert Ausgabewert des Reglers im Eingabeformat*/
2};        /* 1=P, 2=PI, 3=PID */

```

```

_stPID_Regler    stPID_Regler_OS1_Q = { 0,
                                         0,
                                         0,
                                         0,
                                         REG_ABTAZTZEIT,
                                         4096,
                                         2400,
                                         0,
                                         0,
                                         0,
                                         0,
                                         0,
                                         2};

//*****
// 5Hz DFT
_stOS OS_Spg_U_5Hz[2], OS_Spg_V_5Hz[2], OS_Spg_W_5Hz[2];
unsigned short ztime5Hz = 0;

//*****
// Phasenwinkel Korrektur
int Totzeiten = 1000;
unsigned short ztime_mittelw = 0, ztime_alt_mittelw = 0;
short Kreisfrequenz_U[4096], Kreisfrequenz_V[4096], Kreisfrequenz_W[4096];
int Kreisfrequenz_U_sum = 0, Kreisfrequenz_V_sum = 0, Kreisfrequenz_W_sum = 0;
unsigned short Phase_U_alt = 0, Phase_V_alt = 0, Phase_W_alt = 0;
unsigned int Phase_U_mittelw = 0, Phase_V_mittelw = 0, Phase_W_mittelw = 0;
short delta_Phase_U[4096], delta_Phase_V[4096], delta_Phase_W[4096];
int delta_Phase_U_sum = 0, delta_Phase_V_sum = 0, delta_Phase_W_sum = 0, Phasen-
winkel_totzeit;
unsigned short Phase_U_ok, Phase_V_ok, Phase_W_ok;

//*****
// Stromueberwachung
int max_strom_stop = 0, max_strom_reset = 0;

//*****
// Zeitablauf der Stromsollwerte
unsigned int I_soll_time = 0, I_soll_start = 0;

//*****
// *** Variable Messung und Auswertung ***
unsigned int messwertstart = 0;
int length=0, ablzeit[17], messung_ein = 0;

int Strom_L1[MESSDAUER], Strom_L2[MESSDAUER], Strom_L3[MESSDAUER];

int Spg_L1[MESSDAUER], Spg_L2[MESSDAUER], Spg_L3[MESSDAUER], Spg_U_zk[MESSDAUER];
/*
int I_soll_tim[MESSDAUER], I_soll_star[MESSDAUER];

// Regelung 50Hz Grundschiwingung
int Spg_alpha[MESSDAUER], Spg_beta[MESSDAUER], Wink_theta[MESSDAUER];

int strom_alpha[MESSDAUER], strom_beta[MESSDAUER];
*/
int strom_d_soll[MESSDAUER];

int strom_d_ist[MESSDAUER], strom_q_ist[MESSDAUER];
/*
int Regler_d[MESSDAUER], Regler_q[MESSDAUER];

// Regelung OS1
int strom_OS1_d_soll[MESSDAUER], ztime_alt_50[MESSDAUER];

int strom_OS1_d_ist[MESSDAUER], strom_OS1_q_ist[MESSDAUER];

```



```

int cos_OS1p[MESSDAUER], sin_OS1p[MESSDAUER], cos_OS1[MESSDAUER],
sin_OS1[MESSDAUER];
int winkelOS[MESSDAUER], winkelOSp[MESSDAUER];
*/
int Sollwert_OS1_d[MESSDAUER], Sollwert_OS1_q[MESSDAUER];
/*
int Regler_OS1_d[MESSDAUER], Regler_OS1_q[MESSDAUER];

int ztim[MESSDAUER], ztim_alt[MESSDAUER], StromU[MESSDAUER];

// Normierung und RZM
int Regler_alpha[MESSDAUER],Regler_beta[MESSDAUER];

int Regler_alpha_norm[MESSDAUER],Regler_beta_norm[MESSDAUER];

int TR_U_ein[MESSDAUER],TR_V_ein[MESSDAUER], TR_W_ein[MESSDAUER];

int TR_U_ei[MESSDAUER], TR_V_ei[MESSDAUER], TR_W_ei[MESSDAUER], TR_[MESSDAUER],
TR_ein[MESSDAUER];

int maintime [MESSDAUER];
*/
// OS Spannung Messung
int ampl_OS_Spg_U[MESSDAUER], ampl_OS_Spg_V[MESSDAUER], ampl_OS_Spg_W[MESSDAUER];
int phas_OS_Spg_U[MESSDAUER], phas_OS_Spg_V[MESSDAUER], phas_OS_Spg_W[MESSDAUER];
/*
int Re_OS_Spg_U[MESSDAUER], Im_OS_Spg_U[MESSDAUERr];

// Phasenwinkel korrektur
int phas_OS_Spg_U[MESSDAUER], phas_OS_Spg_U_ok[MESSDAUER], kreis-
freq_U[MESSDAUER];
int kreisfreq_U_sum[MESSDAUER], phas_U_mittelw[MESSDAUER], del-
ta_phas_U[MESSDAUER], delta_phas_U_sum[MESSDAUER];
int Phas_totzeit[MESSDAUER], delta_phas[MESSDAUER];
*/

//*****

// *****
// *          HAUPTPROGRAMM          *
// *****

void main(void) {

    int hilf;    //DAC

    init();                // Ports, Timer und Interrupts initialisieren

    *pPORTFIO_CLEAR = FREIGABE;

    arctan_tabelle_anlegen ();

    inbetriebnehmen();    // Uebersetzungsverhaeltnis vom Stelltransforma-
tor

    *pPORTFIO_SET = RELAI2; // Widerstände zuschalten

    *pPORTFIO_SET = FREIGABE; // Freigabe der Steuerung der IGBTs

    while(1) {

        // while(1): Warten, while(0): Weiter
        while (!(flag && t_ein));                //auf Interrupt warten

```

```

// *** Rechteckige Zeitablauf des Stroms Sollwertes ***

    if (I_soll_start >= 100000)
    {
    if(I_soll_time >= 200000){I_soll_time = 0;} //Rechteck Periode
    else {I_soll_time ++;}

        if (I_soll_time <= 100000)
        {
            if(OS_system >= 0){
                stPID_Regler_OS1_D.sollwert = -
(cos_liste[Phase_U_ok] >> 3);
                stPID_Regler_OS1_Q.sollwert = -
(sin_liste[Phase_U_ok] >> 3);}
            else{
                stPID_Regler_OS1_D.sollwert = -
(cos_liste[Phase_U_ok] >> 3);
                stPID_Regler_OS1_Q.sollwert = (sin_liste[Phase_U_ok]
>> 3);}

                stPID_Regler_D.sollwert = 0;
                stPID_Regler_Q.sollwert = -4000;
            }
        else
        {
            stPID_Regler_OS1_D.sollwert = 0;
            stPID_Regler_OS1_Q.sollwert = 0;

            stPID_Regler_D.sollwert = 0;
            stPID_Regler_Q.sollwert = 0;
        }
    }

    else
    {
        I_soll_start++;
        if (I_soll_start >= 1000) *pPORTFIO_SET = RELAI1;
//Wechselrichter direkt am Netz
    }

    ADC_read();

    messwertverteilung(&stDreiPhasenNetzSpannung, &stDreiPhasenNetzStrom,
&stSystemParameter);

    clarktransformation_unsymmetrisch(&stDreiPhasenNetzSpannung,
&stAlphaBetaNetzSpannung);

    clarktransformation_unsymmetrisch_ohne_theta(&stDreiPhasenNetzStrom,
&stAlphaBetaNetzStrom);

    park_unsymmetrie(&stAlphaBetaNetzStrom, &stParkKomponenteStrom,
&stAlphaBetaNetzSpannung);

    DFT(); // Messung und Regelung der Oberschwingungen

    if (I_soll_start >= 5000) //Reglung der Grundschiwingung nach 500ms
Starten

```

```

{
    stPID_Regler_U_zk.istwert = stADCDaten.kanal5;
    PID_regler(&stPID_Regler_U_zk);    //Reglung von U_zk
    stPID_Regler_D.sollwert = -stPID_Regler_U_zk.stellwert;

    stPID_Regler_D.istwert = stParkKomponenteStrom.d_part;
    stPID_Regler_Q.istwert = stParkKomponenteStrom.q_part;

    PID_regler(&stPID_Regler_D);
    PID_regler(&stPID_Regler_Q);

    invers_park_unsymmetrie(stPID_Regler_D.stellwert,
stPID_Regler_Q.stellwert, &stSystemParameter, &stAlphaBetaNetzSpannung);
}

    stSystemParameter.regler_alpha += Regler_OS1_alpha; //Summe 50Hz + OS-
Reglung
    stSystemParameter.regler_beta += Regler_OS1_beta;

    normierung(&stAlphaBetaNetzSpannung, &stSystemParameter);

    RZM(stSystemParameter.rzm_alpha, stSystemParameter.rzm_beta);

    /*
    // Ausgabe an DAC zur Visualisierung am Oszi
    hilf = U_ein << 4;
    if (hilf > 32767) hilf = 32767;
    DAC_A(hilf);    // Daten senden (DAC A)
    hilf = V_ein << 4;
    if (hilf > 32767) hilf = 32767;
    DAC_B(hilf);    // Daten senden (DAC B)
    */

// *** Messwert speicherung    für Auswertung ***

    if ((messwertstart >= 100000) && (length < MESSDAUER) && (mes-
sung_ein >= 6)) //&& (messung_ein == 1)
    {
        messung_ein = 0;
        /*
        Spg_L1[length] = stADCDaten.kanal1;
        Spg_L2[length] = stADCDaten.kanal2;
        Spg_L3[length] = stADCDaten.kanal3;
        */
        Spg_U_zk[length] = stADCDaten.kanal5;
        /*
        I_soll_tim[length] = I_soll_time;
        I_soll_star[length] = I_soll_start;

        Spg_alpha[length] = stAlphaBetaNetzSpannung.alpha;
        Spg_beta[length] = stAlphaBetaNetzSpannung.beta;
        Wink_theta[length] = stAlphaBetaNetzSpannung.theta;
        */
        Strom_L1[length] = stADCDaten.kanal8;
        Strom_L2[length] = stADCDaten.kanal7;
        Strom_L3[length] = stADCDaten.kanal6;
        /*
        strom_alpha[length] = stAlphaBetaNetzStrom.alpha;
        strom_beta[length] = stAlphaBetaNetzStrom.beta;
        */
        strom_d_ist[length] = stParkKomponenteStrom.d_part;

```

```

strom_q_ist[length] = stParkKomponenteStrom.q_part;

strom_d_soll[length] = stPID_Regler_D.sollwert;
/*
Regler_d[length] = stPID_Regler_D.stellwert;
Regler_q[length] = stPID_Regler_Q.stellwert;

Regler_d_ek[length] = stPID_Regler_D.e_k_minus_1;
Regler_q_ek[length] = stPID_Regler_Q.e_k_minus_1;

strom_OS1_d_ist[length] = stPID_Regler_OS1_D.istwert;
strom_OS1_q_ist[length] = stPID_Regler_OS1_Q.istwert;

cos_OS1p[length] = cos_winkelOS1p;
sin_OS1p[length] = sin_winkelOS1p;
cos_OS1[length] = cos_winkelOS1;
sin_OS1[length] = sin_winkelOS1;
winkelOS[length] = winkelOS1;
winkelOSp[length] = winkelOS1p;

Regler_OS1_d[length] = stPID_Regler_OS1_D.stellwert;
Regler_OS1_q[length] = stPID_Regler_OS1_Q.stellwert;

Regler_alpha[length] = Regler_OS1_alpha;
Regler_beta[length] = Regler_OS1_beta;

ztim[length] = ztime;
ztim_alt[length] = ztime_alt_50Hz;
StromU[length] = Strom_U[ztime];
delta_strom[length] = delta_strom_U_50Hz;

ztime_alt_50[length] = ztime_alt_50Hz;

Regler_alpha_norm[length] = stSystemParameter.rzm_alpha;
Regler_beta_norm[length] = stSystemParameter.rzm_beta;

TR_U_ein[length] = U_ein;
TR_V_ein[length] = V_ein;
TR_W_ein[length] = W_ein;

Re_OS_Spg_U[length] = OS_Spg_U_5Hz[1].Re;
Im_OS_Spg_U[length] = OS_Spg_U_5Hz[1].Im;
*/
Sollwert_OS1_d[length] = stPID_Regler_OS1_D.sollwert;
Sollwert_OS1_q[length] = stPID_Regler_OS1_Q.sollwert;

ampl_OS_Spg_U[length] = OS_Spg_U_5Hz[1].ampl;
ampl_OS_Spg_V[length] = OS_Spg_V_5Hz[1].ampl;
ampl_OS_Spg_W[length] = OS_Spg_W_5Hz[1].ampl;
phas_OS_Spg_U[length] = OS_Spg_U_5Hz[1].phas;
phas_OS_Spg_V[length] = OS_Spg_V_5Hz[1].phas;
phas_OS_Spg_W[length] = OS_Spg_W_5Hz[1].phas;
/*
Phas_totzeit[length] = Phasenwinkel_totzeit;
delta_phas[length] = (short) (OS_Spg_U_5Hz[1].phas - (Phase_U_mittelw >>
16));

phas_OS_Spg_U_ok[length] = Phase_U_ok;
kreisfreq_U[length] = Kreisfrequenz_U[ztime_mittelw];
kreisfreq_U_sum[length] = (Kreisfrequenz_U_sum );
phas_U_mittelw[length] = (Phase_U_mittelw >> 16);
delta_phas_U[length] = delta_Phase_U[ztime_mittelw];
delta_phas_U_sum[length] = (delta_Phase_U_sum );
*/

//maintime [length] = *pTIMER0_COUNTER;

length++;

```

```

    }
    messung_ein++;

    messwertstart++;

    // *** Stromueberwachung ***
    if ((stADCDaten.kanal8 > MAX_STROM) || (stADCDaten.kanal7 >
MAX_STROM) || (stADCDaten.kanal6 > MAX_STROM))
    {
        if( (stADCDaten.kanal8 > MAX_STROM2) || (stADCDaten.kanal7 >
MAX_STROM2) || (stADCDaten.kanal6 > MAX_STROM2))
        {
            *pPORTFIO_CLEAR = FREIGABE;
            *pPORTFIO_CLEAR = RELAI2;
            *pPORTFIO_CLEAR = RELAI1;
            while(1);
        }
        max_strom_stop++;
        max_strom_reset = 0;
        if( max_strom_stop > 200)
        {
            *pPORTFIO_CLEAR = FREIGABE;
            *pPORTFIO_CLEAR = RELAI2;
            *pPORTFIO_CLEAR = RELAI1;
            while(1);
        }
    }
    else
    {
        if (max_strom_reset < 20) max_strom_reset++;
        if (max_strom_reset > 15) max_strom_stop = 0;
    }

    ablzeit[15]=*pTIMER0_COUNTER;

    flag = 0;
}

// ***** ISR *****
// *****

// ***** ISR für Timer0 *****

EX_INTERRUPT_HANDLER(Timer0_ISR) {
    // confirm interrupt handling
    *pTIMER_STATUS = 0x0001;

    delay_85ns(); // Warten dass die Timer eine neu Periode gestartet
    haben // vor die neu Schaltzeiten Laden sonst mischmach

    if(t_ein >= 1)
    {
        *pTIMER1_WIDTH = U_ein;
        *pTIMER2_WIDTH = V_ein;
        *pTIMER3_WIDTH = W_ein;
        t_ein = 0;
    }
    else
    {
        *pTIMER1_WIDTH = ZWEI_T - U_ein;
        *pTIMER2_WIDTH = ZWEI_T - V_ein;
        *pTIMER3_WIDTH = ZWEI_T - W_ein;
        t_ein = 1;
    }
}

```

```

    flag = 1;
}

// ***** INITIALISIERUNGS-ROUTINEN *****
// *****

void init(void) {
    // ***** Initialisieren der Ports (PPI-Anschluß) *****

    // 16-bit-Parallelport für Dateneingang
    *pPORTG_FER = 0x0000;           // GPIO-Modus
    *pPORTGIO_DIR = 0x0000;        // Port als Eingang
    *pPORTGIO_INEN = 0xFFFF;      // Eingänge aktiviert

    // Steuerleitungen
    *pPORTF_FER = LED3 | LED2 | LED1 | CONVSTA_B; // TMRx-Ausgänge zum
    Umrichter - Function Enabled Register kommt von Timer
    *pPORTFIO_DIR = RELAI2 | RELAI1 | RELAI3 | FREIGABE | LED1 | LED2 | LED3 |
    LED6 // Ausgänge werden vom Benutzer gesetzt ganz normal im Programmablauf
    | RESET | CS_RD | CONVSTA_B | RANGE | STBY;

    *pPORTFIO_INEN = BUSY;         // Eingänge
    // *pPORTFIO_EDGE = TASTER;     // Interrupt auf steigende
    Flanke
    // *pPORTFIO_MASKA = TASTER;

    // ***** ADC INIT *****

    *pPORTFIO_SET = CS_RD | STBY | CONVSTA_B; // einschalten
    delay_us(100000);                    // 100 ms warten, bis
    initialisiert
    *pPORTFIO_SET = RESET;                // kurzen Reset-Impuls
    erzeugen
    delay_85ns();
    *pPORTFIO_CLEAR = RESET;
    delay_85ns();
    *pPORTFIO_SET = RANGE;                // Eingangsspannungsbe-
    reich ±10 V

    // ***** SPORT0 INIT *****
    *pSPORT0_TCR1 = 0;
    *pSPORT0_TCR1 = ITCLK | ITFS | TFSR;
    *pSPORT0_TCR2 = Bitlaenge;
    *pSPORT0_TCLKDIV = Teiler;
    *pSPORT0_TFSDIV = Bitlaenge + 2;     // Pausenzeit vor nächstem Byte
    asm("ssync;");
    *pSPORT0_TCR1 |= TSPEN;                // Sender einschalten
    asm("ssync;");

    // ***** DAC INIT *****
    while(*pSPORT0_STAT & TXF);           // warten, bis TX-Puffer leer
    *pSPORT0_TX = 0x280001;               // Software-Reset ("Power-on Re-
    set")

    // ***** TIMER INIT *****
    // Timer-Taktrate ist 1/5 Core-Clock, d.h. 120 MHz

    *pTIMER0_CONFIG = PWM_OUT | PERIOD_CNT | IRQ_ENA;
    *pTIMER0_PERIOD = 6000;                // Timer für 50 µs
    Zykluszeit ergibt 20 kHz Interruptrate
    *pTIMER0_WIDTH = 0x0000000A;

    // PWM für LED3 = U
    *pTIMER1_CONFIG = PWM_OUT | PERIOD_CNT | TOGGLE_HI | PULSE_HI;
    *pTIMER1_PERIOD = 6000;

```

```

*pTIMER1_WIDTH = 100;

// PWM für LED2 = V
*pTIMER2_CONFIG = PWM_OUT | PERIOD_CNT | TOGGLE_HI | PULSE_HI;
*pTIMER2_PERIOD = 6000;
*pTIMER2_WIDTH = 100;

// PWM für LED1 = W
*pTIMER3_CONFIG = PWM_OUT | PERIOD_CNT | TOGGLE_HI | PULSE_HI;
*pTIMER3_PERIOD = 6000;
*pTIMER3_WIDTH = 100;

// Timer für Blink-LED (1 Hz)
*pTIMER4_CONFIG = PWM_OUT | PERIOD_CNT | OUT_DIS;
*pTIMER4_PERIOD = 0x07270E00;
*pTIMER4_WIDTH = 0x03938700;

// ***** INTERRUPT INIT *****

// assign core IDs to interrupts
*pSIC_IAR0 = 0xffffffff;
*pSIC_IAR1 = 0xffffffff;
*pSIC_IAR2 = 0xffff4fff; // Timer0 -> ID4; Timer2, Timer 3
*pSIC_IAR3 = 0xffffffff; // PORTF IVG12

// assign ISRs to interrupt vectors
register_handler(ik_ivg11, Timer0_ISR); // Timer0 ISR -> IVG 11
//register_handler(ik_ivg12,PORTF_IntA_ISR);

// Interrupt Timer0 einschalten
*pSIC_IMASK = 0x08080000;
// Timer einschalten
*pTIMER_ENABLE = 0x001F;
}

// ***** ADC auslesen *****
// *****

void ADC_read(void) {
    int i;

    while(*pPORTFIO & BUSY); // warten, bis fertig gewandelt

    for(i=0; i<8;i++) {
        *pPORTFIO_CLEAR = CS_RD;
        delay_85ns();
        adc[i] = *pPORTGIO; // alle 8 Kanäle nach adc[] auslesen
        *pPORTFIO_SET = CS_RD; // dauert 1360 ns
        delay_85ns();
    }

    stADCDaten.kanal1 = adc[0];
    stADCDaten.kanal2 = adc[1];
    stADCDaten.kanal3 = adc[2];
    stADCDaten.kanal4 = adc[3];
    stADCDaten.kanal5 = adc[4];
    stADCDaten.kanal6 = adc[5];
    stADCDaten.kanal7 = adc[6];
    stADCDaten.kanal8 = adc[7];
    stADCDaten.update = 1;
}

```

```

// ***** VERZÖGERUNGS-ROUTINEN *****
// *****

// Übergabewert: Verzögerungszeit in µs
void delay_us(unsigned long int zeit) {
    int countup;
    for(countup=0; countup<DELAY_MUL*zeit; countup++)
        asm("nop;");
}

// feste Verzögerungszeit 85 ns
void delay_85ns(void) {
    for(i=0; i<4; i++)
        asm("nop;");
}

// ***** Daten an DAC senden *****
// Übergabewert: 16-bit (unsigned) int
// *****

// an DAC Kanal A senden
void DAC_A(int daten) {
    while(*pSPORT0_STAT & TXF); // warten, bis TX-Puffer leer
    *pSPORT0_TX = ((daten | KANAL_A) /*& 0x00FFFFFF*/); // Daten senden (DAC
A)
}

// an DAC Kanal B senden
void DAC_B(int daten) {
    while(*pSPORT0_STAT & TXF); // warten, bis TX-Puffer leer
    *pSPORT0_TX = ((daten | KANAL_B) /*& 0x00FFFFFF*/); // Daten senden (DAC
B)
}

```

normierung.h

```

#ifndef _NORMIERUNG_H_
#define _NORMIERUNG_H_

void messwertverteilung(_stDatenInDreiAchsen *, _stDatenInDreiAchsen *,
_stSystemParameter *);

void normierung(_stDatenInFesteAchsen *, _stSystemParameter *);

void strommittel(_stDatenInDreiAchsen *);

void gleichrichtwert(_stDatenInDreiAchsen *);

void gleichrichtwert_DQ(_stDatenInRotierendeAchsen *);

#endif

/* EOF */

```

normierung.c

```

/*****
Normierung WR Betrieb
*****/

#include "main.h"
#include "div_tabelle.h"

extern      struct
{
    short kanall; /*Rohdaten Spannung U*/

```



```

    short kanal2; /*Rohdaten Spannung V*/
    short kanal3; /*Rohdaten Spannung W*/
    short kanal4; /*noch Frei*/
    short kanal5; /*Rohdaten Spannung UZK*/
    short kanal6; /*Rohdaten Strom W*/
    short kanal7; /*Rohdaten Strom V*/
    short kanal8; /*Rohdaten Strom U*/
    short update; /*Daten werden Verarbeitet und in eigene Struktur
geschrieben.
geschehen ist. Nach neuer Wert wird auf 0 Gesetzt wenn dies
setzt*/ Wandlung wird wieder auf 1 ge-
} stADCCDaten;

/*extern _stDatenInDreiAchsen      stDreiPhasenNetzSpannung;
extern _stDatenInDreiAchsen      stDreiPhasenNetzStrom;
extern _stSystemParameter        stSystemParameter;
extern _stDatenInFesteAchsen     stAlphaBetaNetzSpannung;*/

void messwertverteilung(_stDatenInDreiAchsen *spannungen, _stDatenInDreiAchsen
*stroeme, _stSystemParameter *zwischenkreis)
{
    spannungen->u_phas = ((stADCCDaten.kanal1 * SPANNUNG_UVW) >> 15); //Q10.6: -500V
bis 500V <=> -32000 bis 32000
    spannungen->v_phas = ((stADCCDaten.kanal2 * SPANNUNG_UVW) >> 15);
    spannungen->w_phas = ((stADCCDaten.kanal3 * SPANNUNG_UVW) >> 15);

    zwischenkreis->zwischenkreisspannung = ((stADCCDaten.kanal5 * SPANNUNG_UZK) >>
15); //Q12.4: -1500V bis 1500V <=> -24000 bis 24000

    stroeme->u_phas = ((stADCCDaten.kanal8 * STROM_UVW) >> 11); //Q5.11: -20A bis
20A <=> -40960 bis 40960
    stroeme->v_phas = ((stADCCDaten.kanal7 * STROM_UVW) >> 11);
    stroeme->w_phas = ((stADCCDaten.kanal6 * STROM_UVW) >> 11);

    //Begrenzung der Strommesswert gegen überlaufgefahr
    if (stroeme->u_phas > 32700) stroeme->u_phas = 32700;
    if (stroeme->v_phas > 32700) stroeme->v_phas = 32700;
    if (stroeme->w_phas > 32700) stroeme->w_phas = 32700;
    if (stroeme->u_phas < -32700) stroeme->u_phas = -32700;
    if (stroeme->v_phas < -32700) stroeme->v_phas = -32700;
    if (stroeme->w_phas < -32700) stroeme->w_phas = -32700;
}

/*****
Funktion benÄtigt die Alpha- und Beta-Werte vom Netz und vom Reglerausgang
die zurÄcktransformierten Alpha- Beta-Werte
FÄr die Funktion RZM werden die Alpha- und Beta-Werte auf das Format Q1.15
normiert
*****/

void normierung(_stDatenInFesteAchsen *netz_ct, _stSystemParameter
*rzm_parameter)
{
    int hilf = 0;
    int ergeb_div = 0;
    int hilf_alpha_n = 0;
    int hilf_beta_n = 0;
    int U_zk_wurzel_3 = 0;

    // Normierung vom Netz für alpha auf Q1.15 <=>
    // rzm_alpha = (U_alpha * SQRT3HALBE * trafo_u) / (U_zk * EINS DURCHSQRT3)
    hilf = ((netz_ct->alpha * SQRT3HALBE) >> 14);

```

```

// Q10.6 = (Q10.6 * Qn.14) >> 14
// hilf_MAX = 28377 = (32767 * 14189) >> 14

//trafo_uebersetzungsverhaeltniss Idealwert 22937 = 0,7 Q1.15 sollte 22446> u
>23429
hilf_alpha_n = ((hilf * rzm_parameter->uebersetzungsverhaeltniss) >> 15);
// Q10.6 = (Q10.6 * Q1.15) >> 15
// hilf_alpha_n_MAX = 20290 = (28377 * 23429) >> 15

U_zk_wurzel_3 = ((rzm_parameter->zwischenkreisspannung * EINS DURCHSQRT3) >>
15);
// Q12.4 = (Q12.4 * Qn.15) >> 15
// U_zk_wurzel_3_MAX = 13856 = 24000 * EINS DURCHSQRT3
// AUFLÖSUNG VERLUST U_zk = 600V <=> 9600 --> U_zk_wurzel_3 = 5543

//Division anhand Tabelle: adress = nenner, data = | format | 1/nenner |
ergeb_div = div_liste[U_zk_wurzel_3]; // | format: bit 31 ... 16 |
1/U_zk_wurzel_3: bit 15 ... 0 |

//temp_beta[length] = U_zk_wurzel_3;
//temp_alpha[length] = ergeb_div >> 16;

rzm_parameter->rzm_alpha = (hilf_alpha_n * (signed short)ergeb_div) >>
((ergeb_div >> 16) - 13);
// Q1.15 = (Q10.6 << 13) / Q12.4

// Normierung für Alpha von der Regelung auf das Format Q1.15 und Addition zu
den Wert vom Netz
rzm_parameter->rzm_alpha += ((rzm_parameter->regler_alpha * (signed
short)ergeb_div) >> ((ergeb_div >> 16) - 13));
// Q1.15 = Q1.15 + (Q10.6 << 13) / Q12.4
// regler_alpha MAX = 46340
// 46340 * 32768 = 1518469120 < 2^31 multiplikation ok!

if (rzm_parameter->rzm_alpha > 32767) rzm_parameter->rzm_alpha = 32767;
if (rzm_parameter->rzm_alpha < -32767) rzm_parameter->rzm_alpha = -32767;

// Normierung vom Netz für beta auf Q1.15 (gleich wie alpha)
hilf = ((netz_ct->beta * SQRT3HALBE) >> 14);

hilf_beta_n = ((hilf * rzm_parameter->uebersetzungsverhaeltniss) >> 15);

rzm_parameter->rzm_beta = (hilf_beta_n * (signed short)ergeb_div) >>
((ergeb_div >> 16) - 13);

rzm_parameter->rzm_beta += ((rzm_parameter->regler_beta * (signed
short)ergeb_div) >> ((ergeb_div >> 16) - 13));

if (rzm_parameter->rzm_beta > 32767) rzm_parameter->rzm_beta = 32767;
if (rzm_parameter->rzm_beta < -32767) rzm_parameter->rzm_beta = -32767;
}

```

parktransformation.h

```

#ifndef _PARKTRANSFORMATION_H_
#define _PARKTRANSFORMATION_H_

```

```

void park_unsymmetrie(_stDatenInFesteAchsen *, _stDatenInRotierendeAchsen *,
_stDatenInFesteAchsen *);
void invers_park_unsymmetrie(int , int, _stSystemParameter *,
_stDatenInFesteAchsen *);

```

```

#endif

```

parktransformation.c

```

#include "main.h"
#include "cos_tabelle.h"

```

```

#include "sin_tabelle.h"

/*****
----- Funktion für Wechselrichterbetrieb -----
Funktionsaufruf durch
park_unsymmetrie(&stParkKomponenteStrom, &stAlphaBetaNetzStrom,
&stAlphaBetaNetzSpannung);
invers_park_unsymmetrie(&stAlphaBetaNetzStrom, &stParkKomponenteStrom,
&stAlphaBetaNetzSpannung);
Funktion berechnet mit Drehwinkel der Spannung die Parkkomponenten des Stromes
Äbergeben werden Zeiger auf Strukturen
*****/

void park_unsymmetrie( _stDatenInFesteAchsen *eingabewerte,
                     _stDatenInRotierendeAchsen *ausgabewerte,
                     _stDatenInFesteAchsen *winkel)
{
    ausgabewerte->d_part = ((cos_liste[winkel->theta] * eingabewerte->alpha) >>
15)
                          + ((sin_liste[winkel->theta] * eingabewerte->beta) >>
15);

    ausgabewerte->q_part = ((cos_liste[winkel->theta] * eingabewerte->beta) >>
15)
                          - ((sin_liste[winkel->theta] * eingabewerte->alpha)
>> 15);
}

void invers_park_unsymmetrie(int d_part, int q_part, _stSystemParameter
*sysparameter, _stDatenInFesteAchsen *winkel)
{
    sysparameter->regler_alpha = ((cos_liste[winkel->theta] * d_part) >> 15)
                                - ((sin_liste[winkel->theta] * q_part) >> 15);
    //Q10.6 = ((Q10.6 * Q1.15) >> 15) - ((Q10.6 * Q1.15) >> 15)

    sysparameter->regler_beta = ((cos_liste[winkel->theta] * q_part) >> 15)
                                + ((sin_liste[winkel->theta] * d_part) >> 15);
}

```

PID_Regler.h

```

#ifndef _PID_REGLER_H_
#define _PID_REGLER_H_

void PID_regler(_stPID_Regler *);
#endif

/** EOF */

```

PID_Regler.c

```

#include "main.h"
#include "PID_Regler.h"
#include "DFT.h"

void PID_regler(_stPID_Regler *reglerwerte)
{
    int e_k;
    int ki_mal_Ttakt, I_output;

    e_k = reglerwerte->sollwert - (reglerwerte->istwert);

    /*Begrenzung der Regler Abweichung */
    if (e_k < -(P_REG_Y_MAX)){e_k = -(P_REG_Y_MAX);}
}

```

```

    else {if (e_k > P_REG_Y_MAX) e_k = P_REG_Y_MAX;}

reglerwerte->p_anteil = ((reglerwerte->kp * e_k) >> 12);
// Q5.11 = (Qn.12 * Q5.11) >> 12

/*Begrenzung des P-Anteils */
    if (reglerwerte->p_anteil < -(PA_MAX)) {reglerwerte->p_anteil = -(
PA_MAX);}
    if (reglerwerte->p_anteil > PA_MAX) {reglerwerte->p_anteil = PA_MAX;}
    //reglerwerte->p_anteil_MAX = 196608

reglerwerte->i_anteil = reglerwerte->i_anteil + e_k; //Q5.11 = Q5.11 + Q5.11

//Begrenzung des I-Anteils IA_MAX = 2^23 = 8388608
    if (reglerwerte->i_anteil < -(IA_MAX)) {reglerwerte->i_anteil = -(IA_MAX);}
    if (reglerwerte->i_anteil > IA_MAX) {reglerwerte->i_anteil = IA_MAX;}

ki_mal_Ttakt = ((reglerwerte->ki * reglerwerte->abtastzeit) >> 12);
// Qn.16 = (Qn.4 * Qn.24) >> 12
//ki_mal_Ttakt_MAX = 2582 = (6303 * 1678) >> 12

I_output = ((reglerwerte->i_anteil >> 4) * ki_mal_Ttakt) >> 12;
// Q5.11 = (Qn.16 * (Q5.11 >> 4)) >> 112 = (Qn.16 * Q9.7) >> 12
//I_output_MAX = 7212 = ((16777216 >> 4) * 983) >> 12 = (1048576 * 983) >> 12
= 1030750208 >> 12 = 251648

reglerwerte->stellwert = (reglerwerte->p_anteil + I_output) >> 5;
//Q10.6 = (Q5.11 + Q5.11) >> 5
//reglerwerte->stellwert_MAX

/*Begrenzung der Stellgröße */
    if (reglerwerte->stellwert < -32767) {reglerwerte->stellwert = -32767;}
    if (reglerwerte->stellwert > 32768) {reglerwerte->stellwert = 32768;}

reglerwerte->e_k_minus_1 = e_k;
}

```

RZM.c

```

#include "main.h"
#include <math.h>

extern unsigned int Tr_schaltzeit_over;

void RZM(int usalpha, int usbeta)
{
    int betrag_alpha = abs(usalpha);
    int betrag_beta = abs(usbeta);
    int u = betrag_alpha - ((betrag_beta * WURZEL3_REZ) >> 16);

    if(usbeta < 0) {
        // Q3 oder Q4
        if(usalpha < 0) {
            // Q3
            if(u < 0) {
                // S5/Q3
                U_ein = (T_HALBE * (32768 + (betrag_alpha << 1))) >>
15;
                // 32768 <=> 1 in Qn.15
                V_ein = (T_HALBE * (32768 + ((betrag_beta * WUR-
ZEL3_REZ) >> 15))) >> 15;
                W_ein = (T_HALBE * (32768 - ((betrag_beta * WUR-
ZEL3_REZ) >> 15))) >> 15;
            }
            else {
                // S4
                U_ein = (T_HALBE * (32768 + betrag_alpha + ((be-
trag_beta * WURZEL3_REZ) >> 16))) >> 15;
                V_ein = (T_HALBE * (32768 - betrag_alpha + ((WURZEL3 *
betrag_beta) >> 15))) >> 15;
                W_ein = (T_HALBE * (32768 - betrag_alpha - ((be-
trag_beta * WURZEL3_REZ) >> 16))) >> 15;
            }
        }
    }
}

```

```

    }
    else { // Q4
        if(u < 0) { // S5/Q4
            U_ein = (T_HALBE * (32768 - (betrag_alpha << 1))) >>
15;
            V_ein = (T_HALBE * (32768 + ((betrag_beta * WUR-
ZEL3_REZ) >> 15))) >> 15;
            W_ein = (T_HALBE * (32768 - ((betrag_beta * WUR-
ZEL3_REZ) >> 15))) >> 15;
        }
        else { // S6
            U_ein = (T_HALBE * (32768 - betrag_alpha - ((be-
trag_beta * WURZEL3_REZ) >> 16))) >> 15;
            V_ein = (T_HALBE * (32768 + betrag_alpha + ((be-
trag_beta * WURZEL3_REZ) >> 16))) >> 15;
            W_ein = (T_HALBE * (32768 + betrag_alpha - ((WURZEL3 *
betrag_beta) >> 15))) >> 15;
        }
    }
}
else { // Q1 oder Q2
    if(usalpha < 0) { // Q2
        if(u < 0) { // S2/Q2
            U_ein = (T_HALBE * (32768+ (betrag_alpha << 1))) >> 15;
            V_ein = (T_HALBE * (32768 - ((betrag_beta * WUR-
ZEL3_REZ) >> 15))) >> 15;
            W_ein = (T_HALBE * (32768 + ((betrag_beta * WUR-
ZEL3_REZ) >> 15))) >> 15;
        }
        else { // S3
            U_ein = (T_HALBE * (32768 + betrag_alpha + ((be-
trag_beta * WURZEL3_REZ) >> 16))) >> 15;
            V_ein = (T_HALBE * (32768 - betrag_alpha - ((be-
trag_beta * WURZEL3_REZ) >> 16))) >> 15;
            W_ein = (T_HALBE * (32768 - betrag_alpha + ((WURZEL3 *
betrag_beta) >> 15))) >> 15;
        }
    }
    else { // Q1
        if(u < 0) { // S2/Q1
            U_ein = (T_HALBE * (32768 - (betrag_alpha << 1))) >>
15;
            V_ein = (T_HALBE * (32768 - ((betrag_beta * WUR-
ZEL3_REZ) >> 15))) >> 15;
            W_ein = (T_HALBE * (32768 + ((betrag_beta * WUR-
ZEL3_REZ) >> 15))) >> 15;
        }
        else { // S1
            U_ein = (T_HALBE * (32768 - betrag_alpha - ((be-
trag_beta * WURZEL3_REZ) >> 16))) >> 15;
            V_ein = (T_HALBE * (32768 + betrag_alpha - ((WURZEL3 *
betrag_beta) >> 15))) >> 15;
            W_ein = (T_HALBE * (32768 + betrag_alpha + ((be-
trag_beta * WURZEL3_REZ) >> 16))) >> 15;
        }
    }
}

if(U_ein < 60) {U_ein = 60; Tr_schaltzeit_over++;}
if(V_ein < 60) {V_ein = 60; Tr_schaltzeit_over++;}
if(W_ein < 60) {W_ein = 60; Tr_schaltzeit_over++;}
if(U_ein > 5940) U_ein = 5940;
if(V_ein > 5940) V_ein = 5940;
if(W_ein > 5940) W_ein = 5940;
}

```


Literatur

- [KLEI10] Kleinert, Siegfried, Energieübertragung und –verteilung, Vorlesungsskript Hochschule Mittweida, 2010
- [LAMP12] Lampe, Alexander, Signal- und Systemtheorie, Vorlesungsskript Hochschule Mittweida, 2012
- [NPE12] Gemeinsame Geschäftsstelle Elektromobilität der Bundesregierung (GGEMO): Zweiter Bericht der Nationalen Plattform Elektromobilität, URL: <http://www.bmu.de/files/pdfs/allgemein/application/pdf/bericht_e_mob_2.pdf>, verfügbar am 05.02.2013
- [QUAN99] Quang, Nguyen Phung, Praxis der feldorientierten Drehstromantriebsregelungen, expert Verlag, 1999
- [SCMO08] Schlabbach, Mombauer, Power Quality, VDE Verlag, 2008
- [SCHU04] Schulz, Detlef, Netzzrückwirkungen, VDE Verlag, 2004
- [SCHU13] Schulz, Detlef, Poster Impedanzmessung, URL: <http://www.hsu-hh.de/download-1.4.1.php?brick_id=a5FC66OxRwmAgqEz>, verfügbar am 03.04.2013
- [TÜVS12] URL: <http://www.netinform.net/GA/Wegweiser/Guide2.aspx?Ebene1_ID=23&Ebene2_ID=339>, verfügbar am 22.01.2012

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, den 17.07.2013

Jérémie Foulquier