



TECHNISCHE
UNIVERSITÄT
WIEN



Institut für Energiesysteme
und Elektrische Antriebe

Entwurf und Realisierung eines überstromfesten Sinuswechselrichters mit parallelen Brückenzweigen und phasenversetzter Taktung

DIPLOMARBEIT

Ausgeführt zum Zwecke der Erlangung des akademischen Grades eines
Diplom-Ingenieurs (Dipl.-Ing.)

unter der Leitung von

Ao. Univ. Prof. Dipl.-Ing. Dr. techn. Johann Ertl

eingereicht an der

Technischen Universität Wien

Fakultät für Elektrotechnik und Informationstechnik
Institut für Energiesysteme und Elektrische Antriebe

von

Christoph Friedrich
Matrikelnummer 1126125
Gusenleithnergasse 5/1/17
1140
Wien

Wien, im Dezember 2019

Arbeitsgebiet Elektrische Antriebe und Maschinen

A-1040 Wien, Gußhausstr. 25–29/370-2, Internet: <https://www.ieam.tuwien.ac.at>

Vorwort

Die vorliegende Diplomarbeit entstand im Rahmen meines Studiums der Energie- und Automatisierungstechnik am Institut für Energiesysteme und elektrische Antriebe der Technischen Universität Wien.

Meinem Betreuer Herrn Ao. Univ. Prof. Dipl.-Ing. Dr. techn. Johann Ertl möchte ich hier für die Möglichkeit der Durchführung und die großartige Unterstützung bei der Erstellung der Arbeit danken.

Weiters bedanke ich mich bei Dipl.-Ing. Thomas Fuchslueger für die lehrreichen Gespräche und die Hilfe bei den Arbeiten im Labor.

Besonderer Dank gilt auch meinen Eltern Susanne und Andreas für die Unterstützung während des gesamten Studiums.

Wien, im Dezember 2019

Abstract

This diploma thesis deals with the design and implementation of a sine-wave inverter utilizing a special circuit topology to connect interleaved half bridge legs in parallel by a network of transformers. The resulting multi-level characteristic allows to achieve effective switching frequencies of more than 100 kHz with low-cost IGBTs, or even higher frequencies using SiC-Mosfets. Due to the mutual winding directions of the transformer coils, no essential DC excitation appears in the cores, which allows a smaller transformer design. The stray inductances are maximized by using a special winding arrangement and are sufficient to form a filter inductance for the sine-wave output filter due to the high effective switching frequency. Therefore, the inverter is tolerant to high peak output currents, being relevant, e.g., for high-dynamic servo drives.

To evaluate the concept, a single-phase laboratory prototype with four IGBT bridge legs (switching frequency 25 kHz, effectively 100 kHz) for 600 V and 20 A was designed and implemented. The performance is highly satisfactory - an efficiency of more than 97 % was achieved, and the proposed peak current capability was validated by applying a 60 A-load current pulse being three times the nominal current for 500 ms. Without using a balancing control the output load current does not split in a perfect equal manner to the four legs due to component and switching delay tolerances. This was successfully remedied by including isolated shunt current sensors in each stage, and implementing a PI controller for equalizing the leg currents.

Kurzzusammenfassung

Die vorliegende Diplomarbeit behandelt die Analyse, Dimensionierung und den Aufbau eines Sinuswechselrichters, der eine spezielle Schaltungstopologie nutzt, um phasenversetzt getaktete Halbbrücken über ein Netzwerk aus Transformatoren parallel zu schalten. Die resultierende Multilevel-Charakteristik ermöglicht es, auch beim Einsatz von günstigen IGBTs effektive Schaltfrequenzen von mehr als 100 kHz zu erreichen - mit SiC-Mosfets sind entsprechend höhere Frequenzen möglich. Durch die gegengleichen Wicklungssinne der Transformatoren kommt es idealerweise zu keiner DC-Aussteuerung der Kerne, wodurch diese kleiner dimensioniert werden können. Die Streuinduktivitäten werden durch eine spezielle Wicklungsanordnung erhöht und reichen dann aufgrund der hohen effektiven Schaltfrequenz zur Nutzung als Filterinduktivität aus. Weil hierbei als magnetisches Medium primär Luft wirkt, ist der Sättigungsstrom ein Vielfaches des thermisch begrenzten Nennstromes, woraus eine hohe Überstromfestigkeit des Wechselrichters resultiert.

Zur Evaluierung des Konzeptes wurde ein einphasiger Laborprototyp mit vier IGBT-Brückenweigen (Schaltfrequenz 25 kHz, effektiv 100 kHz) für die Nenn-daten 600 V, 20 A dimensioniert und aufgebaut. Messungen zeigten, dass sich der Ausgangsstrom aufgrund von Bauteiltoleranzen nicht gleichmäßig auf die Brückenweige aufteilt, die Abweichung allerdings auf den Sättigungsstrom der Hauptinduktivität der Transformatoren beschränkt ist, und ein Betrieb daher auch ohne Regelung möglich wäre. Durch Implementierung eines Zweigstromreglers konnten die Ströme perfekt symmetriert und eine gleichmäßige Auslastung der Halbbrücken und Transformatoren erreicht werden. Der erzielte Wirkungsgrad von über 97 % und die hohe Überstromfestigkeit (3-facher Nennstrom, 60 A) des rein mit Standardkomponenten aufgebauten Laborprototypes demonstrieren die Leistungsfähigkeit des Konzeptes.

Inhaltsverzeichnis

1	Einleitung	1
2	Konzept	4
2.1	Berechnung und Simulation	7
2.1.1	Schaltfolge abcd	9
2.1.2	Schaltfolge acbd	12
2.1.3	Weitere Schaltfolgen und Vergleich	15
2.2	Nutzung der Streuinduktivitäten als Filterinduktivität	17
2.3	Symmetrie der Zweigströme	17
3	Realisierung	19
3.1	Halbbrücken	19
3.1.1	IGBT	20
3.1.2	Gatetreiber	20
3.1.3	Zwischenkreiskondensatoren	21
3.1.4	Kühlung	21
3.1.5	Platinenlayout	22
3.1.6	Messungen	22
3.2	Transformatoren	23
3.2.1	Dimensionierung und Simulation	23
3.2.2	Realisierung	25
3.2.3	Ermittlung des Sättigungsstromes	28
3.3	PWM-Erzeugung	31
3.4	Strom- und Spannungsmessung	32
3.5	Regelung	34
3.6	Ergebnis	37

4 Messungen	38
4.1 Thermische Messungen	38
4.2 Zweigspannungen- und Ströme	39
4.3 Aufteilung des Ausgangsstromes auf die Brückenzeige	40
4.4 Wirkungsgrad	42
4.5 Kurzzeitige Überlastung	43
4.6 Fazit	46
5 Zusammenfassung und Ausblick	47
A Schaltpläne	49
B Platinenlayouts	54
C Bauteillisten	61
D Software	63

Abbildungsverzeichnis

1.1	Vereinfachte Darstellung des Leistungsteiles eines Umrichterstranges mit parallel geschalteten Zweigen und LC-Ausgangsfilter. . .	1
1.2	Darstellung der in dieser Arbeit genutzten Schaltungstopologie für $N = 4$ Zweige, welche über ein Netzwerk aus gekoppelten Transformatoren parallel geschaltet werden.	2
2.1	Eine Saugdrossel als Einzelement betrachtet. (a) Ausgangsschaltung, die Spannungsquellen stellen die Ausgänge der Halbbrücken dar, (b) Ersatzschalbild.	4
2.2	Vereinfachte Darstellung des Konzeptes mit allen Spannungen und Strömen, die in der Berechnung verwendet werden.	8
2.3	Simulationsergebnisse für Schaltfolge 'abcd', von oben nach unten: Halbbrücken-Ausgangsspannungen, Spannung an einer Transformatorwicklung, Fluss im Eisenkern eines Transformators, Strom im Zweig 'a', und Ausgangsstrom.	10
2.4	Simulationsergebnisse für Schaltfolge 'acbd', von oben nach unten: Halbbrücken-Ausgangsspannungen, Spannung an einer Transformatorwicklung, Fluss im Eisenkern eines Transformators, Strom im Zweig 'a', und Ausgangsstrom bei Schaltfolge 'acbd'. Die Daten wurden mit denselben Simulationsparametern wie in Abbildung 2.3 erzeugt.	13
2.5	Zwischenzweig Rippelstrom (Peak-Peak) bei Schaltfolge 'abcd' und 'acbd'. Es handelt sich nur um den Rippelstrom zwischen den Zweigen, der Anteil des Ausgangsstromes ($i_2/4$) ist nicht enthalten.	16
2.6	Maximaler Fluss in den Transformator-kernen (Peak-Peak) bei Schaltfolge 'abcd' und 'acbd'.	16
3.1	Ausschnitt aus dem Schaltplan der Halbbrücken-Module.	19
3.2	Steigende- und fallende Flanke des Ausgangssignales einer Halbbrücke, unbelastet bei Nennspannung.	23
3.3	Simulation der Flusdichteverteilung in den Transformatoren für die Hauptinduktivität (links) und Streuinduktivität (rechts). . .	25
3.4	Fotos des Transformators. Die Wicklungen sind räumlich so weit wie möglich getrennt angeordnet, um die Streuinduktivität zu erhöhen.	26

3.5	Schaltungsarten für die Messungen mit dem LCR-Meter und die Stoßstrommessungen. (a) dient zur Messung der Hauptinduktivität, (b) zur Messung der Streuinduktivität, und (c) zur direkten Messung der am Ausgang wirksamen Induktivität.	27
3.6	Versuchsaufbau für die Stoßstrommessungen an den Transformatoren.	28
3.7	Stoßstrommessung an der Hauptinduktivität nach Schaltungsart (a).	29
3.8	Stoßstrommessung an der Streuinduktivität nach Schaltungsart (b).	30
3.9	Stoßstrommessung an der Parallelschaltung aller vier Drosseln nach Schaltungsart (c).	30
3.10	Schaltungsteil zur Messung des Zweigstromes i_a	32
3.11	Schaltungsteil zur Addition der vier Zweigströme zum Gesamtstrom i_2	33
3.12	Schaltungsteil zur Messung der Ausgangsspannung u_2	34
3.13	Schema von Zweigstrom-, Gesamtstrom- und Spannungsregelung.	35
3.14	Foto des Versuchsaufbaues. Im Hintergrund sind die vier Halbbrücken-Module auf einem gemeinsamen Kühlkörper zu sehen. Der Vordergrund zeigt die Platine mit Transformatoren, Ausgangsfilter-Kondensatoren, und Strom- und Spannungs-Messwandlern. Die rote Platine rechts ist das DSP Evaluation Board.	37
4.1	Foto und Wärmebild des Versuchsaufbaues nach 30 Minuten Betrieb bei halber Nennleistung.	38
4.2	Steigende- und fallende Flanke des Ausgangssignales einer Halbbrücke bei angeschlossenem Transformatornetzwerk, unbelastet und belastet.	39
4.3	Ausgangsstrom und -spannung einer Halbbrücke bei angeschlossenem Transformatornetzwerk, unbelastet (links) und belastet mit $2,7\ \Omega$, $i_2 \approx 20\ \text{A}$ (rechts).	40
4.4	Zweigströme ohne Regelung abhängig vom Sekundärstrom.	41
4.5	Zweigströme mit Regelung abhängig vom Sekundärstrom.	42
4.6	Wirkungsgrad über Sekundärstrom, gemessen bei $d = 0,5$ und Variation des Sekundärstromes über eine Widerstandsdekade.	43
4.7	Schaltbild der Last, die für die Stoßstrommessung verwendet wurde. $R_1 = 13,2\ \Omega$ stellt die Grundlast dar, zu der kurzzeitig $R_2 = 1,15\ \Omega$ parallel geschaltet wird.	44
4.8	Stoßstrommessung, Gesamtstrom i_2	45
4.9	Stoßstrommessung, Zweigströme ohne Zweigstromregler.	45
4.10	Stoßstrommessung, Zweigströme mit Zweigstromregler.	46

Tabellenverzeichnis

2.1	Auslegungsdaten des einsträngigen Laborprototypes.	7
3.1	Kenndaten des ausgewählten IGBT Typs. Quelle: [8]	20
3.2	Ergebnisse zu den Einzelmessungen der Saugdrosseln mit dem LCR-Meter gemäß den Schaltungsarten aus Abbildung 3.5. Verwendetes Messgerät: Hioki 3532-50 LCR HiTester	27
C.1	Bauteilliste der Halbbrücken-Module. Mengenangaben pro Modul, es werden vier Module benötigt.	61
C.2	Bauteilliste der Messplatine. Es wird eine Platine benötigt.	62

1 Einleitung

Aktuelle Entwicklungen zeigen, dass der Gesamtwirkungsgrad von Antriebssystemen höher ist, wenn der Motor von Umrichtern mit Sinusausgangsspannungen anstelle von konventionellen PWM-Umrichtern gespeist wird [1]. Die geringere Flankensteilheit führt außerdem zu einer kleineren EMV Belastung, was den Einsatz von einfachen, ungeschirmten Motorkabeln über große Entfernungen ermöglicht [2], und kleineren Verzerrungen aufgrund frequenzabhängiger Signallaufzeiten bei langen Motorkabeln [3]. Weitere Vorteile sind der geräuscharme Betrieb, und eine gesteigerte Motorlebensdauer aufgrund der geringeren Isolationsbelastung sowie reduzierter Lagerströme.

Sinusausgangsspannungen werden in der Regel durch Einfügen eines LC-Filters zwischen einem konventionellen PWM-Wechselrichter und der Last erreicht (Abb. 1.1), wobei die Filterinduktivität aber aufgrund der vergleichsweise geringen Schaltfrequenz von IGBT-Wechselrichtern groß und teuer wird. Dies gilt insbesondere, wenn ein hoher Spitzenstrom bereitgestellt werden muss, z.B. bei hochdynamischen Antrieben, da die Filterinduktivität magnetisch auf Spitzenstrombelastung ausgelegt werden muss.

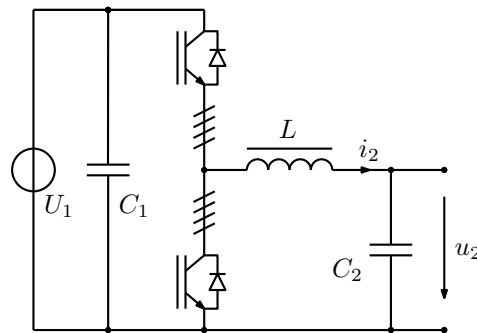


Abbildung 1.1: Vereinfachte Darstellung des Leistungsteiles eines Umrichterstranges mit parallel geschalteten Zweigen und LC-Ausgangsfilter.

In dieser Arbeit wird eine Schaltungsart präsentiert und evaluiert, die es ermöglicht, die Frequenz des Ausgangsstrom-Rippels ohne Veränderung der Schaltfrequenz zu vervielfachen, was eine Verkleinerung der Filterinduktivität ermöglicht.

Jeder Wechselrichter-Brückenstrang wird dabei in N parallele Zweige aufgeteilt, wie es bereits gängige Praxis ist, um hohe Ausgangsleistungen mit Standardkomponenten zu erreichen. Anstatt die einzelnen Zweige aber hart parallelzuschalten, werden sie über ein Netzwerk von Transformatoren gekoppelt und mit phasenversetzten PWM Signalen angesteuert (Abb. 1.2). Dabei wird versucht, die Streuinduktivität der Transformatoren als Ausgangsfilter-Induktivität zu nutzen, wodurch eine sehr hohe Überstromfestigkeit erreicht werden kann.

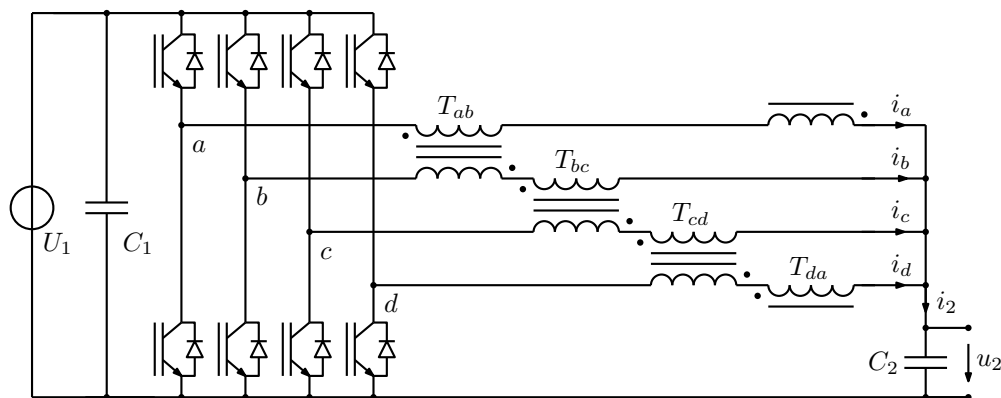


Abbildung 1.2: Darstellung der in dieser Arbeit genutzten Schaltungstopologie für $N = 4$ Zweige, welche über ein Netzwerk aus gekoppelten Transformatoren parallel geschaltet werden.

Weil sich der Ausgangsstrom auf N Zweige aufteilt, werden die Drosseln und IGBTs für den Strom i_2/N dimensioniert. Aufgrund der gegengleichen Wicklungssinne der Transformatorwicklungen kommt es im Idealfall zu keiner DC-Aussteuerung, die Kerne müssen lediglich über die Differenzspannung der Zweige dimensioniert werden. Wegen der Phasenverschiebung ist die Frequenz des Zweig- und Ausgangsstromrippels um den Faktor N höher als die Schaltfrequenz, und die Amplitude um den Faktor N verringert, wodurch der Kernquerschnitt kleiner dimensioniert werden kann.

Die Streuinduktivitäten der Transformatoren werden in Verbindung mit einem Folienkondensator C_2 als Ausgangsfilter genutzt. Weil als magnetisches Medium der Streupfade hauptsächlich Luft wirkt, ist der Sättigungsstrom deutlich höher als der thermisch begrenzte Nennstrom der Transformatoren, woraus eine hohe Überstromfestigkeit resultiert. Die vergleichsweise geringe Induktivität wird durch eine speziell gewählte Wicklungsanordnung erhöht, und reicht aufgrund der Multilevelcharakteristik des Umrichters zur Glättung aus.

Im Zuge dieser Arbeit wird die beschriebene Schaltung für $N = 4$ Zweige analysiert und simuliert, und die Auswirkungen von unterschiedlichen Schaltfolgen der PWM-Signale, der Rippelstrom zwischen den Zweigen, etc. behandelt. Zur

Überprüfung des realen Verhaltens wird ein Laborprototyp für die Nenndaten $U_2 = 600 \text{ V}$, $I_2 = 20 \text{ A}$ dimensioniert und aufgebaut, und besonders auf die Auslegung der Transformatoren eingegangen. Um der durch Bauteiltoleranzen verursachte ungleichmäßige Aufteilung des Ausgangsstromes auf die Zweige entgegenzuwirken, wird ein Zweigstromregler implementiert. Zuletzt werden am Prototyp Messungen zur Überstromfestigkeit, Stromaufteilung und Wirkungsgrad durchgeführt.

2 Konzept

Die in Abbildung 1.2 dargestellte Schaltung basiert auf der Funktion einer ‘Saugdrossel’. Es handelt sich dabei um eine Drossel mit Mittelanzapfung, bzw. um einen Transformator, dessen Primär- und Sekundärwicklung galvanisch verbunden sind (Spartransformator).

Parallelschalten von zwei Halbbrückenzeigen

Im Folgenden wird eine einzelne Saugdrossel, mit der zwei Halbbrückenzeige (dargestellt als Spannungsquellen) parallelgeschaltet werden können, als Einzelelement betrachtet (Abbildung 2.1a) und ihre Wirkungsweise analysiert.

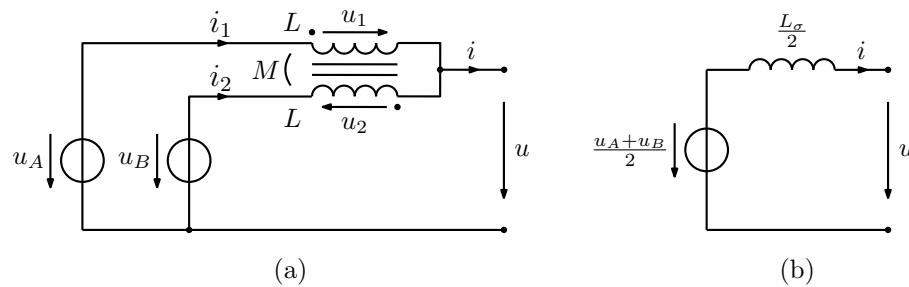


Abbildung 2.1: Eine Saugdrossel als Einzelelement betrachtet. (a) Ausgangsschaltung, die Spannungsquellen stellen die Ausgänge der Halbbrücken dar, (b) Ersatzschalbild.

Die Transformatorgleichungen können mit der Eigeninduktivität L und der Gegeninduktivität M geschrieben werden als

$$\begin{aligned} u_1 &= L \frac{di_1}{dt} - M \frac{di_2}{dt} \\ u_2 &= M \frac{di_1}{dt} - L \frac{di_2}{dt} \end{aligned} \quad (2.1)$$

Deren Subtraktion führt mit $i = i_1 + i_2$ und $L_\sigma = L - M$ auf

$$u_1 - u_2 = (L - M) \left(\frac{di_1}{dt} + \frac{di_2}{dt} \right) = L_\sigma \frac{di}{dt} \quad (2.2)$$

Die Maschengleichungen sind

$$\begin{aligned} u_A &= u + u_1 \\ u_B &= u - u_2 \end{aligned} \quad (2.3)$$

Die Addition ergibt zusammen mit dem Ergebnis aus Gleichung 2.2

$$u = \frac{u_A + u_B}{2} - \frac{u_1 - u_2}{2} = \frac{u_A + u_B}{2} - \frac{L_\sigma}{2} \frac{di}{dt} \quad (2.4)$$

Gleichung 2.4 beschreibt die in Abbildung 2.1b dargestellte Ersatzspannungsquelle. Die Saugdrossel bildet den Mittelwert der beiden Eingangsspannungen. Als Ausgangsimpedanz wirkt die halbe Streuinduktivität, die Hauptinduktivität ist am Ausgang wirkungslos. Aufgrund dieser Eigenschaften können phasenversetzt getaktete Halbbrücken mit Hilfe von Saugdrosseln parallelgeschaltet werden.

Betrachtet man die magnetische Induktion im Kern unter der Annahme eines Luftspaltes der Länge l und gleicher Windungszahl N der beiden Spulen, ergibt sich aufgrund der Wicklungssinne

$$\begin{aligned} \Theta &= Hl = \frac{B}{\mu_0} l = N i_1 - N i_2 \\ B &= \frac{N \mu_0}{l} (i_1 - i_2) \end{aligned} \quad (2.5)$$

Die Induktion wird also nur durch die Stromdifferenz ($i_1 - i_2$) erzeugt. Diese Ströme können in einen Gleichanteil $i/2$ und einen Wechselanteil i_{AC} aufgeteilt werden.

$$\begin{aligned} i_1 &= \frac{i}{2} + i_{AC} \\ i_2 &= \frac{i}{2} - i_{AC} \end{aligned} \quad (2.6)$$

Der Gleichanteil fließt über den Ausgang ab ($i_1 + i_2 = i$) und erzeugt keine Induktion im Kern. Der Wechselanteil wird von der Saugdrossel aufgenommen und ist laut Gleichung 2.5 relevant für die Dimensionierung des Kernes ($i_1 - i_2 = 2 i_{AC}$).

Parallelschalten von n Halbbrückenzeigen

Die beschriebene Schaltung kann sehr einfach für die Addition von 2^n Zweigen erweitert werden, indem immer zwei Zweige mit einer Saugdrossel addiert werden, deren Ausgänge mit einer weiteren Stufe addiert werden. Es ergibt sich eine Kaskade an Saugdrosseln ('Whiffletree Configuration'), wobei aber jede Stufe anders dimensioniert werden muss, und die Anzahl der Zweige eine Potenz von 2 sein muss.

[4] präsentiert weitere Schaltungen, die eine höhere Anzahl an Saugdrosseln/Transformatoren erfordern, aber andere Vorteile haben. Es wurde im Rahmen dieser Arbeit die in Abbildung 1.2 dargestellte 'Cyclic Cascade Configuration' gewählt, bei der die Anzahl der Zweige nicht auf die Potenzen von 2 beschränkt ist, sondern eine beliebige natürliche Zahl sein kann - für die Addition von N Brückenzeigen werden immer N Transformatoren benötigt. Das PWM Signal zur Ansteuerung der Halbbrücken ist um T/N phasenverschoben. Alle Halbbrücken und Transformatoren werden thermisch auf denselben Strom i_2/N ausgelegt. Im Folgenden wird der Begriff 'Transformator' verwendet, weil von beiden Spulen beide Anschlüsse herausgeführt sein müssen, und das Bauteil daher nicht mehr als Induktivität mit Mittelanzapfung dargestellt werden kann.

Ein weiterer Vorteil ist, dass (zumindest bei bestimmten Schaltfolgen, mehr dazu siehe Kapitel 2.1) alle Transformatoren gleich ausgelastet werden, und daher gleich ausgelegt werden können, was fertigungstechnische Vorteile mit sich bringt.

Die Streuinduktivität ist auch bei dieser Schaltung als $L_\sigma/2$ am Ausgang wirksam und kann daher vergleichsweise gut als Ausgangsfilterinduktivität genutzt werden.

Wie bei der klassischen Saugdrossel erzeugen die DC-Anteile der Zweigströme aufgrund der gegenseitigen Wicklungssinne der Transformatorenwindungen keine magnetische Durchflutung im Transformator Kern, so lange sie gleich groß sind. Für die Implementierung einer Regelung können die Zweigströme am Punkt der Zusammenführung (Strompfeile in Abbildung 1.2) gut mit Shunts gemessen werden. Dieser Punkt liegt am Ausgangskondensator, und somit auf nicht schaltendem Potential.

Definition der Auslegungsdaten

Für die folgende Schaltungsanalyse, Dimensionierung und den Bau eines Prototypen wird $N = 4$ gewählt. Die weiteren Auslegungsdaten für einen einzelnen Umrichter-Strang sind Tabelle 2.1 zu entnehmen.

Zwischenkreisspannung U_2	600 V
Ausgangsstrom I_2	20 A
Nennleistung	12 kW
Schaltfrequenz der Halbbrücken f_S	25 kHz
Schaltertyp	IGBT

Tabelle 2.1: Auslegungsdaten des einsträngigen Laborprototypes.

Alle Komponenten werden aufgrund der hohen Nennspannung von 600 V und den üblicherweise auftretenden Schaltüberspannungen und Überspannungen durch Rückspeisung (im Falle der Nutzung als Antriebs-Umrichter) für eine Spannung von mindestens 1000 V dimensioniert. Mosfets sind wegen des hohen Kanalwiderstandes und daraus folgender Verlustleistung ab 500 V Durchbruchsspannung nicht sinnvoll einsetzbar. SiC-Mosfets haben geringere Kanalwiderstände bei hohen Spannungen, sind aber momentan noch sehr teuer, und wegen der nicht sehr hohen Schaltfrequenz von 25 kHz nicht erforderlich. IGBTs sind günstig und robust, und haben bei Spannungen ab 500 V geringere Leitverluste als Mosfets. Die Schaltverluste sind höher, trotzdem können IGBTs aber bis etwa 25 kHz sinnvoll eingesetzt werden. (Quelle: [5])

Es werden daher IGBTs mit einer Durchbruchspannung von 1200 V verwendet, die mit 25 kHz betrieben werden. Bei vier Halbbrücken ergibt sich am Ausgang eine Stromrippelfrequenz von 100 kHz.

Die Spannungsversorgung des Prototyps erfolgt über Gleichspannungs-Labornetzteile mit Stützkondensatoren C_1 . Die Streuinduktivitäten der Transformatoren bilden mit dem Folienkondensator C_2 ein Ausgangsfilter. Sie wirken wie Drosseln mit sehr großem Luftspalt und sättigen daher erst bei hohen Strömen, was für das Ausgangsfilter vorteilhaft ist. Bei der Auslegung der Saugdrosseln wird daher versucht, eine möglichst hohe Streuinduktivität zu erhalten.

2.1 Berechnung und Simulation

Zur Dimensionierung der Transformatoren wird die Funktionsweise der Schaltung analysiert, und der Fluss sowie der Rippelstrom in den Drosseln bestimmt. Abhängig vom Tastverhältnis und der Schaltreihenfolge der Zweige ergeben sich verschiedene Betriebszustände, die im folgenden analytisch betrachtet werden.

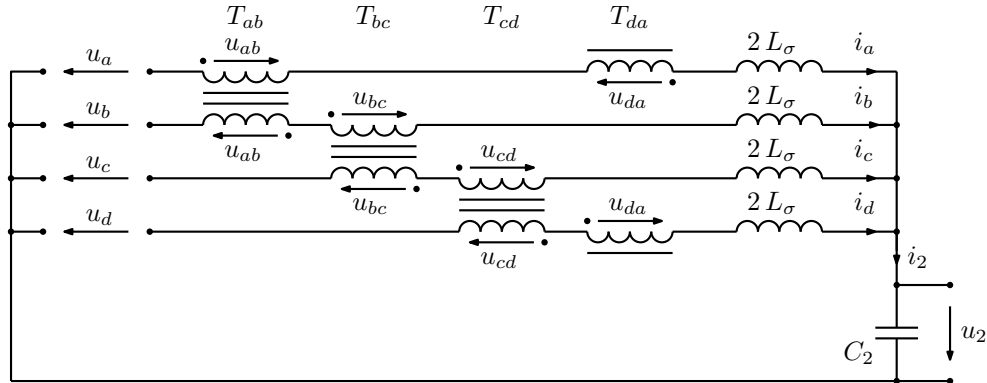


Abbildung 2.2: Vereinfachte Darstellung des Konzeptes mit allen Spannungen und Strömen, die in der Berechnung verwendet werden.

Abbildung 2.2 zeigt eine vereinfachte Darstellung des Konzeptes. Die Gleichungen zur Beschreibung der Schaltung lauten wie folgt, wobei L_H die Hauptinduktivität, M die Gegeninduktivität, und L_σ die Streuinduktivität der Saugdrosseln bezeichnen. Für die Schaltungsanalyse wird zur Vereinfachung ideale Kopplung, also $M = L_H$ angenommen. Die Streuinduktivität der Transformatoren wird durch eine separate Induktivität L_σ in Serie zu jeder Saugdrosselwicklung dargestellt. Der exakte Kopplungsgrad wird nach dem Bau der Transformatoren mit Hilfe von Messungen bestimmt (siehe Kapitel 3.2).

$$\begin{aligned}
 u_a &= -u_{da} + u_{ab} + 2u_{a\sigma} + U_2 \\
 u_b &= -u_{ab} + u_{bc} + 2u_{b\sigma} + U_2 \\
 u_c &= -u_{bc} + u_{cd} + 2u_{c\sigma} + U_2 \\
 u_d &= -u_{cd} + u_{da} + 2u_{d\sigma} + U_2
 \end{aligned} \tag{2.7}$$

$$\begin{aligned}
 u_{ab} &= L_H \dot{i}_a - M \dot{i}_b = L_H (\dot{i}_a - \dot{i}_b) \\
 u_{bc} &= L_H \dot{i}_b - M \dot{i}_c = L_H (\dot{i}_b - \dot{i}_c) \\
 u_{cd} &= L_H \dot{i}_c - M \dot{i}_d = L_H (\dot{i}_c - \dot{i}_d) \\
 u_{da} &= L_H \dot{i}_d - M \dot{i}_a = L_H (\dot{i}_d - \dot{i}_a)
 \end{aligned} \tag{2.8}$$

$$\begin{aligned}
 u_{a\sigma} &= L_\sigma \dot{i}_a \\
 u_{b\sigma} &= L_\sigma \dot{i}_b \\
 u_{c\sigma} &= L_\sigma \dot{i}_c \\
 u_{d\sigma} &= L_\sigma \dot{i}_d
 \end{aligned} \tag{2.9}$$

Der Ausgangsstrom ergibt sich zu

$$i_2 = i_a + i_b + i_c + i_d \quad (2.10)$$

Aus diesen Gleichungen können die Stromanstiegsraten der Zweigströme $i_a \dots i_d$ für verschiedene Betriebsfälle berechnet werden. Die Betriebsfälle unterscheiden sich durch die Schaltfolge der Zweige und das Tastverhältnis.

2.1.1 Schaltfolge abcd

Abbildung 2.3 zeigt in der ersten Zeile die Eingangsspannungen bei der Schaltfolge 'abcd'. Darunter sind die Ergebnisse einer Spice-Simulation für die Spannung an einer Drossel, den daraus durch Integration berechneten Fluss einer Drossel, und der Rippelstrom eines Zweiges dargestellt. Für alle folgenden Simulationen wurden die Parameter $f_S = 25 \text{ kHz}$, $U_1 = 600 \text{ V}$, $L_H = 2 \text{ mH}$, $L_\sigma = 100 \text{ uH}$, $C_2 = 10 \text{ uF}$, und $R_L = 30 \Omega$ gewählt.

Es wurden vier Fälle ausgewählt, durch die das Verhalten über den gesamten Arbeitsbereich beschrieben werden kann. Im Folgenden werden diese Fälle analytisch betrachtet.

Fall 1: $d < 0,25$ Es sind abwechselnd keiner oder nur ein Brückenweig aktiv, in diesem Beispiel ist Zweig 'a' aktiv:

$$\begin{aligned} u_a &= U_1 \\ u_b &= u_c = u_d = 0 \end{aligned} \quad (2.11)$$

Die Stromanstiegsraten ergeben sich zu

$$\begin{aligned} \dot{i}_a &= U_1 \frac{L_H^2 + 4L_H L_\sigma + 2L_S^2}{4L_\sigma(L_H + L_\sigma)(2L_H + L_\sigma)} - U_2 \frac{1}{2L_\sigma} = U_1 \frac{5L_H + 3L_\sigma}{8(L_H + L_\sigma)(2L_H + L_\sigma)} + \frac{\dot{i}_2}{4} \\ \dot{i}_b &= U_1 \frac{L_H}{4L_\sigma(2L_H + L_\sigma)} - U_2 \frac{1}{2L_\sigma} = -U_1 \frac{1}{8(2L_H + L_\sigma)} + \frac{\dot{i}_2}{4} \\ \dot{i}_c &= U_1 \frac{L_H^2}{4L_\sigma(L_H + L_\sigma)(2L_H + L_\sigma)} - U_2 \frac{1}{2L_\sigma} = -U_1 \frac{3L_H + L_\sigma}{8(2L_H^2 + 3L_H L_\sigma + L_\sigma^2)} + \frac{\dot{i}_2}{4} \\ \dot{i}_d &= \dot{i}_b \\ \dot{i}_2 &= \frac{1}{2L_\sigma} (U_1 - 4U_2) \end{aligned} \quad (2.12)$$

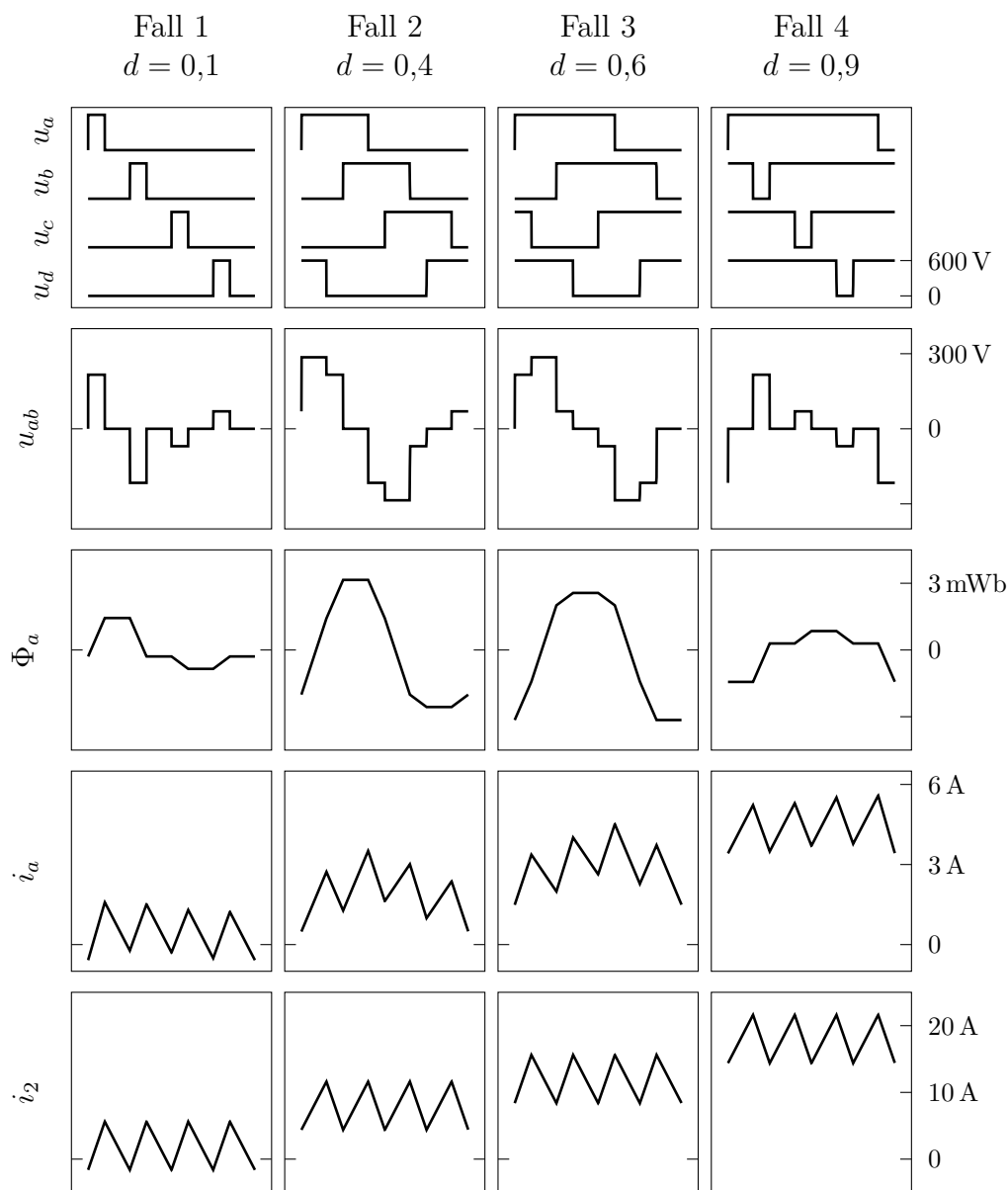


Abbildung 2.3: Simulationsergebnisse für Schaltfolge 'abcd', von oben nach unten: Halbbrücken-Ausgangsspannungen, Spannung an einer Transformatorwicklung, Fluss im Eisenkern eines Transformators, Strom im Zweig 'a', und Ausgangsstrom.

Der aktive Zweig 'a' hat die höchste Anstiegsrate. Die beiden benachbarten, also direkt mit ihm gekoppelten Zweige 'b' und 'd' haben gleiche, etwas geringere Anstiegsraten. Im ungekoppelten Zweig 'c' ist die Anstiegsrate am geringsten. Aus diesen Ergebnissen können der Stromrippel und der Fluss in den Saugdrosseln berechnet werden, die als Ausgangswerte zur Dimensionierung der Drosseln dienen (siehe Kapitel 3.2).

Zum besseren Vergleich wurden die Stromanstiege als Summe von $\dot{i}_2/4$ und einem zweigspezifischen Term dargestellt. Der Rippelstrom \dot{i}_2 tritt am Ausgang auf und wird vom Ausgangskondensator aufgenommen. Von diesem Teil kann jedem Zweig ein Viertel zugeordnet werden. Er ist für die Dimensionierung der Drosseln nicht relevant, weil er in allen Zweigen gleichzeitig fließt, und sich der von ihm erzeugte Fluss aufgrund des gegengleichen Wicklungssinnes der Drosseln aufhebt.

Der zweigspezifische Term fließt zwischen den Zweigen und ist für die Dimensionierung der Drosseln wichtig. Den höchsten Wert hat der aktive Zweig 'a', aus diesem wird durch Integration der Zweigstromrippel Δi (Peak-Peak) berechnet:

$$\Delta i_{abcd,1} = U_1 T d \frac{5L_H + 3L_\sigma}{8(L_H + L_\sigma)(L_H + 2L_\sigma)} \quad (2.13)$$

Der Fluss wird durch Einsetzen der Stromänderungsraten in das Gleichungssystem 2.8 und Integration der so erhaltenen Transformatorspannungen über die Zeit berechnet. Der Peak-Peak Wert des Flusses ergibt sich zu

$$\Delta \Phi_{abcd,1} = U_1 d T \frac{1}{2} \frac{L_H}{L_H + L_\sigma} \quad (2.14)$$

Fall 2: $0,25 < d < 0,5$ Es sind abwechselnd nur ein Brückenweig, oder zwei benachbarte, also gekoppelte Brückenweige gleichzeitig aktiv. Ein Beispiel für zwei benachbarte aktive Zweige ist

$$\begin{aligned} u_a &= u_b = U_1 \\ u_c &= u_d = 0 \end{aligned} \quad (2.15)$$

In diesem Fall sind die Stromanstiegsraten:

$$\begin{aligned} \dot{i}_a &= U_1 \frac{L_H + 2L_\sigma}{4L_\sigma(L_H + L_\sigma)} - U_2 \frac{1}{2L_\sigma} = U_1 \frac{1}{4(L_H + L_\sigma)} + \frac{\dot{i}_2}{4} \\ \dot{i}_b &= \dot{i}_a \\ \dot{i}_c &= U_1 \frac{L_H}{4L_\sigma(L_H + L_\sigma)} - U_2 \frac{1}{2L_\sigma} = -U_1 \frac{1}{4(L_H + L_\sigma)} + \frac{\dot{i}_2}{4} \\ \dot{i}_d &= \dot{i}_c \\ \dot{i}_2 &= \frac{1}{L_\sigma} (U_1 - 2U_2) \end{aligned} \quad (2.16)$$

Der maximale Zweigstromrippel ist

$$\Delta i_{abcd,2} = U_1 T \left[\left(\frac{1}{2} - d \right) \frac{5L_H + 3L_\sigma}{8(L_H + L_\sigma)(L_H + 2L_\sigma)} + \left(d - \frac{1}{4} \right) \frac{1}{2(L_H + L_S)} \right] \quad (2.17)$$

Der maximale Fluss ist

$$\Delta \Phi_{abcd,2} = U_1 T \frac{1}{8} \frac{L_H}{L_H + L_\sigma} \quad (2.18)$$

Fall 3: $0,5 < d < 0,75$ Es sind abwechselnd zwei oder drei benachbarte Zweige aktiv. Der Fall kann als invertierter Fall 2 gesehen werden und wird deshalb nicht behandelt.

Fall 4: $d > 0,75$ Es sind abwechselnd drei oder alle vier Zweige aktiv. Der Fall kann als invertierter Fall 1 gesehen werden und wird deshalb nicht behandelt.

2.1.2 Schaltfolge acbd

Abbildung 2.4 zeigt die Eingangsspannungen und Simulationsergebnisse bei Schaltfolge ‘acbd’. Sie wurden mit denselben Parametern wie die Daten aus Abbildung 2.3 erzeugt.

Aufgrund der anderen Schaltfolge sind jetzt in manchen Fällen auch zwei Zweige gleichzeitig aktiv, die nicht gekoppelt sind. Dadurch ergeben sich an den Spulen T_1 und T_3 andere Ergebnisse als an den Transformatoren T_2 und T_4 . Dies wird im folgenden anhand der Selben vier Fälle genauer untersucht.

Fall 1: $d < 0,25$ Die Stromanstiegsraten und Zweigstromrippel sind gleich wie bei Schaltfolge ‘abcd’, weil immer nur ein Zweig aktiv und die Reihenfolge daher belanglos ist.

Bei der Berechnung des Flusses über die Integration der Spannungen spielt aber die Reihenfolge der Spannung-Zeit-Flächen eine wichtige Rolle. Beispielsweise ergibt sich ein höherer Peak-Peak Wert des Flusses, wenn hintereinander zwei positive Spannungsimpulse anliegen, als wenn ein positiver und ein negativer abwechselnd anliegt. Dies ist auch ersichtlich, wenn Abbildung 2.4 Der maximale Fluss ist bei allen vier Drosseln gleich, aber anders als bei Schaltfolge ‘abcd’:

$$\Delta\Phi_{acbd,1} = U_1 T d \frac{1}{4} \frac{L_H(3L_H + 2L_\sigma)}{(2L_H + L_\sigma)(L_H + L_\sigma)} \quad (2.19)$$

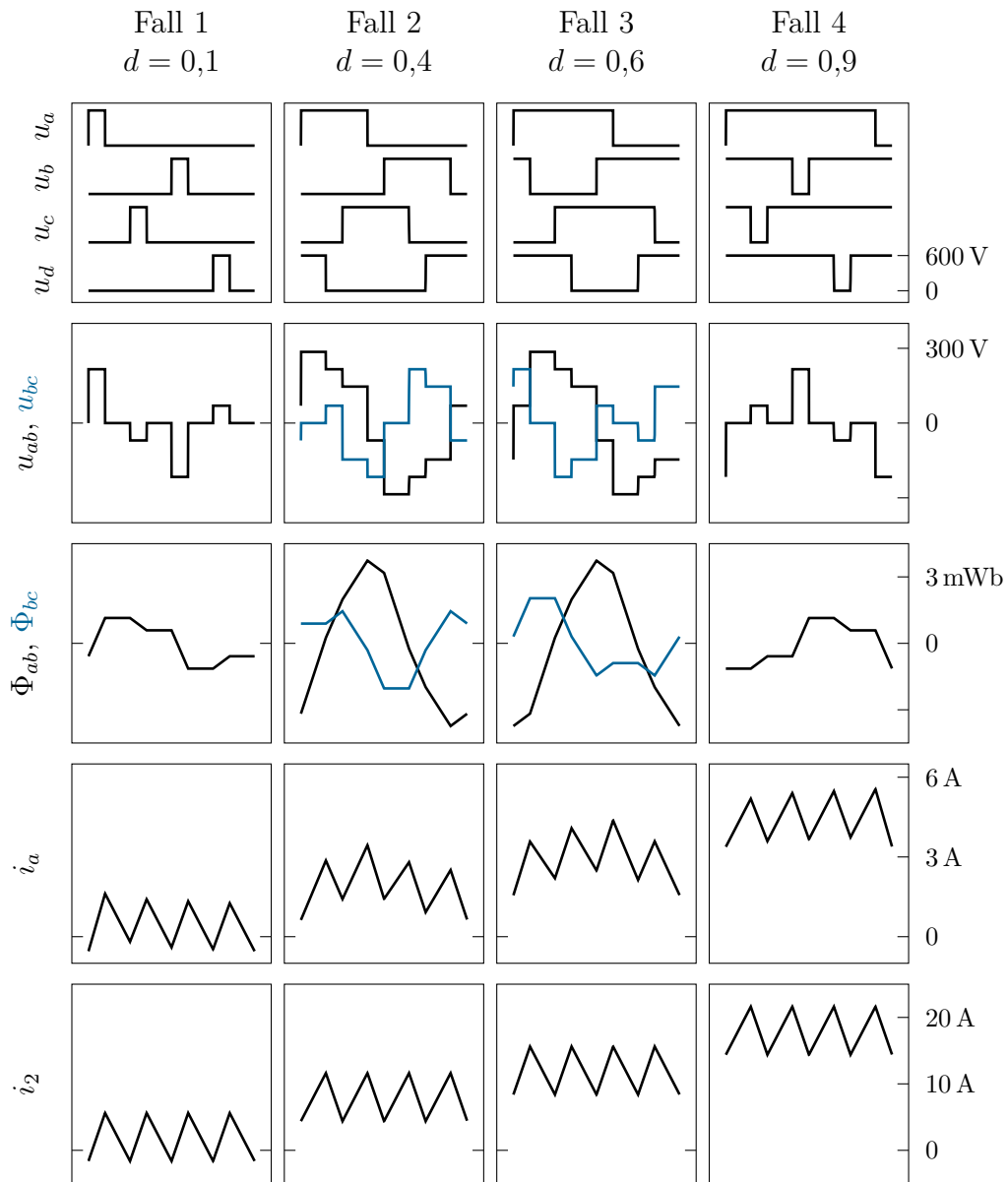


Abbildung 2.4: Simulationsergebnisse für Schaltfolge ‘acbd’, von oben nach unten: Halbbrücken-Ausgangsspannungen, Spannung an einer Transformatorwicklung, Fluss im Eisenkern eines Transformators, Strom im Zweig ‘a’, und Ausgangsstrom bei Schaltfolge ‘acbd’. Die Daten wurden mit denselben Simulationsparametern wie in Abbildung 2.3 erzeugt.

Fall 2: $0,25 < d < 0,5$ Durch die andere Schaltfolge können in Fall 2 auch zwei nicht benachbarte, und daher nicht gekoppelte Zweige gleichzeitig aktiv sein. Betrachtet wird der Fall

$$\begin{aligned} u_a &= u_c = U_1 \\ u_b &= u_d = 0 \end{aligned} \quad (2.20)$$

Es ergeben sich die Stromanstiegsraten zu

$$\begin{aligned} \dot{i}_a &= U_1 \frac{L_H + L_\sigma}{2L_\sigma(2L_H + L_\sigma)} - U_2 \frac{1}{2L_\sigma} = U_1 \frac{1}{4(2L_H + L_\sigma)} + \frac{\dot{i}_2}{4} \\ \dot{i}_b &= U_1 \frac{L_H}{2L_\sigma(2L_H + L_\sigma)} - U_2 \frac{1}{2L_\sigma} = -U_1 \frac{1}{4(2L_H + L_\sigma)} + \frac{\dot{i}_2}{4} \\ \dot{i}_c &= \dot{i}_a \\ \dot{i}_d &= \dot{i}_b \\ \dot{i}_2 &= \frac{1}{L_\sigma}(U_1 - 2U_2) \end{aligned} \quad (2.21)$$

Der maximale Zweigstromrippel ist

$$\begin{aligned} \Delta i_{abcd,2} &= U_1 T \left[\left(\frac{1}{2} - d \right) \frac{5L_H + 3L_\sigma}{8(L_H + L_\sigma)(L_H + 2L_\sigma)} \right. \\ &\quad \left. + \left(d - \frac{1}{4} \right) \frac{1}{4(L_H + L_\sigma)} + \left(d - \frac{1}{4} \right) \frac{1}{4(2L_H + L_\sigma)} \right] \end{aligned} \quad (2.22)$$

Der maximale Fluss ist jetzt nicht mehr bei allen vier Transformatoren gleich. Weil auch Zweige gleichzeitig aktiv sind, die nicht direkt miteinander gekoppelt sind, ergeben sich an T_{ab} und T_{cd} andere Spannung-Zeit-Flächen als an T_{bc} und T_{da} . Die Integration führt somit auf verschiedene Flüsse, und daher unterschiedliche magnetische Auslastungen der Drosseln.

$$\begin{aligned} \Delta \Phi_{abcd,2,L1,L3} &= U_1 T \frac{1}{8} \frac{L_H(L_H + 4dL_H + 4dL_S)}{(L_H + L_\sigma)(2L_H + L_\sigma)} \\ \Delta \Phi_{abcd,2,L2,L4} &= U_1 T \frac{1}{8} \frac{L_H(2L_H - 2dL_H + L_S)}{(L_H + L_\sigma)(2L_H + L_\sigma)} \end{aligned} \quad (2.23)$$

Fall 3: $0,5 < d < 0,75$ Der Fall kann wie bei Schaltfolge 'abcd' als invertierter Fall 2 gesehen werden und wird deshalb nicht behandelt.

Fall 4: $d > 0,75$ Der Fall kann wie bei Schaltfolge ‘abcd’ als invertierter Fall 1 gesehen werden und wird deshalb nicht behandelt.

2.1.3 Weitere Schaltfolgen und Vergleich

Es gibt 24 Permutationen der Schaltfolge ‘abcd’. Weil die Zweige zyklisch geschaltet werden ergibt aber beispielsweise die Abfolge ‘dabc’ dasselbe Ergebnis wie ‘abcd’. Es können daher alle Schaltfolgen, die nicht mit ‘a’ beginnen, verworfen werden, und es verbleiben sechs Folgen, von denen die ersten beiden bereits behandelt wurden:

abcd
acbd
abdc
acdb
adbc
adcb

Die letzten drei Schaltfolgen entsprechen der umgekehrten Reihenfolge der ersten drei, und können daher ebenfalls verworfen werden. Es verbleibt nur noch ‘abdc’ als letzte unbehandelte Schaltfolge. Bei dieser können genauso wie in der bereits behandelten Schaltfolge ‘acbd’ in Fall 2 auch zwei nicht benachbarte, und daher auch nicht gekoppelte Zweige gleichzeitig aktiv sein. Es wird erwartet, dass die Abfolge zu denselben Ergebnissen wie ‘acbd’ führt, was durch eine Spice-Simulation bestätigt wurde.

Bei vier Zweigen existieren daher nur zwei Schaltfolgen, die zu unterschiedlichen Ergebnissen führen. Zu deren Vergleich wird der maximale Zweigstromrippel und der maximale Fluss über dem Tastverhältnis dargestellt.

Abbildung 2.5 zeigt den Zweigstromrippel bei beiden Schaltfolgen. Im Bereich $d < 0,25$ (Fall 1) ist der Zweigstromrippel exakt gleich. Im Bereich $0,25 < d < 0,5$ (Fall 2) ist er bei Schaltungsart ‘abcd’ deutlich höher. Der Verlauf ist um $d = 0,5$ gespiegelt, weil Fall 3 dem invertierten Fall 2, und Fall 4 dem invertierten Fall 1 entspricht.

Der Fluss (Abbildung 2.6) ist auch im Bereich $d < 0,25$ (Fall 1) unterschiedlich, und bei Schaltfolge ‘abcd’ etwas höher. Im Bereich $0,25 < d < 0,5$ (Fall 2) ist er bei ‘abcd’ in allen Transformatoren konstant auf mittlerem Niveau, und bei ‘acbd’ in T_{ab} und T_{cd} deutlich höher als in T_{bc} und T_{da} .

Für alle weiteren Überlegungen, Simulationen und den Aufbau wurde die Schaltfolge ‘abcd’ gewählt, weil diese zur gleichen Auslastung aller vier Drosseln führt. Das hat die Vorteile, dass beim Aufbau nur ein Typ von Saugdrossel verwendet werden muss, und dass die Verluste und dadurch die Erwärmung aller Drosseln gleich sind.

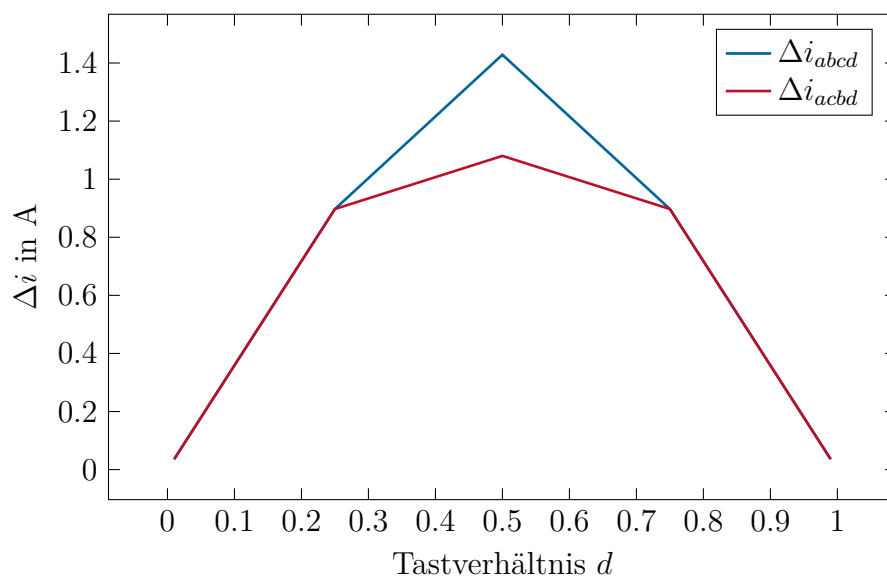


Abbildung 2.5: Zwischenzweig Rippelstrom (Peak-Peak) bei Schaltfolge 'abcd' und 'acbd'. Es handelt sich nur um den Rippelstrom zwischen den Zweigen, der Anteil des Ausgangsstromes ($i_2/4$) ist nicht enthalten.

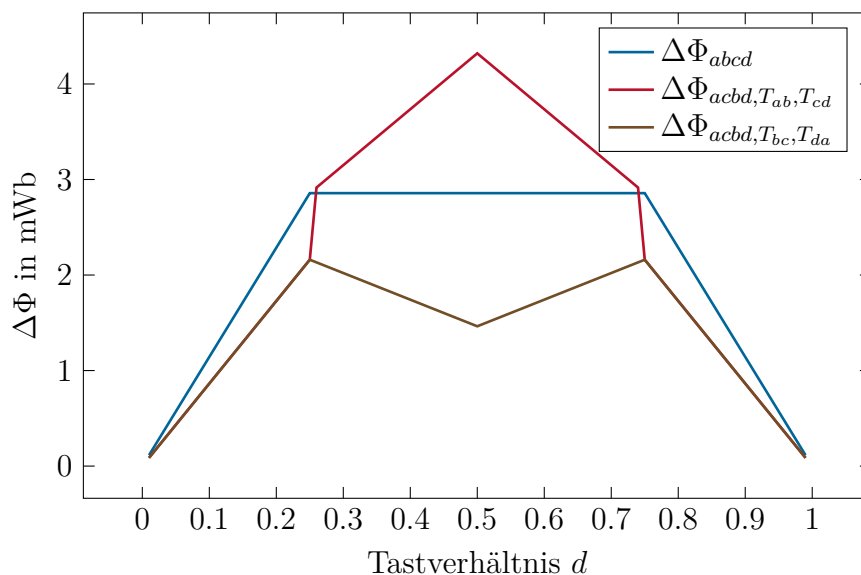


Abbildung 2.6: Maximaler Fluss in den Transformator-kernen (Peak-Peak) bei Schaltfolge 'abcd' und 'acbd'.

2.2 Nutzung der Streuinduktivitäten als Filterinduktivität

Wie in 2 ausgeführt, werden die Hauptinduktivitäten der Transformatoren im Idealfall immer von gleichen Strömen in unterschiedlicher Richtung durchflossen, wodurch sich die erzeugten Durchflutungen im Ferritkern aufheben, und die Hauptinduktivität am Ausgang nicht wirksam ist. Die Streuinduktivität kann aber zu $L = 2L_\sigma/4 = L_\sigma/2$ zusammengefasst, und mit dem Kondensator C als Ausgangsfilter verwendet werden.

Die Feldlinien der Streuinduktivität laufen hauptsächlich über Luft und nur zu einem kleinen Teil über den Ferritkern. Das hat den Nachteil, dass die Induktivität des Ausgangsfilters relativ klein ist, aber den Vorteil, dass der Sättigungsstrom sehr hoch ist, was zu einer hohen Überstromfestigkeit des Ausgangsfilters führt.

Es wird versucht, die Streuinduktivität wie in [6] beschrieben zu maximieren, indem die Spulen im Wickelraum räumlich möglichst weit voneinander getrennt angeordnet werden.

2.3 Symmetrie der Zweigströme

Durch die Wahl der Schaltreihenfolge ‘abcd’ werden die Zweige rein rechnerisch gleichmäßig ausgelastet, und die Durchflutungen, die der Ausgangsstrom in den Saugdrosseln bewirkt, heben sich auf.

Am realen Aufbau wird sich der Ausgangsstrom aber aufgrund von Bauteiltoleranzen nicht gleichmäßig aufteilen. Beispielsweise führen unterschiedliche IGBT-Flussspannungen und Ein- oder Ausschaltzeiten zu unterschiedlichen Spannungs-Zeit-Flächen an den Transformatorspulen, was zur unvollständigen Auslöschung der Durchflutungen im Kern führt. Auch wenn es nur sehr kleine Toleranzen gibt, bewirken diesen durch die Aufsummierung über viele Perioden eine ungleiche Stromverteilung, die zur Sättigung von einzelnen Transformatoren führen kann.

Es wird daher die Möglichkeit zur Regelung der Zweigströme mit PI-Reglern vorgesehen. Die Zweigströme werden am Zusammenführungspunkt (siehe Strompfeile $i_a \dots i_d$ in Abbildung 1.2) gemessen. Weil dieser Punkt am Ausgangskondensator liegt ist die Spannung gegen Masse wesentlich konstanter als direkt an den Halbbrücken, was zu kleineren Common-Mode Fehlern führt.

Die Zweigströme werden in Hardware zum Gesamtstrom i_2 addiert. Dieser wird für den überlagerten Gesamtstromregler verwendet, der ein Basis-Tastverhältnis für alle vier Halbbrücken berechnet.

Parallel dazu wird aus den Zweigströmen durch Mittelung über viele Perioden der Gleichanteil bestimmt. Vier Zweigstromregler berechnen aus den Abweichungen vom Idealwert ($i_2/4$) Offset-Werte, die zum Basis-Tastverhältnis addiert werden. Die PWM-Generatoren werden mit diesem korrigierten Tastverhältnis gespeist. Um den Gesamtstromregler nicht zu beeinflussen muss darauf geachtet werden, dass der Mittelwert der korrigierten Tastverhältnisse dem Basis-Tastverhältnis entspricht.

Weil die Bauteiltoleranzen im allgemeinen sehr klein sind, wird erwartet dass zur Kompensation nur sehr kleine Abweichungen im Tastverhältnis notwendig sind. Es muss daher ein PWM-Generator mit ausreichend hoher Auflösung eingesetzt werden.

3 Realisierung

Dieses Kapitel behandelt die Bauteilauswahl und den Aufbau des Versuchsumrichters, der sich in die Bereiche Halbbrücken, Transformatoren, PWM-Erzeugung, Strom- und Spannungsmessung, und Regelung unterteilt.

Zur Verifikation des Konzeptes wird in dieser Arbeit nur ein Strang bestehend aus vier Zweigen aufgebaut, und als Tiefsetzsteller verwendet. Durch Hinzufügen von zwei weiteren, baugleichen Strängen ergäbe sich ein dreiphasiger Sinuswechselrichter mit einer Leistung von $P = 3 \cdot 600 \text{ V} / 2 \cdot 20 \text{ A} = 18 \text{ kW}$.

3.1 Halbbrücken

Die vier Halbbrückenmodule bestehen aus den IGBTs mit zugehörigen Treibern und deren Spannungsversorgung, sowie Zwischenkreiskondensatoren. Im Folgenden wird auf die Dimensionierung, Bauteilauswahl, Kühlung und den physikalischen Aufbau der Halbbrücken eingegangen. Abbildung 3.1 zeigt den wichtigsten Schaltplanausschnitt der Halbbrücken-Platine. Die vollständigen Schaltpläne und zugehörigen Platinenlayouts sind in Anhang A und B zu finden.

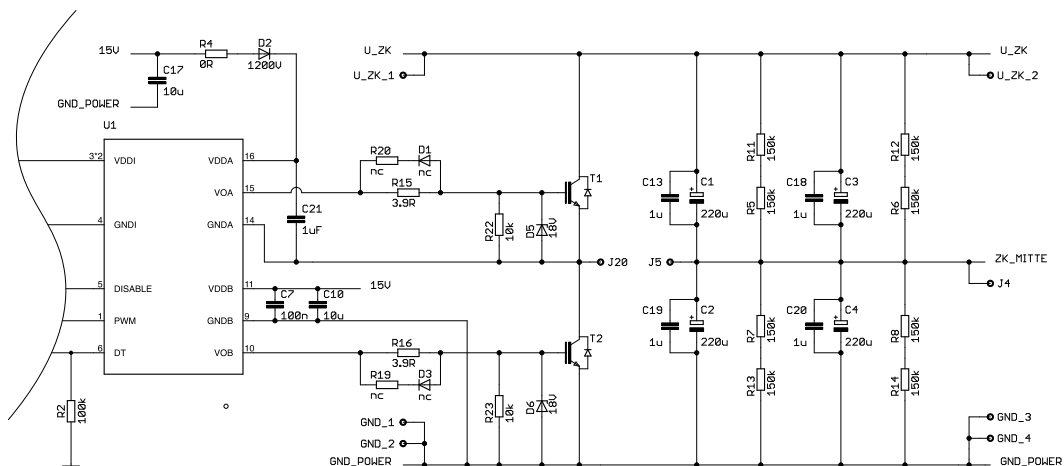


Abbildung 3.1: Ausschnitt aus dem Schaltplan der Halbbrücken-Module.

3.1.1 IGBT

Die IGBTs stellen das Hauptelement der Halbbrücken dar. Bei der Auswahl eines geeigneten Typs müssen die folgenden Kriterien betrachtet werden.

Bei einem Nennstrom von 20 A muss jede der vier Halbbrücken einen Strom von 5 A dauerhaft führen können. Weil die IGBTs nicht am Limit betrieben werden sollen, und der Wechselrichter auch überlastbar sein soll, werden die IGBTs für einen kontinuierlichen Drain-Strom von mindestens 10 A bei einer Temperatur von 100 °C ausgelegt.

Die Drain-Source Durchbruchspannung $U_{CE,br}$ gibt die Spannungsfestigkeit des Schalters an. Bei einer geforderten Nennspannung von 600 V muss eine deutlich höhere Durchbruchspannung von mindestens 1000 V gewählt werden. Zum einen ist der Ausschaltvorgang wegen der hohen di/dt -Werte und unvermeidbaren Verdrahtungsinduktivitäten mit Schaltüberspannungen verbunden (siehe [7]). Andererseits kann die Zwischenkreisspannung durch Rückspeisevorgänge, wie sie beispielsweise bei Bremsvorgängen im Falle des Einsatzes als Antriebsumrichter auftreten können, höhere Werte als die Nennspannung annehmen.

Außerdem sollte die Gate-Ladung möglichst klein sein, um hohe Schaltgeschwindigkeiten erreichen zu können. Um die Leitverluste zu minimieren, ist wiederum eine geringe Kollektor-Emitter-Sättigungsspannung vorteilhaft.

Unter Berücksichtigung dieser Anforderungen wurde der gut erhältliche und günstige Typ 'IKW15N120H3' von Infineon gewählt, der für geringe Schaltverluste bei hohen Schaltfrequenzen ausgelegt und ist im Standardgehäuse TO-247 mit bereits integrierter Freilaufdiode verfügbar ist. Die wichtigsten Daten sind in Tabelle 3.1 zusammengefasst (Datenblatt siehe [8]).

max. collector-emitter voltage	$U_{CE,br}$	1200 V
max. DC collector current (at 100 °C)	I_C	15 A
max. pulsed collector current	$I_{C,puls}$	60 A
max. power dissipation (at 100 °C)	P_{tot}	105 W
max. gate-emitter voltage	V_{GE}	± 20 V
Gate-emitter threshold voltage	$V_{GE,th}$	5,8 V
Collector-emitter saturation voltage (at 125 °C)	$V_{CE,sat}$	2,5 V

Tabelle 3.1: Kenndaten des ausgewählten IGBT Typs. Quelle: [8]

3.1.2 Gatetreiber

Um IGBTs mit Logikpegeln ansteuern zu können, werden Gatetreiber benötigt. Das Gate wird zwar prinzipiell leistungslos geschaltet, allerdings muss bei jedem Schaltvorgang die Gate-Kapazität umgeladen werden, was zu kurzen, aber

hohen Stromimpulsen führt, die der Ausgang eines Mikrocontrollers nicht liefern kann. Außerdem muss der High-Side IGBT (T1 in Abbildung 3.1) mit einer Gatespannung bezogen auf die Ausgangsspannung der Halbbrücke angesteuert werden, was eine Potentialtrennung erfordert.

Als Gatetreiber wird der Typ Si8234BB-D-IS von Silicon Laboratories gewählt [9], der zwei potentialgetrennte Treiber in einem Package vereint, welche mit einem einzigen PWM-Signal angesteuert werden können. So wird sichergestellt, dass auch bei Steuerungsfehlern niemals beide IGBTs gleichzeitig durchschalten. Die Totzeit gibt die Pause zwischen dem Ausschalten von T1 und dem Einschalten von T2 an (und umgekehrt) - in dieser Zeit sind beide IGBTs gesperrt. Sie wird mit dem Widerstand R2 auf 1 μ s eingestellt.

Der Gate-Strom wird aufgrund des maximalen Ausgangsstromes des Treibers von 4 A mit den Widerständen R15 und R16 auf $15 \text{ V} / 3,9 \Omega = 3,85 \text{ A}$ begrenzt. Mit den Widerständen R19 und R20 kann der Ausschaltstrom unabhängig vom Einschaltstrom gewählt werden, sie bleiben aber vorerst unbestückt. Die Zenerdioden D5 und D6 schützen den Treiber vor Spannungsimpulsen, die beim Schaltvorgang am Gate auftreten können.

Wegen der Potentialtrennung benötigt der Ausgangsteil des Treibers eine separate Spannungsversorgung, die von einem potentialgetrennten DC-DC Wandler bereitgestellt wird. Der High-Side Treiber wird über eine Ladungspumpe (bestehend aus D2 und C21) versorgt.

3.1.3 Zwischenkreiskondensatoren

Im Zwischenkreis kommen Elektrolytkondensatoren mit insgesamt 220 μ F pro Halbbrücke zum Einsatz (880 μ F gesamt). Bei einer Serienschaltung von Kondensatoren wird die Spannung im Verhältnis der Kapazitäten und durch Leckströme aufgeteilt, welche allerdings hohen Toleranzen unterliegen. Die Widerstände R5, R6, R7, R8, R11, R12, R13, R14 gewährleisten, dass die Spannung an den einzelnen Kondensatoren deren Spannungsfestigkeit von 450 V nicht überschreitet. Bei Nennspannung wird an diesen Symmetrierwiderständen eine Leistung von $(600 \text{ V})^2 / 300 \text{ k}\Omega = 1,2 \text{ W}$ pro Halbbrücke umgesetzt, was bei kleinen Ausgangsleistungen einen bedeutenden Anteil an der Verlustleistung darstellt. Diese Symmetrierwiderstände sorgen auch für eine Entladung der Kondensatoren nach Abschaltung des Versuchsaufbaues auf unter 60 V in etwa 2,5 Minuten.

3.1.4 Kühlung

Zur Abschätzung des Kühlaufwandes wird der Wirkungsgrad vorerst mit $\eta = 0,95$ angenommen. Daraus ergibt sich bei einem mittleren Tastverhältnis von $\bar{d} = 0,5$

eine Verlustleistung von $P_V = U_N I_N (1 - \eta) \bar{d} = 600 \text{ V} \cdot 20 \text{ A} \cdot 0,05 \cdot 0,5 = 300 \text{ W}$. Diese wird hauptsächlich an den IGBTs (Leit- und Schaltverluste) und an den Transformatoren (Kupfer- und Eisenverluste) auftreten. Die IGBTs im TO247 Gehäuse werden mit Hilfe von elektrisch isolierenden Wärmeleitpads auf einen Aluminiumkühlkörper mit dem Wärmewiderstand von $0,24 \text{ }^\circ\text{C/W}$ geschraubt. Da die Oberfläche zur rein passiven Abgabe der Verlustleistung nicht ausreicht, werden Lüfter montiert, deren Luftstrom sowohl den Kühlkörper, als auch die Transformatoren erreicht.

3.1.5 Platinenlayout

Die Halbbrücken werden modular ausgeführt. Es werden jeweils die beiden IGBTs mit zugehörigen Zwischenkreiskondensatoren, Treibern und deren Stromversorgung (Abbildung 3.1 dargestellt) auf einer Platine verbaut. Die so entstehenden Module sind beidseitig mit Steckerleisten für die Zwischenkreisspannung und die Versorgungsspannung des Treibers ausgestattet, sodass sie in beliebiger Anzahl ohne gesteigerten Verkabelungsaufwand aneinandergereiht werden können. Es werden vier Halbbrücken, d.h. vier Module verwendet, und auf einen gemeinsamen Kühlkörper geschraubt.

Im Layout wurde zur Verringerung von Schaltüberspannungen auf möglichst kurze Verbindungen zwischen IGBTs und Treibern und auf eine niederinduktive Anbindung des Zwischenkreises geachtet. Zur Erhöhung der Strombelastbarkeit und Verringerung der Verluste wurde auf den Platinen die doppelte Kupferdicke ($70 \mu\text{m}$) gewählt.

3.1.6 Messungen

Zur Überprüfung der Halbbrückenmodule wurden diese mit PWM Signalen versorgt, und die Ausgangsspannungen der Halbbrücken (bzw. u_{CE} des unteren IGBT) im unbelasteten Zustand gemessen. Abbildung 3.2 zeigt die entsprechenden Spannungsverläufe. Beide Schaltvorgänge dauern etwa 20 ns und setzen sich aus zwei Bereichen mit unterschiedlichen Anstiegsraten zusammen. In den ersten 6 ns wird die Gatekapazität umgeladen, danach werden die zur Leitfähigkeit erforderlichen bipolaren Ladungsträger in der Driftzone aufgebaut.

Die Schaltüberspannung beträgt bedingt durch das gute Layout nur etwa 15 V beim Einschalt-, und 10 V beim Ausschaltvorgang. Bei Nennspannung treten also maximal 615 V an den Pins der IGBTs auf, die Spannungsfestigkeit der Bauteile ist daher ausreichend dimensioniert.

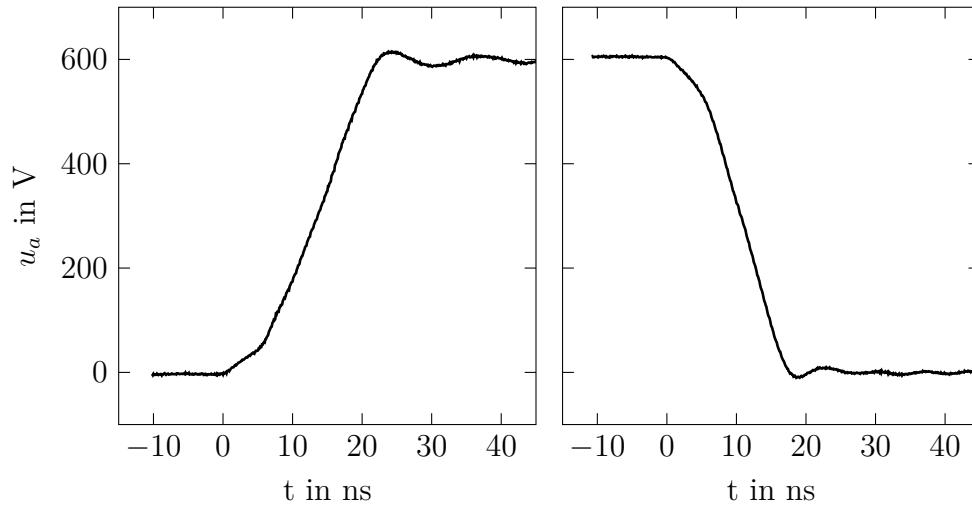


Abbildung 3.2: Steigende- und fallende Flanke des Ausgangssignales einer Halbbrücke, unbelastet bei Nennspannung.

3.2 Transformatoren

3.2.1 Dimensionierung und Simulation

Wie bereits in Kapitel 2.1.3 ausgeführt, wurde die Schaltfolge ‘abcd’ gewählt, weil diese zur gleichen Auslastung aller vier Drosseln führt. Der maximale Fluss (Peak-Peak) in den Transformatoren tritt bei $d = 0,5$ auf und ergibt sich nach Gleichung 2.18 unter Annahme von $L_\sigma \ll L_H$ zu

$$\Delta\Phi_{max} = \frac{U_1 T}{8} \frac{L_H}{L_H + L_\sigma} \approx \frac{U_1 T}{8} = 3 \text{ mVs} \quad (3.1)$$

Dieser Wert dient als Ausgangspunkt für die Dimensionierung. Die wichtigsten Designparameter der Saugdrossel sind:

- Kern-Material
- Kern-Querschnitt
- Kupferquerschnitt
- Windungszahl
- Luftspaltlänge

Die Stromdichte in den Wicklungen sollte 10 A/mm^2 nicht überschreiten, um die Transformatoren auch ohne aktive Kühlung betreiben zu können. Bei einem Nennstrom von 5 A ist daher ein Drahtquerschnitt von mindestens $0,5 \text{ mm}^2$

erforderlich. Es wurde ein Durchmesser von 1 mm gewählt, was einem Querschnitt von etwa 0,8 mm bzw. einer Stromdichte von etwa 6,4 A/mm² entspricht.

Als Kernmaterial kommt aufgrund der hohen effektiven Rippelfrequenz von 100 kHz ein Ferrit zum Einsatz. Im ersten Ansatz wird der gut erhältliche Typ N87 in der Bauform E42/20 gewählt, der eine Wicklungsfläche von 256 mm² hat. Weil die Streuinduktivität als Filterinduktivität genutzt wird und daher so groß wie möglich sein soll, werden die Wicklungen wie in [6] beschrieben möglichst weit voneinander entfernt angeordnet. Bei einem Abstand von 6 mm ergibt sich für eine Wicklung die Fläche 102 mm². Es wird ein Wicklungsfaktor von $w_f = 2$ angenommen, der die nicht-optimale Ausnutzung der Fläche durch den runden Drahtquerschnitt berücksichtigt. Die Anzahl der maximal möglichen Windungen pro Spule ist daher

$$N = \frac{A_w}{w_f A_{CU}} = \frac{102 \text{ mm}^2}{2 \cdot 0,8 \text{ mm}^2} \approx 60 \quad (3.2)$$

Es wurden $N = 40$ Windungen gewählt.

Die Querschnittsfläche eines E42 Kernes ist $A = 234 \text{ mm}^2$. Daher berechnet sich die Flussdichte (Peak-Peak) im E42 Kern bei gegebenem Fluss $\Delta\Phi$ zu

$$\Delta B = \frac{\Delta\Phi}{NA} = \frac{3 \text{ mV}}{40 \cdot 234 \text{ mm}^2} \approx 320 \text{ mT} \quad (3.3)$$

Das entspricht einer Magnetisierung von $\pm 160 \text{ mT}$. Das gewählte N87 Material ist bis etwa $B_{max,N87} = \pm 300 \text{ mT}$ verwendbar (siehe Datenblatt [10]) und ist daher ausreichend dimensioniert.

Im idealen Betrieb teilt sich der Gleichanteil des Ausgangsstromes gleichmäßig auf alle Zweige auf, und die von ihm erzeugten Durchflutungen heben sich in den Transformatorkernen auf. Durch Bauteiltoleranzen und Unsymmetrien im Aufbau werden die Zweigströme aber nicht exakt gleich sein. Um zu verhindern, dass die Drosseln in diesem Fall sofort in Sättigung geraten, wird ein Luftspalt vorgesehen, der aber die Hauptinduktivität verringert.

Die Hauptinduktivität bestimmt den Rippelstrom Δi_{ZZ} , der gemäß Gleichung 2.17 zwischen den Zweigen auftritt und ebenfalls bei $d = 0,5$ maximal ist:

$$\Delta i_{ZZ,max} = \frac{U_1 T}{8} \frac{1}{L_H + L_\sigma} \approx \frac{\Delta\Phi_{max}}{L_H} \quad (3.4)$$

Dieser ist am Ausgang nicht wirksam, tritt aber zwischen den Zweigen auf und trägt somit zu den Kupferverlusten bei. Um diese zu minimieren sollte L_H möglichst groß gewählt werden.

Als Kompromiss zwischen Robustheit gegen Unsymmetrie und geringem Zwischenzweig-Rippelstrom wurde ein Luftspalt von $l = 0,3 \text{ mm}$ gewählt. Dieser erlaubt eine maximale Gleichstromdifferenz zwischen den Zweigen von

$$\Delta I_{max} = \frac{(B_{max,N87} - \frac{\Delta B}{2}) l}{\mu_0 N} = \frac{(300 \text{ mT} - 160 \text{ mT}) \cdot 0,3 \text{ mm}}{4\pi \cdot 10^{-7} \text{ Vs}/(\text{A m}) \cdot 40} \approx 850 \text{ mA} \quad (3.5)$$

Dies entspricht etwa 17 % des Nennstromes. Durch eine FEMM-Simulation mit den hier berechneten Werten wurde die Hauptinduktivität zu $L_H = 1,8 \text{ mH}$ und die Streuinduktivität zu $L_\sigma = 72 \text{ uH}$ ermittelt. Gemäß Gleichung 3.4 ergibt sich dadurch ein Zwischenzweig-Rippelstrom von etwa 1,6 A.

Abbildung 3.3 zeigt das Ergebnis der FEMM-Simulation (Feldlinien und Flussdichtevertelung). Dabei ist erkennbar, dass für die Streuinduktivität ein sehr großer Luftspalt von 8,7 mm (Breite des Transformatorfensters) wirkt.

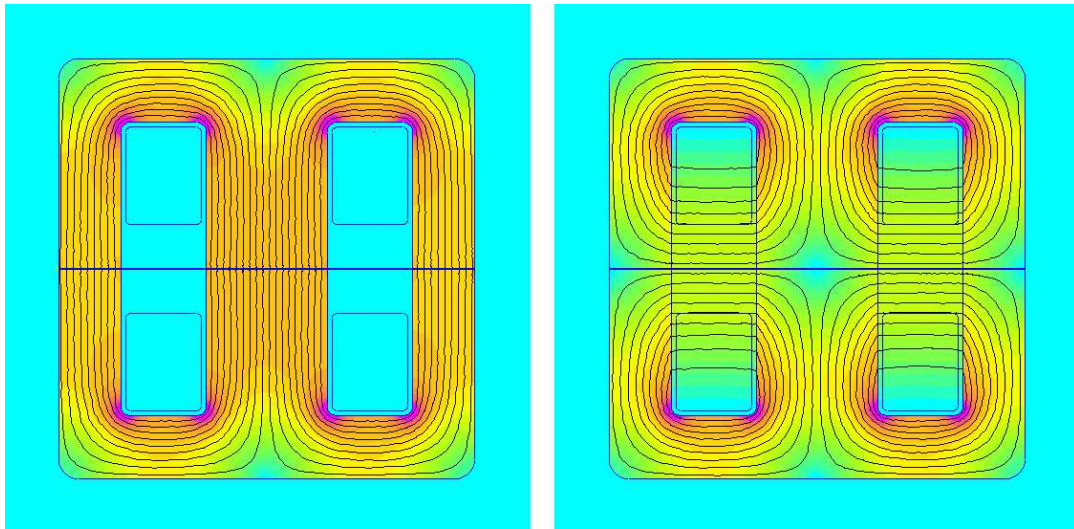


Abbildung 3.3: Simulation der Flussdichteverteilung in den Transformatoren für die Hauptinduktivität (links) und Streuinduktivität (rechts).

3.2.2 Realisierung

Weil die beiden Wicklungen der Saugdrosseln zur Erhöhung der Streuinduktivität räumlich getrennt angebracht, und die Wicklungsfläche möglichst gut ausgenutzt werden soll, wird kein Wickelkörper verwendet. Stattdessen werden die Spulen lose gewickelt, und mit Kaptonband fixiert. Zwischen den Spulen wird ein 6 mm breiter Abstandshalter angebracht, der gleichzeitig zur Montage der Transformatoren auf der Platine dient. Papier-Plättchen mit einer Stärke

von 0,15 mm, die an allen Kontaktstellen zwischen den Ferritkernen eingeklemmt werden, ergeben einen Luftspalt von 0,3 mm. Abbildung 3.4 zeigt das Foto eines Transformators.

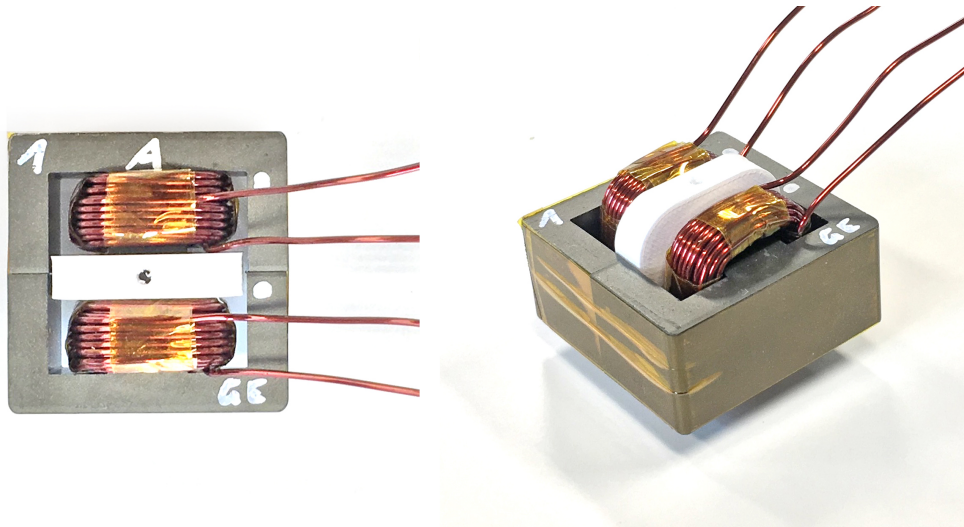


Abbildung 3.4: Fotos des Transformators. Die Wicklungen sind räumlich so weit wie möglich getrennt angeordnet, um die Streuinduktivität zu erhöhen.

Die Haupt- und Streuinduktivität wurde mit einem LCR-Meter in den Schaltungsarten laut Abbildung 3.5 gemessen. Die Ergebnisse sind in Tabelle 3.2 zusammengefasst. Bei Schaltungsart (a) werden beide Spulen vom selben Strom in der selben Richtung durchflossen, die gemessene Induktivität ist daher die vierfache Hauptinduktivität. Im Mittel ergibt sich $L_H = 1,895$ mH bei 100 kHz. Mit Schaltungsart (b) wird die halbe Streuinduktivität gemessen, durchschnittlich ist $L_\sigma = 104,8$ uH. Daraus ergibt sich die Gegeninduktivität M und der Koppelfaktor k zu

$$\begin{aligned} M &= L_H - L_\sigma \approx 1,8 \text{ mH} \\ k &= M/L_H \approx 0,95 \end{aligned} \quad (3.6)$$

Schaltungsart (c) ist die Parallelschaltung aller vier Saugdrosseln, wie sie am Ausgang wirkt, es wird daher wie in 2.2 beschrieben ebenfalls die halbe Streuinduktivität gemessen. Die Messfrequenz hat wenig Einfluss auf die Ergebnisse (61,7 uH bei 100 Hz bis 52,9 uH bei 1 MHz), was darauf hindeutet, dass hauptsächlich Luft als magnetisches Medium wirkt.

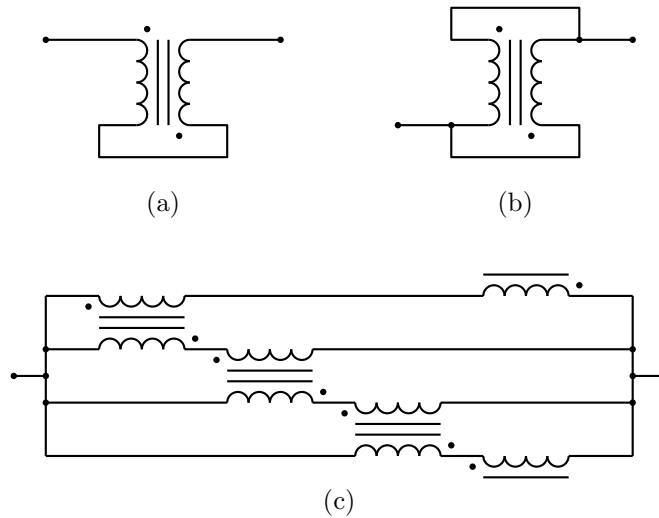


Abbildung 3.5: Schaltungsarten für die Messungen mit dem LCR-Meter und die Stoßstrommessungen. (a) dient zur Messung der Hauptinduktivität, (b) zur Messung der Streuinduktivität, und (c) zur direkten Messung der am Ausgang wirksamen Induktivität.

Schaltungsart	Spule Nr.	f/kHz	R/Ω	$L/\mu\text{H}$	Q	C/nF
(a)	1	100	28,1	7572	168	0,33
	2		27,7	7594	173	0,33
	3		23,3	7570	203	0,33
	4		28,2	7582	170	0,33
(b)	1	100	2,9	51,8	11,25	48,6
	2		3,0	52,1	11,09	48,2
	3		3,1	51,9	10,68	48,3
	4		3,2	53,8	10,76	46,7
(c)		0,1	0,1	61,7	0,6	10 583 000
		1	0,1	61,7	5,7	3 986 800
		10	0,2	61,1	18,3	4133
		100	3,0	52,8	11,2	47,6
		1000	15,0	52,9	22,2	0,5

Tabelle 3.2: Ergebnisse zu den Einzelmessungen der Saugdrosseln mit dem LCR-Meter gemäß den Schaltungsarten aus Abbildung 3.5.

Verwendetes Messgerät: Hioki 3532-50 LCR HiTester

3.2.3 Ermittlung des Sättigungsstromes

Um den Sättigungsstrom der Drosseln zu ermitteln, werden sie in den Schaltungsarten nach Abbildung 3.5 mit einem Spannungssprung beaufschlagt, und der Verlauf des Stromes aufgezeichnet. Abbildung 3.6 zeigt den geladenen Kondensator $C = 2 \cdot 4000 \text{ uF}$ als Spannungsquelle, sowie einen Shunt $R_S = 5 \text{ m}\Omega$ zur Messung des Stromes mit dem Oszilloskop.

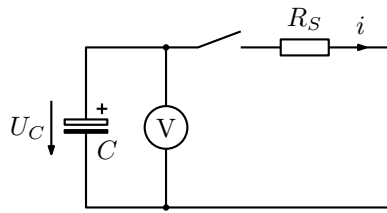


Abbildung 3.6: Versuchsaufbau für die Stoßstrommessungen an den Transformatoren.

Die Messung nach Schaltungsart 1 ergibt den Sättigungsstrom der Hauptinduktivität zu etwa 1 A (Abbildung 3.7). Ab diesem Strom sinkt die Induktivität und der Strom steigt stark an, bis er nur noch durch den Kupferwiderstand der Drossel begrenzt wird. Wird die Drossel zwischen zwei Brücken Zweigen eingesetzt, darf die Differenz der Zweigströme somit maximal 2 A betragen. Aus der Stromanstiegsrate vor der Sättigung kann mit Hilfe der Kondensatorspannung (bei dieser Messung $U_C = 14 \text{ V}$) die Hauptinduktivität berechnet werden, welche mit den zuvor mit dem LCR-Meter ermittelten Werten übereinstimmt:

$$L_H = \frac{1}{4} \frac{U_C}{di/dt} = \frac{1}{4} \frac{14 \text{ V}}{920 \text{ mA}/500 \text{ us}} = 1,9 \text{ mH} \quad (3.7)$$

Abbildung 3.8 zeigt die Stoßstrommessung nach Schaltungsart (b), aus der zur Plausibilitätsprüfung mit Hilfe der Stromanstiegsrate und der Kondensatorspannung von $U_C = 66 \text{ V}$ die Induktivität berechnet wird:

$$L_\sigma = 2 \cdot \frac{U_C}{di/dt} = 2 \cdot \frac{66 \text{ V}}{60 \text{ A}/70 \text{ us}} = 154 \text{ uH} \quad (3.8)$$

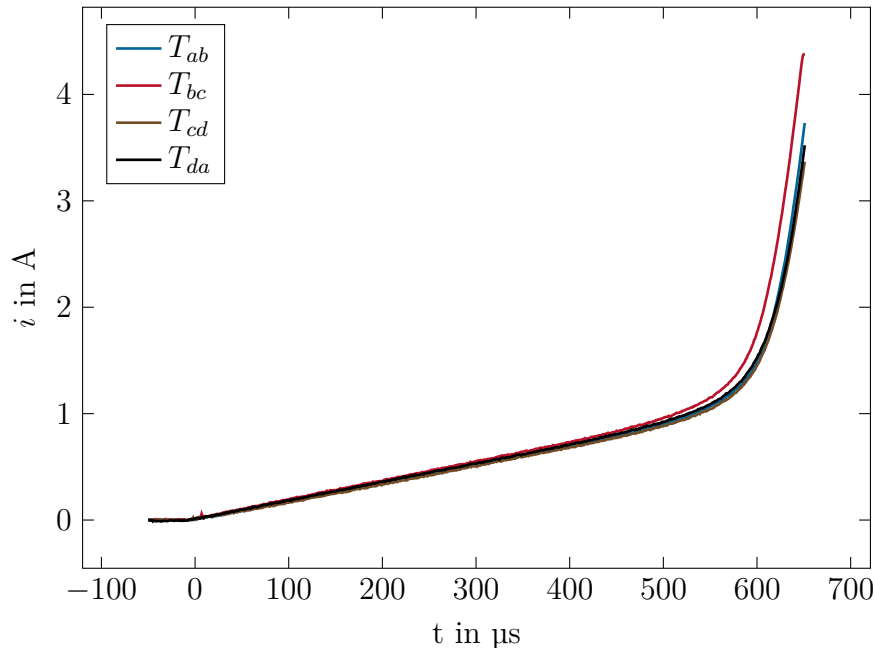


Abbildung 3.7: Stoßstrommessung an der Hauptinduktivität nach Schaltungsart (a).

Diese ist um etwa 50% höher als die mit dem LCR-Meter ermittelte Streuinduktivität, was auf die Entladung des Kondensators während der Messung und die ohm'schen Spannungsabfälle der Zuleitungen zurückzuführen ist, die bei einem Entladestrom in dieser Größenordnung nicht mehr vernachlässigbar sind.

Aus der Lage des Knickes kann der Sättigungsstrom der Streuinduktivität ermittelt werden. Er tritt erst bei etwa 63 A auf, was darauf zurückzuführen ist, dass als magnetisches Medium hauptsächlich Luft wirkt. Weil die Streuinduktivität in Verbindung mit einem Kondensator als Ausgangsfilter verwendet werden soll, ergibt sich dadurch eine sehr hohe Überstromfestigkeit (12-facher Nennstrom). Im Fall von vier Zweigen ist der Sättigungsstrom theoretisch $4 \cdot 63 \text{ A} = 252 \text{ A}$. Zur Überprüfung, ob dieser in der Praxis auch erreicht werden kann, wurde die Stoßstrommessung auch nach Schaltungsart (c) durchgeführt (Abbildung 3.9). Die Messung zeigt, dass die Parallelschaltung der vier Transformatoren in der Praxis bei 170 A sättigt, was dem achtfachen Nennstrom entspricht.

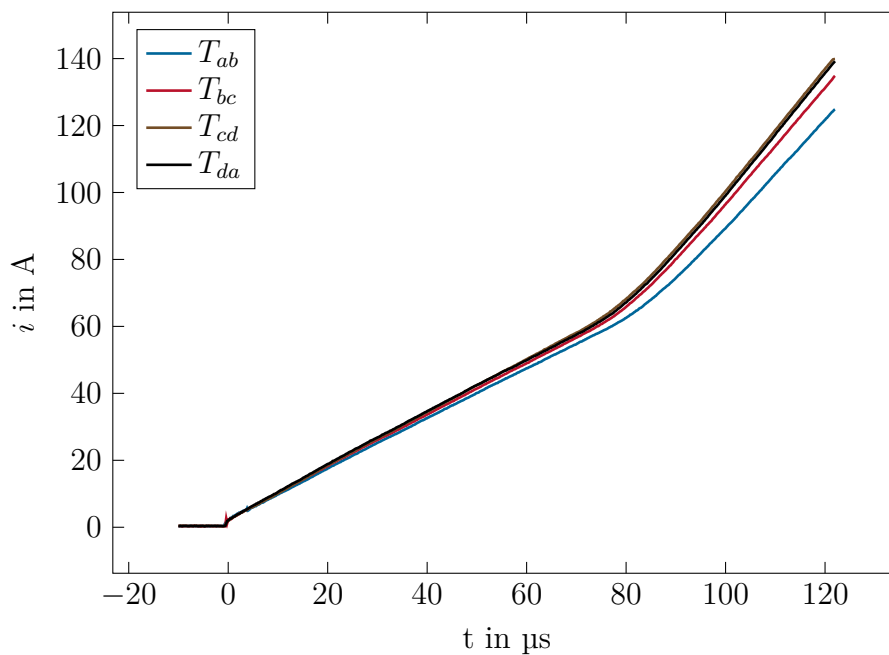


Abbildung 3.8: Stoßstrommessung an der Streuinduktivität nach Schaltungsart (b).

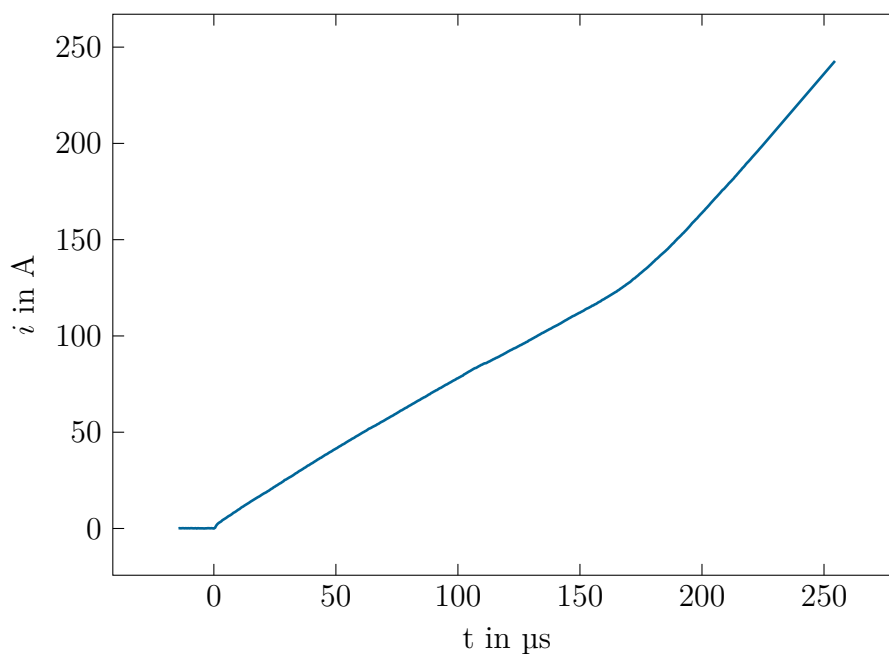


Abbildung 3.9: Stoßstrommessung an der Parallelschaltung aller vier Drosseln nach Schaltungsart (c).

3.3 PWM-Erzeugung

Zur Erzeugung der PWM-Signale für die Halbbrücken wird ein Digitaler Signalprozessor (DSP) vom Typ Texas Instruments TMS320F28069 verwendet. Die wichtigsten Daten sind hier zusammengefasst, das Datenblatt ist unter [11] zu finden.

- CPU Frequenz 90 MHz, 32 Bit
- 8 High Resolution PWM Ausgänge (HRPWM)
- 16 Analog-Digital Converter Kanäle, 12 Bit, 3,46 MSPS
- Floating Point Unit
- Direct Memory Access (DMA)

Von Texas Instruments ist auch ein mit diesem DSP bestücktes Development Board mit integriertem Debugger erhältlich (LAUNCHXL-F28069M, Datenblatt siehe [12]), das für den Aufbau verwendet wurde.

Bei der Ansteuerung der Halbbrücken muss darauf geachtet werden, dass die Phasenverschiebung der Ausgangssignale stets $T/4$ ist. Die Sollwerte der Zähler für die Erzeugung der PWM werden aber immer nur beim Zählerstand 0 aktualisiert. Bei einer Änderung dauert es daher $1/f_S = 40 \mu\text{s}$ bis alle vier Halbbrücken mit dem neuen Tastverhältnis angesteuert werden. Weil auch während dieser Zeit gewährleistet sein soll, dass die Phasenverschiebung genau $T/4$ ist, kommt eine symmetrische PWM zu Einsatz.

Mit 90 MHz CPU Frequenz und 25 kHz PWM-Frequenz sind 3600 Zählerschritte während einer Periode möglich, bei symmetrischer PWM daher nur 1800 Tastverhältnis-Schritte (≈ 11 Bit). Bei einer Nennspannung von 600 V entspricht ein solcher Schritt 333 mV.

Wie in 2.3 beschrieben sollen Abweichungen der Zweigströme durch geringfügige Korrektur der PWM-Tastverhältnisse ausgeregelt werden. Die Abweichungen, beispielsweise in der Vorwärtsspannung der IGBTs liegen im Bereich von wenigen Millivolt, und können daher mit einer Auflösung von 333 mV pro Tastverhältnis-Schritt nicht sinnvoll ausgeregelt werden. Mit Hilfe der High Resolution PWM Module [13] kann diese Auflösung aber erhöht werden. Dabei wird das mit dem gewöhnlichen PWM-Generator erzeugte Tastverhältnis mit Hilfe von sehr schnellen Verzögerungsgliedern um ein Vielfaches von 150 ps verlängert - innerhalb eines Zählerschrittes haben etwa 74 Verzögerungs-Schritte Platz (temperaturabhängig). Die Auflösung erhöht sich dadurch auf etwa $1800 \cdot 74 = 133\,200$ Schritte (≈ 17 Bit), wobei ein Schritt bei Nennspannung etwa 4,5 mV entspricht.

3.4 Strom- und Spannungsmessung

Zur Implementierung des Zweigstromreglers müssen die Zweigströme, bzw. deren Gleichanteil gemessen werden. Wegen der kleineren Offset-Fehler und geringen Kosten werden Shunts anstelle von magnetischen Messwandlern verwendet. Wenn diese direkt am Zusammenführungspunkt (siehe Strompfeile in Abbildung 1.2) angebracht werden, liegen sie an einem Ende am selben Potential, was die Messung vereinfacht. Dieses Potential ist außerdem deutlich 'ruhiger' als direkt an den Halbbrückenausgängen, weil es sich am Ausgangskondensator und damit schon hinter dem Ausgangsfilter befindet.

Am Laboraufbau werden die Spannungen an den Shunts potentialgetrennt gemessen. Dazu wird ein für Strommessungen optimierter Trennverstärker von Texas Instruments, Typ AMC1302 [14] verwendet. Er hat einen Eingangsspannungsbereich von ± 50 mV, eine fest eingestellte Verstärkung von 41, und eine Bandbreite von 280 kHz.

Abbildung 3.10 zeigt die Schaltung zur Messung der Zweigströme mit dem Analog-Digital-Wandler des DSP, der einen Eingangsspannungsbereich von $0 \text{ V} \dots 3,3 \text{ V}$ hat. Mit einem $3 \text{ m}\Omega$ Shunt können Ströme bis etwa $\pm 16 \text{ A}$ (dreifacher Nennstrom) gemessen werden. Der Trennverstärker erzeugt daraus eine Spannung von $\pm 2,5 \text{ V}$ zwischen den Pins $I1P$ und $I1N$. Um diese Spannungsdifferenz in Bezug auf GND messen zu können, ist ein weiterer Differenzverstärker notwendig. Dieser übernimmt drei Aufgaben: Er skaliert die Ausgangsspannung des Trennverstärkers mithilfe des Spannungsteilers $R59$ - $R61$ bzw. $R60$ - $R62$, addiert eine Referenzspannung von $1,65 \text{ V}$, um beide Stromrichtungen erfassen zu können, und wirkt mithilfe von $C67$ und $R62$ als aktives Tiefpassfilter. Weil nur der Gleichanteil der Zweigströme relevant ist, wurde die Grenzfrequenz des Filters mit $2,5 \text{ kHz}$ eine Dekade kleiner als die PWM Schaltfrequenz gewählt.

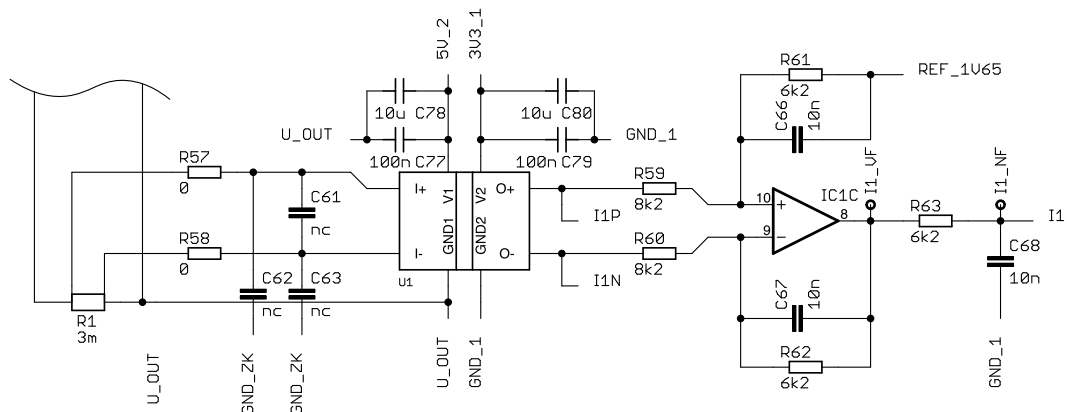


Abbildung 3.10: Schaltungsteil zur Messung des Zweigstromes i_a .

Der Gesamtstromregler ist deutlich schneller als die Zweigstromregler, der Ausgangsstrom darf daher nicht so stark gefiltert werden. Die Analog-Digital-Wandler haben eine Abtastzeit von etwa $1\ \mu\text{s}$ pro Kanal. Weil neben den vier Zweigströmen und dem Ausgangsstrom auch die Ausgangsspannung und die Zwischenkreisspannung gemessen wird, wird als Abtastfrequenz $100\ \text{kHz}$ gewählt. Zur Messung des Ausgangsstromes werden die Ausgangssignale aller vier Trennverstärker vor den Filtern abgegriffen, mit einem zusätzlichen Operationsverstärker addiert (Abbildung 3.11) und mit einem schnelleren Filter, bestehend aus C13 und R21, gefiltert. Die Grenzfrequenz ist in diesem Fall etwa $50\ \text{kHz}$, um als Anti-Aliasing-Filter zu wirken.

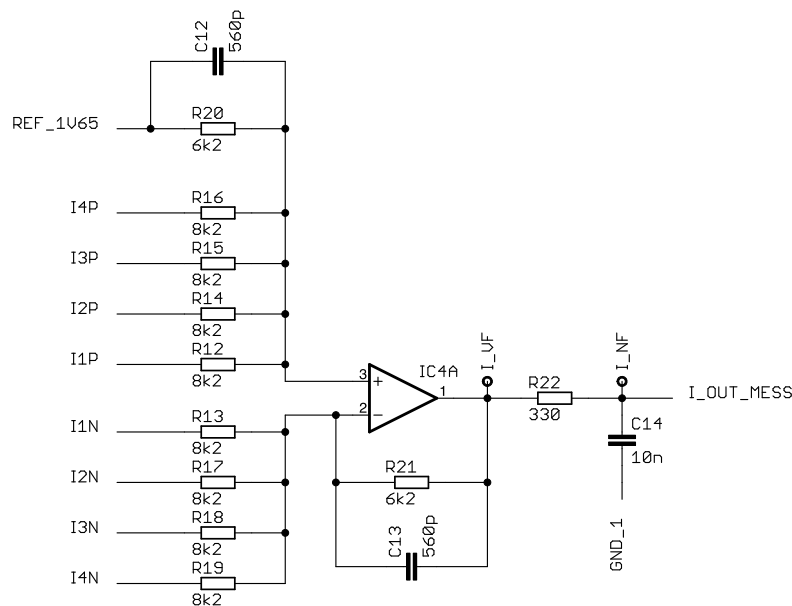


Abbildung 3.11: Schaltungsteil zur Addition der vier Zweigströme zum Gesamtstrom i_2 .

Die Ausgangsspannung und die Zwischenkreisspannung werden mit einem Spannungsteiler und einem Trennverstärker vom Typ AMC1311 [15] gemessen. Abbildung 3.12 zeigt die verwendete Schaltung. Wie bei der Strommessung kommt nach dem Trennverstärker ein Differenzverstärker und ein Anti-Aliasing-Filter mit $50\ \text{kHz}$ Grenzfrequenz zum Einsatz.

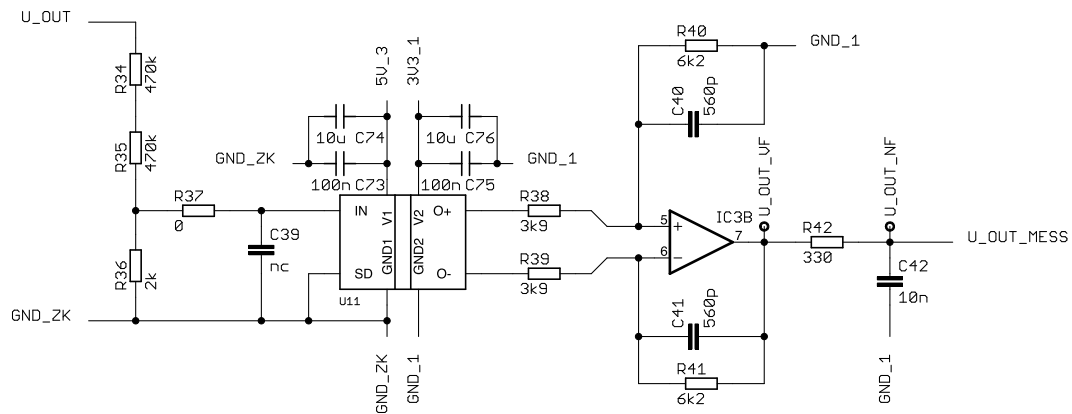


Abbildung 3.12: Schaltungsteil zur Messung der Ausgangsspannung u_2 .

3.5 Regelung

Um den Versuchsaufbau als DC-DC Konverter betreiben zu können wird neben dem Zweigstromregler, der für eine gleichmäßige Aufteilung des Ausgangsstromes auf die Zweige sorgt, auch eine Strom- und Spannungsregelung implementiert. Abbildung 3.13 zeigt die eingesetzte kaskadierte Regelstruktur, die im Folgenden von innen nach außen behandelt wird.

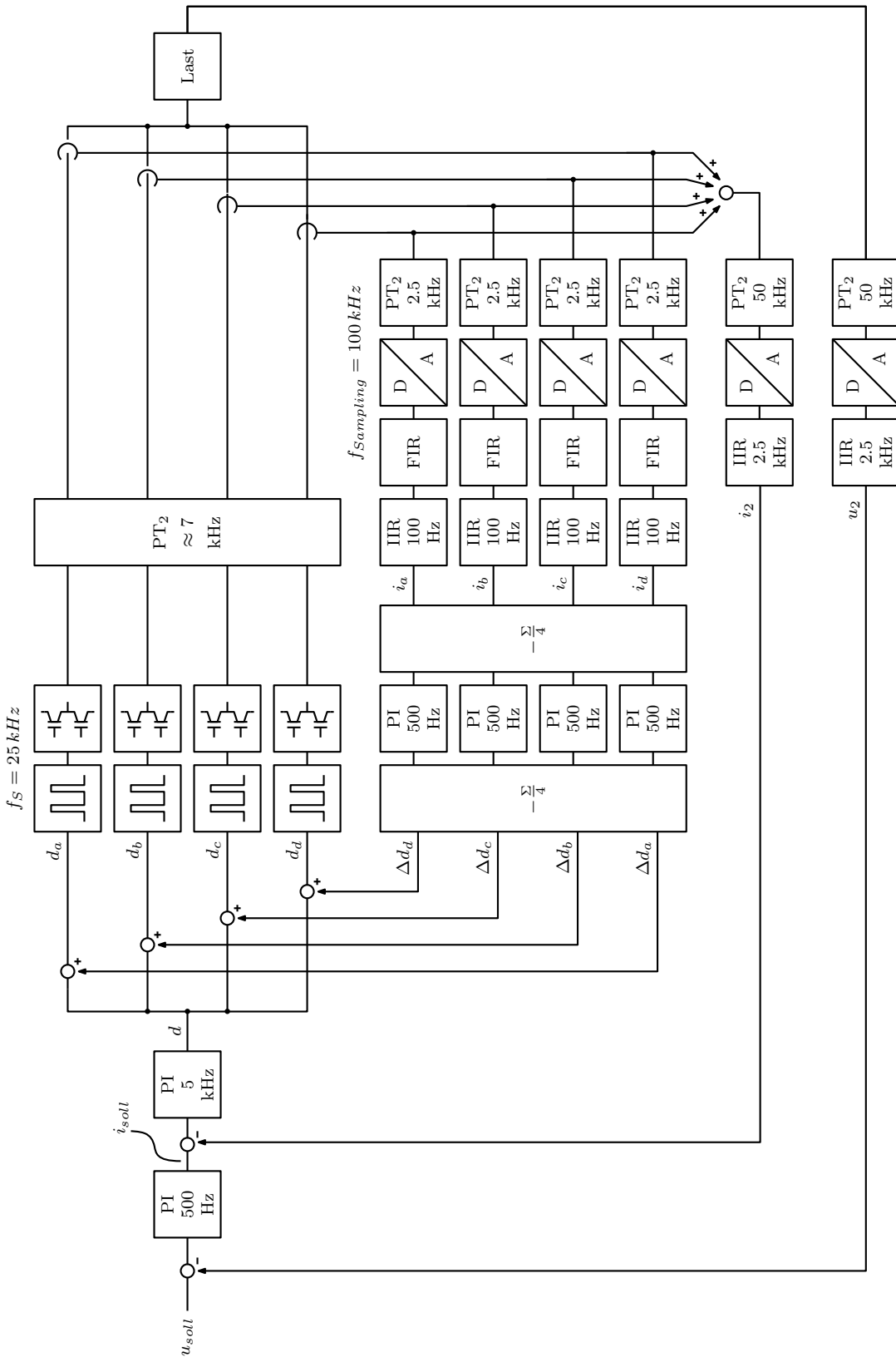


Abbildung 3.13: Schema von Zweigstrom-, Gesamtstrom- und Spannungsregelung.

Der Gesamtstromregler arbeitet mit einer Zykluszeit von 200 μs (5 kHz) am schnellsten, Spannungs- und Zweigstromregler sind um den Faktor 10 langsamer. Der IST-Wert wird wie in Kapitel 3.4 beschrieben durch Addition der Zweigströme gebildet, die am Zusammenführungspunkt mit einer Abtastfrequenz von $f_{\text{Sampling}} = 100 \text{ kHz}$ gemessen werden. Nach dem Anti-Aliasing Filter und der Analog/Digital Wandlung wird er mit einem digitalen IIR-Filter mit einer Grenzfrequenz von 25 kHz gefiltert. Der Regler gibt ein Basis-Tastverhältnis d aus, zu dem die vom Zweigstromregler berechneten Korrekturwerte Δd_x addiert werden ($x \in a \dots d$). Mit diesen Summen $d_x = d + \Delta d_x$ werden die vier PWM-Generatoren und Halbbrücken gespeist.

Der darauf folgende Tiefpass 2. Ordnung mit einer Grenzfrequenz von etwa 7 kHz entsteht durch das Transformatornetzwerk (wirksam als Induktivität $L_\sigma/2 \approx 50 \mu\text{H}$) in Verbindung mit dem Ausgangskondensator $C_2 = 10 \mu\text{F}$.

Für die Zweigstromregler wird durch starke Filterung in Hardware (Grenzfrequenz 2,5 kHz) und Software mit FIR (Fenstergröße 4 Samples, bzw. eine Periode), und IIR Filtern (Grenzfrequenz 100 Hz) der Gleichanteil der Zweigströme bestimmt. Die Abweichung vom arithmetischen Mittelwert des Ausgangsstromes dient als Eingangsgröße für die PI-Regler, der die Korrekturfaktoren berechnet, die zum Basis-Tastverhältnis d addiert werden. Vor der Addition werden die Korrekturwerte um ihren Gleichanteil bereinigt, sodass der Mittelwert der Tastverhältnisse d_x gleich dem Basis-Tastverhältnis d bleibt, und der Gesamtstromregler vom Zweigstromregler nicht beeinflusst wird.

Der äußerste Regelkreis dient der Spannungsregelung. Der IST-Wert wird wie in Kapitel 3.4 beschrieben gemessen, und über ein Filter mit 2,5 kHz einem PI-Regler zugeführt.

Die Integralanteile der PI-Regler wurden mit Anti-Wind-Up Mechanismen ausgestattet, um die Integration zu stoppen, wenn die Stellgröße in Sättigung ist. Im ersten Versuch wurde für den Zweigstromregler ein reiner P-Regler eingesetzt. Allerdings konnte kein Verstärkungsfaktor gefunden werden, der bei jedem Tastverhältnis zufriedenstellende Ergebnisse liefert - abhängig vom Tastverhältnis war der Regler entweder wirkungslos oder schwingend. Weil auch die Einführung eines vom Tastverhältnis abhängigen Verstärkungsfaktors zu keinem zufriedenstellenden Ergebnis führte, wurde ein Integralanteil hinzugefügt. Dadurch arbeitet der Regler sehr zuverlässig und ohne bleibende Regelabweichung, ist aber bewusst träge ausgeführt, um die Ausgangsstromregelung nicht zu beeinflussen. Ein anderer erwähnenswerter Ansatz ist der in [16] vorgestellte Regler, der allerdings nicht weiter verfolgt wurde, weil er sehr genau auf die Transformatoren abgestimmt sein muss, und ebenfalls reine P-Charakteristik hat.

Alle Reglerparameter wurden empirisch ermittelt. Der vollständige Quelltext der Software mit allen Parametern ist in Anhang D zu finden.

3.6 Ergebnis

Abbildung 3.14 zeigt den Versuchsaufbau, auf dem alle weiteren Messungen durchgeführt werden. Die vier Halbbrückenmodule sind auf einem gemeinsamen Kühlkörper montiert, der von unten aktiv belüftet wird. Die Ausgänge der Halbbrücken sind auf eine weitere Platine geführt, die die Transformatoren, Ausgangsfilterkondensatoren, Shunts, Messwandler und Potentialtrennung für Strom- und Spannungsmessung, und die Spannungsversorgung für alle Schaltungsteile enthält. Die rote Platine links ist das DSP Developmentboard mit integriertem Debugger.

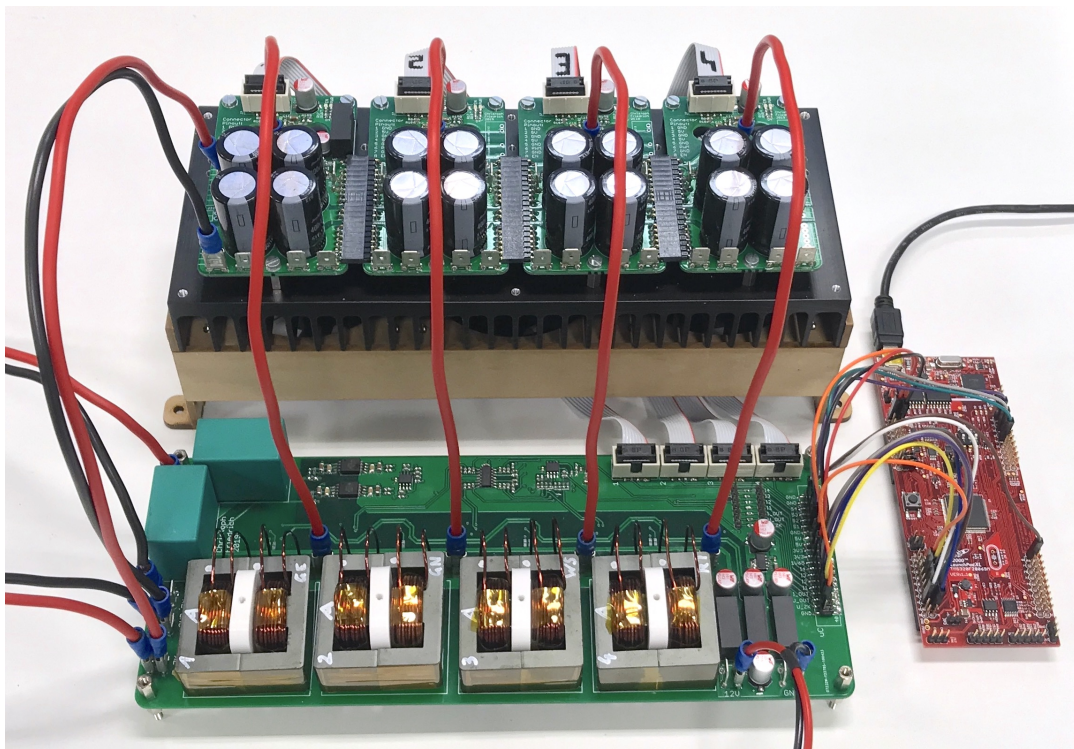


Abbildung 3.14: Foto des Versuchsaufbaues. Im Hintergrund sind die vier Halbbrücken-Module auf einem gemeinsamen Kühlkörper zu sehen. Der Vordergrund zeigt die Platine mit Transformatoren, Ausgangsfilter-Kondensatoren, und Strom- und Spannungsmesswandlern. Die rote Platine rechts ist das DSP Evaluation Board.

4 Messungen

Messungen zu den einzelnen Komponenten (Halbbrücken, Drosseln, etc.) sind im Kapitel 3 zu finden. Um die Funktion des gesamten Wechselrichters zu verifizieren wurden am vollständigen Versuchsaufbau die in diesem Kapitel beschriebenen Messungen zur Stromaufteilung auf die Zweige, Wirkungsgrad und Überlastung durchgeführt.

Der Versuchsaufbau hat eine Nennleistung von 12 kW und wird mit Gleichspannung versorgt. Im Labor stehen dazu zwei Netzteile vom Typ 'Delta Elektronika SM300-10D' mit je 300 V und 10 A zur Verfügung, die in Serie geschaltet werden. Weil damit nur 6 kW Eingangsleistung erreicht werden, wurden einige Versuche nur bei halber Nennleistung durchgeführt. Als Last dient, falls nicht anders angegeben, eine schaltbare Widerstandsdekade.

4.1 Thermische Messungen

Um sicherzustellen, dass die Leistungsbauteile und Kühlkörper thermisch ausreichend dimensioniert wurden, wurde der Versuchsaufbau 30 Minuten bei halber Nennleistung, aber vollem Sekundär-Nennstrom betrieben ($d = 0,5$, $U_1 = 300$ V, $I_2 = 20$ A).

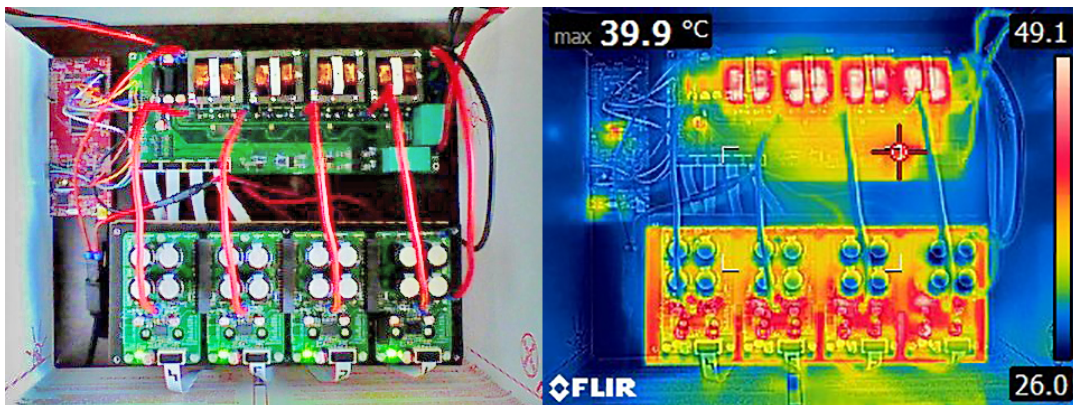


Abbildung 4.1: Foto und Wärmebild des Versuchsaufbaues nach 30 Minuten Betrieb bei halber Nennleistung.

Abbildung 4.1 zeigt ein Thermogramm dieses Betriebszustandes. Bei eingeschalteten Lüftern am IGBT Kühlkörper war die wärmste Komponente der DC-DC Konverter zur Versorgung der IGBT-Treiber mit etwa 40°C . Leistungsbauteile, Kühlkörper und Lüfter sind daher ausreichend dimensioniert.

4.2 Zweigspannungen- und Ströme

Abbildung 4.2 zeigt die Flanken der Ausgangsspannung einer Halbbrücke am fertigen Versuchsaufbau ohne Laststrom sowie bei Nennstrom. Im Vergleich zur Ausgangsspannung an der isolierten Halbbrücke (ohne Transformatornetzwerk, siehe Abbildung 3.2), sind die Schaltzeiten deutlich länger, weil das Transformatornetzwerk als induktive Last wirkt, der Ausgangsstrom der Halbbrücke aber sehr gering ist. Die Totzeit (siehe 3.1.2) ist daher mit $1\ \mu\text{s}$ ausreichend dimensioniert.

Wenn zusätzlich ein Strom i_2 fließt, ist die steigende Flanke, bei der eine Kommutierung von Freilaufdioden auf IGBTs stattfindet, sehr kurz, weil der Sekundärstrom die Freiräumung der Raumladungszone der Dioden begünstigt.

Die fallende Flanke stellt eine Kommutierung von IGBTs auf Freilaufdioden dar, die aufgrund der höheren Trägheit des Einschaltvorganges der Diode (Injektion von Ladungsträgern in die Raumladungszone) generell länger dauert, durch den Sekundärstrom aber auch beschleunigt wird.

Die Schaltüberspannung ist in beiden Fällen geringer als im unbelasteten Fall (Abbildung 3.2), die Spannungsfestigkeit der IGBTs reicht daher aus.

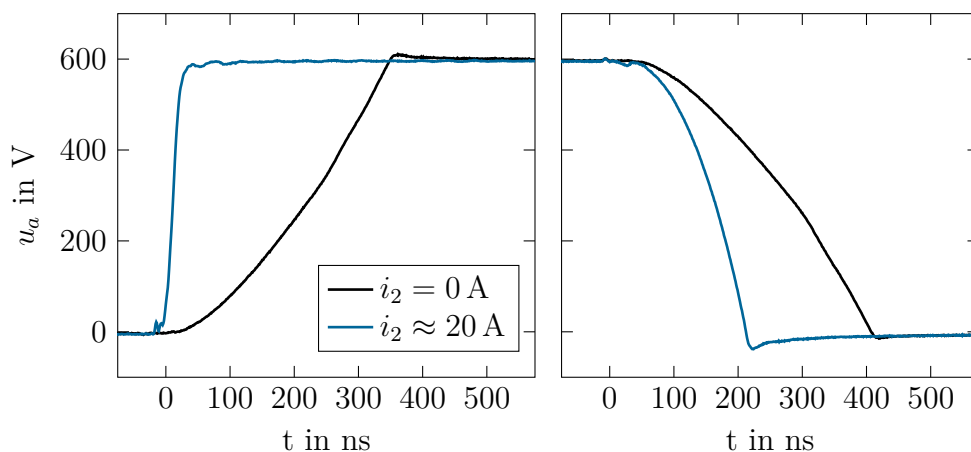


Abbildung 4.2: Steigende- und fallende Flanke des Ausgangssignales einer Halbbrücke bei angeschlossenem Transformatornetzwerk, unbelastet und belastet.

Abbildung 4.3 zeigt Ausgangsstrom und Ausgangsspannung einer Halbbrücke am fertigen Versuchsaufbau ohne Belastung (links) und bei Nennstrom (rechts). Die ‘Sprünge’ im an sich dreieckförmigen Stromverlauf sind auf die Totzeit der IGBT-Treiber zurückzuführen, in der beide Schalter offen sind, und der Ausgang der Halbbrücken daher hochohmig ist.

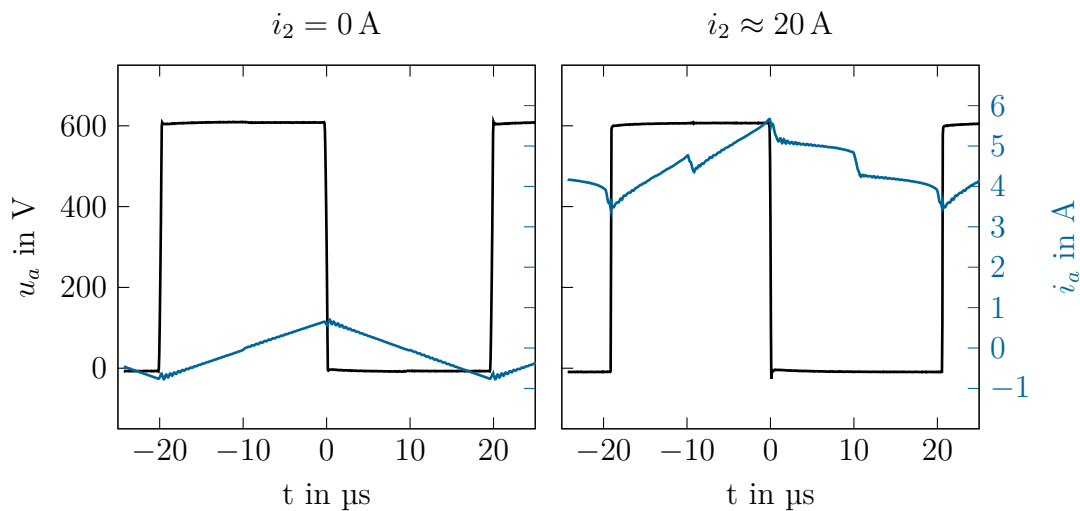


Abbildung 4.3: Ausgangsstrom und -spannung einer Halbbrücke bei angeschlossenem Transformatornetzwerk, unbelastet (links) und belastet mit $2,7\ \Omega$, $i_2 \approx 20\text{ A}$ (rechts).

4.3 Aufteilung des Ausgangsstromes auf die Brückenzeige

Bei der gewählten Schaltungstopologie sind die Leistungsbauteile, insbesondere IGBTs und Transformatoren nur für ein Viertel des Gesamt-Nennstromes ausgelegt. Daher ist es sehr wichtig, dass sich der Ausgangsstrom möglichst gleichmäßig auf die vier Zweige aufteilt. Wie in Kapitel 2 beschrieben haben aber bereits kleine Toleranzen der Ein- und Ausschaltverzögerung sowie der Leitverluste der IGBTs große Auswirkungen auf die Zweigströme.

Zur genauen Untersuchung wurden die Zweigströme bei ausgeschaltetem Zweigstromregler bei verschiedenen Ausgangsströmen mit dem DSP gemessen. Dazu wurde das Tastverhältnis bei Nennspannung und einem Lastwiderstand von $15\ \Omega$ von $d = 0$ bis $d = 0,5$ erhöht. Der Sekundärstrom steigt dabei bis auf den Nennstrom von 20 A . Das Ergebnis ist in Abbildung 4.4 dargestellt.

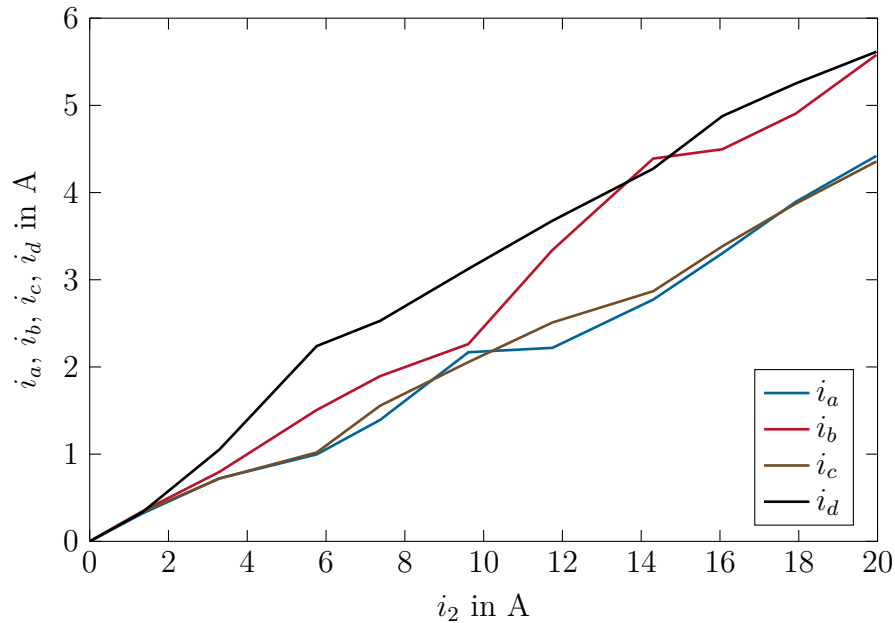


Abbildung 4.4: Zweigströme ohne Regelung abhängig vom Sekundärstrom.

Der Versuch wurde mehrmals direkt hintereinander durchgeführt, die Aufteilung des Gesamtstromes auf die Zweigströme war aber immer stark unterschiedlich. Bei allen Messungen war bis etwa 6 A ersichtlich, dass die Abweichung der Zweigströme vom Idealwert mit steigendem Strom zunimmt. Ab 6 A bleibt aber die Differenz zwischen größtem und kleinstem Zweigstrom auf etwa 2 A begrenzt.

Die Stoßstrommessungen an den einzelnen Transformatoren (siehe Kapitel 3.2.3, insbesondere Abbildung 3.7) zeigen, dass die Hauptinduktivität ab 2 A Eingangsstromdifferenz zu sättigen beginnt. Dadurch sinkt ihre Induktivität, in der gewählten Schaltungstopologie (Abbildung 1.2) werden zwei Halbbrücken über die ohmschen Widerstände zweier Transformatorwicklungen und des übrigen Aufbaues (Kabel, Leiterbahnen) kurzgeschlossen, und der entsprechende Zweigstrom steigt an. Dadurch fällt eine höhere Spannung am ohmschen Widerstand des Zweiges ab, die Spannung an der Hauptinduktivität sinkt wieder, die Spannung-Zeit-Fläche wird verkleinert und der Transformator nähert sich wieder dem ungesättigten Betrieb. So stellen sich also auch ohne zusätzliche Regelung Zweigströme ein, die sich höchstens um den Sättigungsstrom der Hauptinduktivität unterscheiden.

Die Schaltung funktioniert zwar durch die selbstregelnden Eigenschaften auch ohne aktive Regelung der Zweigströme, dies hat aber Nachteile. Wie in Abbildung 4.4 ersichtlich, ist bei 6 A der Strom I_d mehr als doppelt so groß wie I_a . Einerseits entstehen dadurch insgesamt höhere ohmsche Verluste, weil diese quadratisch vom Strom abhängen und daher nur im Fall $I_a = I_b = I_c = I_d$ minimal sind. Andererseits werden die Halbbrücken und Drosseln unterschiedlich stark

belastet, was zu einer ungleichmäßigen Erwärmung führt und eine ausreichende Überdimensionierung notwendig macht.

Nach Aktivierung des Zweigstromreglers wurden die in Abbildung 4.5 dargestellten Messwerte aufgenommen. Es ist erkennbar, dass alle vier Zweigströme exakt gleich sind. Somit werden die vier Transformatoren gleich ausgelastet, und geraten nicht in Sättigung.

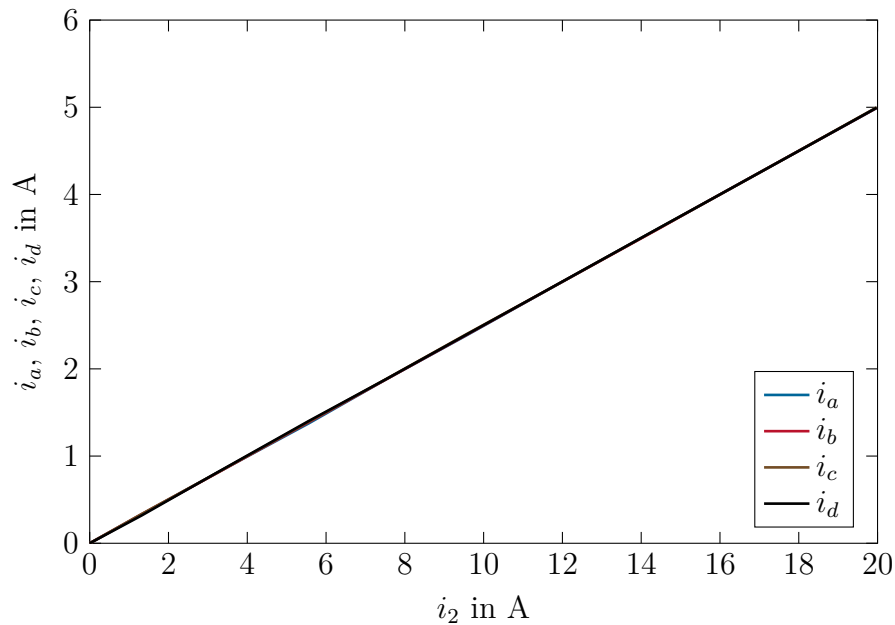


Abbildung 4.5: Zweigströme mit Regelung abhängig vom Sekundärstrom.

4.4 Wirkungsgrad

Desweiteren wurde der Wirkungsgrad des Schaltwandlers mit und ohne Zweigstromregler abhängig vom Sekundärstrom gemessen. Die Primärspannung wurde auf Nennspannung gehalten, das Tastverhältnis konstant auf 0,5. Der Sekundärstrom wurde durch Variation des Lastwiderstandes mit Hilfe einer Widerstandsdekade variiert.

Primär- und Sekundärspannung wurden mit Tischmultimetern vom Typ ‘Agilent U3401A’ gemessen. Zur Messung von Primär- und Sekundärstrom wurden zuvor kalibrierte $1\text{ m}\Omega$ Shunts mit Tischmultimetern vom Typ ‘Agilent U3402A’ verwendet. Um zu verhindern dass die ohmschen Verluste der Zuleitungen die Messung verfälschen wurde besonders darauf geachtet, die Messpunkte möglichst nahe an den Schaltwandler zu legen.

Das Ergebnis ist in Abbildung 4.6 zu sehen. Der Wirkungsgrad von über 95 % ab

einem Sekundärstrom von 1 A, und über 97 % ab 6 A ist sehr zufriedenstellend. Zu beachten ist auch, dass bei dieser Messung die Verluste am Ausgangsfilter (bestehend aus der halben Streuinduktivität der Transformatoren und 10 uF Folienkondensatoren) bereits enthalten sind. Bei Umrichtern mit ungefiltertem PWM-Ausgangssignal treten diese Verluste erst im Motor auf und werden bei der Messung des Umrichter-Wirkungsgrades nicht berücksichtigt.

In der Abbildung ist außerdem erkennbar, dass ab 6 A der Wirkungsgrad mit dem Zweigstromregler etwas höher als ohne Regler ist. Das liegt wie in Kapitel 4.3 beschrieben daran, dass ab 6 A die Unterschiede der Zweigströme so groß werden dass die Drosseln in Sättigung geraten, was Wirkungsgradeinbußen zur Folge hat. Diese sind aber mit 0,15 % relativ gering.

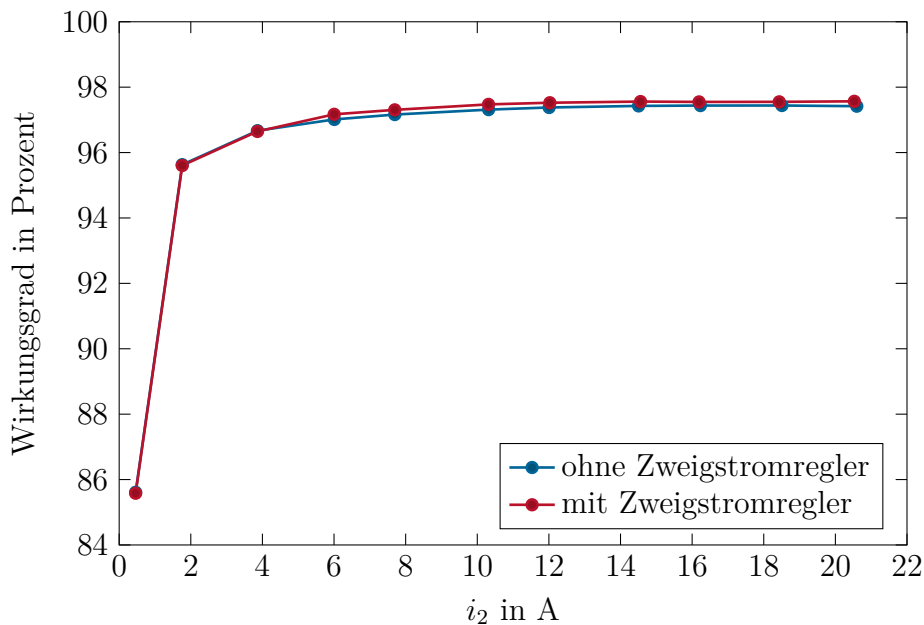


Abbildung 4.6: Wirkungsgrad über Sekundärstrom, gemessen bei $d = 0,5$ und Variation des Sekundärstromes über eine Widerstandsdekade.

4.5 Kurzzeitige Überlastung

Im Falle einer Verwendung dieser Schaltungstopologie als Antriebsumrichter/-Servoumrichter ist die Überlastfähigkeit sehr wichtig. Diese wurde im Folgenden durch eine kurzzeitige Belastung des Schaltreglers mit dem dreifachen Nennstrom (60 A) erprobt.

Für die IGBTs, das Kupfer der Transformatoren und die Leiterbahnen ist eine kurzzeitige Überlastung thermisch kein Problem. Da die Streuinduktivität der

Transformatoren als Ausgangsfilterinduktivität dient, und diese wie eine Spule mit sehr großem Luftspalt wirkt, sollte auch sie nicht in Sättigung geraten. Die Stoßstrommessungen an den Einzeldrosseln (siehe Kapitel 3.2.3) haben ergeben, dass die Parallelschaltung aller vier Saugdrosseln erst bei 170 A sättigt.

Der Schaltregler wurde daher bei Nennspannung und einem Tastverhältnis von $d = 0,15$ ausgehend von einer Grundlast von 6 A für 0,5 s mit 63 A belastet. Als Grundlast wurde die Widerstandsdekade, eingestellt auf $R_1 = 13,2 \Omega$ verwendet. Zu dieser wurde gemäß Abbildung 4.7 mit Hilfe eines IGBT Moduls als Lastschalter eine 100 m Rolle 1,5 mm² Installationsdraht mit einem ohmschen Widerstand von $R_2 \approx 1,15 \Omega$ und geringem induktiven Anteil parallel geschaltet. Der obere IGBT wurde durch Anlegen einer negativen Spannung gesperrt und seine Rückdiode als Freilaufdiode für die Last verwendet. Die Messung des Gesamtstromes erfolgte mit einer Stromzange vom Typ ‘Tektronix TCP305’ und zugehörigem Verstärker ‘TCPA300’. Die Zweigströme wurden mit vier Stromzangen vom Typ ‘Heme PR30’ gemessen. Wegen der starken Störungen, hervorgerufen durch die Schaltflanken der IGBTs mit effektiv 100 kHz wurden alle Messunten über eine Fensterbreite von 4 Messwerten gefiltert.

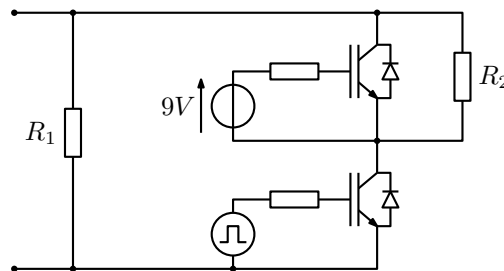
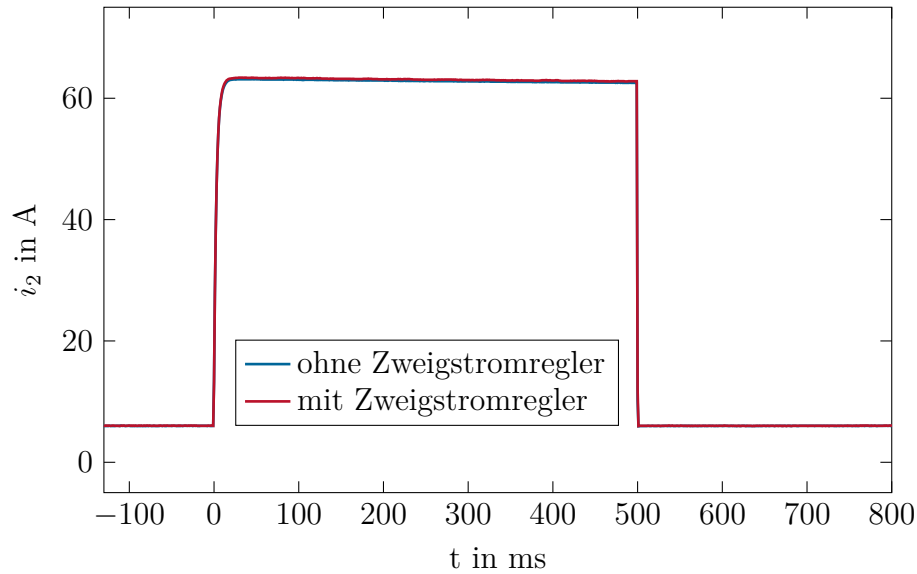


Abbildung 4.7: Schaltbild der Last, die für die Stoßstrommessung verwendet wurde. $R_1 = 13,2 \Omega$ stellt die Grundlast dar, zu der kurzzeitig $R_2 = 1,15 \Omega$ parallel geschaltet wird.

Abbildung 4.8 zeigt den Gesamtstrom an der Last. Für den Schaltregler stellt die kurzzeitige Belastung mit 63 A kein Problem dar. Die Kurven mit und ohne Zweigstromregler unterscheiden sich nicht. Während der Belastung sinkt der Strom leicht, was auf den positiven Temperaturkoeffizienten der Last und der Zuleitungen zurückzuführen ist.

Die einzelnen Zweigströme ohne Regler sind in Abbildung 4.9 zu sehen. Hier sind die Ströme bereits vor dem Laststoß stark unterschiedlich. Bei Belastung wird dieser Unterschied noch größer, aufgrund der selbstregelnden Eigenschaften bleibt er aber limitiert auf den Sättigungsstrom der Hauptinduktivität.

Abbildung 4.8: Stoßstrommessung, Gesamtstrom i_2 .

Mit Zweigstromregler (Abbildung 4.10) sind vor der Belastung alle Zweigströme gleich groß. Bei Belastung sind sie jedoch unterschiedlich, die Abweichungen sind aber geringer als ohne Regler, weil die Halbbrücken noch mit den korrigierten Tastverhältnissen von vor der Belastung arbeiten. Die Korrekturwerte werden während der Belastung nicht mehr verändert, der Regler ist deaktiviert, weil die Strommessung nicht für so hohe Ströme ausgelegt ist. Nach der Belastung sind die Zweigströme wieder geringfügig unterschiedlich, der Regler arbeitet aber wieder und regelt die Abweichung nach etwa 300 ms aus.

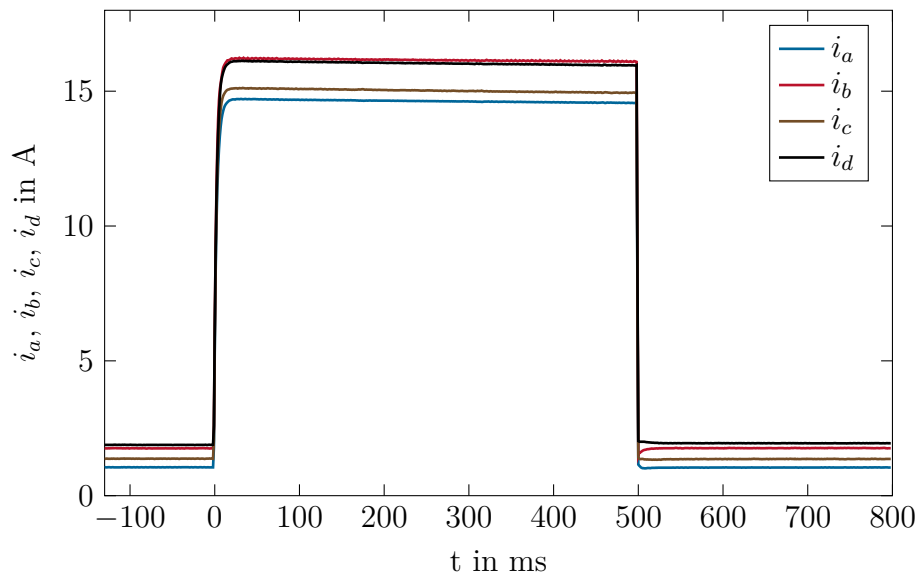


Abbildung 4.9: Stoßstrommessung, Zweigströme ohne Zweigstromregler.

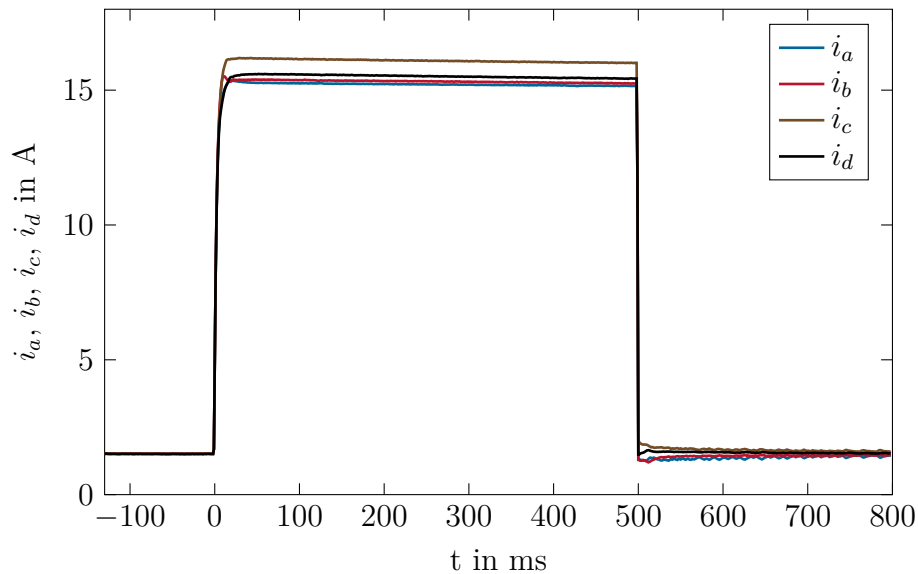


Abbildung 4.10: Stoßstrommessung, Zweigströme mit Zweigstromregler.

4.6 Fazit

Die Messungen zeigen deutlich, dass das beschriebene Konzept in Bezug auf thermische Robustheit, Wirkungsgrad, und Überlastfähigkeit alle Erwartungen erfüllt. Auch die Zweigstromregelung funktioniert zufriedenstellend, und führt zu einer gleichmäßigen thermischen Belastung der Leistungsbauteile und einer geringfügigen Wirkungsgradsteigerung, wäre aber aufgrund der selbstregelnden Eigenschaften nicht zwingend notwendig. Durch den Einsatz von Mosfets oder SiC-Mosfets als Schalter könnte das Verhalten ohne Regelung vermutlich wegen des ohmsch wirkenden Kanalwiderstandes weiter verbessert werden.

5 Zusammenfassung und Ausblick

Im Rahmen dieser Diplomarbeit wurde ein Sinuswechselrichter-Zweig analysiert, dimensioniert und aufgebaut, der eine spezielle Schaltungstopologie nutzt, um vier phasenversetzt getaktete Halbbrücken über ein Netzwerk aus Transformatoren parallel zu schalten.

Dadurch ergibt sich eine Multi-Level Charakteristik, und die Frequenz des Stromrippels am Ausgang ist vier mal so hoch wie die Schaltfrequenz, wodurch auch bei der Verwendung von günstigen IGBTs Frequenzen > 100 kHz am Ausgang erreicht werden können.

Die Streuinduktivität der Transformatoren wurde durch die Anordnung der Wicklungen maximiert und konnte in Verbindung mit einem Kondensator als LC-Ausgangsfilter genutzt werden. Messungen zeigen, dass magnetische Sättigung erst bei 8-fachem thermischen Nennstrom eintritt, und das Filter daher sehr überstromfest ist. Um dies zu demonstrieren, wurde der Wechselrichter erfolgreich mit dreifachem Nennstrom (60 A) belastet.

Bauteiltoleranzen können allerdings dazu führen, dass sich der Ausgangsstrom nicht gleichmäßig auf die Zweige aufteilt. Versuche haben aber gezeigt, dass die Abweichungen auf den Sättigungsstrom der Hauptinduktivität der Transformatoren beschränkt sind. Wenn diese akzeptiert und in der Dimensionierung der Bauteile berücksichtigt werden, ist daher auch ein ungeregelter Betrieb möglich. Eine exakt gleichmäßige Stromaufteilung wurde mit einem Zweigstromregler erreicht, der als PI-Regler ausgeführt ist, und die Tastverhältnisse der Halbbrücken mit einem die Stromdifferenz korrigierenden Offset beaufschlagt. Zu beachten ist, dass die Korrektur der minimalen Bauteiltoleranzen eine sehr hohe PWM-Auflösung erfordert (verwendet wurden 17 Bit). Die Ausgangsstrom und -spannungsregelung wurde ebenfalls als PI-Regler in Form einer kaskadierten Regelung implementiert.

Auch der Wirkungsgrad von über 97% im Laststrombereich 25...100% ist für einen Sinuswechselrichter inklusive Ausgangsfilter, welcher ausschließlich aus Standardkomponenten aufgebaut ist, sehr zufriedenstellend.

Die vorgestellte Schaltungstopologie hat außerdem fertigungstechnische Vorteile gegenüber anderen Arten, Halbbrücken parallelzuschalten. Die Anzahl der Zweige kann beliebig gewählt werden, und alle Transformatoren sind baugleich ausgeführt. Umrichter könnten daher sehr einfach modular aufgebaut, und je

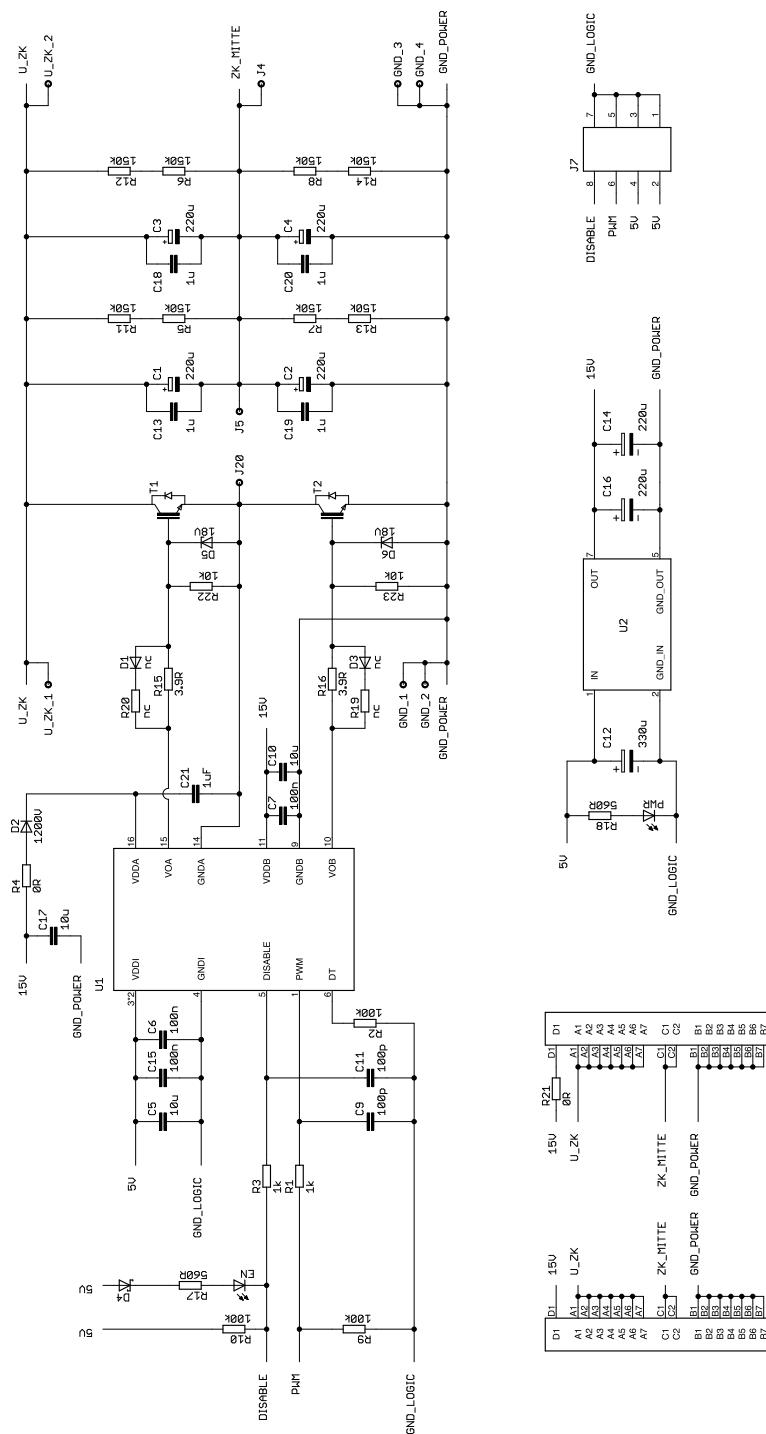
nach benötigter Leistung parallel geschaltet werden.

Als Kerne für die Transformatoren wurden Ferrite genutzt, die auch bei deutlich höheren Frequenzen einsetzbar sind. Die Verwendung von SiC-Mosfets anstatt der IGBTs würde es ermöglichen, beispielsweise bei Schaltfrequenzen von 100 kHz eine effektive Ausgangsfrequenz von 400 kHz (bei 4 Zweigen) zu erreichen, wobei die Transformatoren dank der hohen Frequenzen sehr klein realisierbar wären. Es ist auch zu erwarten, dass der positive Temperaturkoeffizient des Kanalwiderstandes von (SiC-)Mosfets die zuvor beschriebenen selbstregelnden Eigenschaften der Zweigstromaufteilung verbessert.

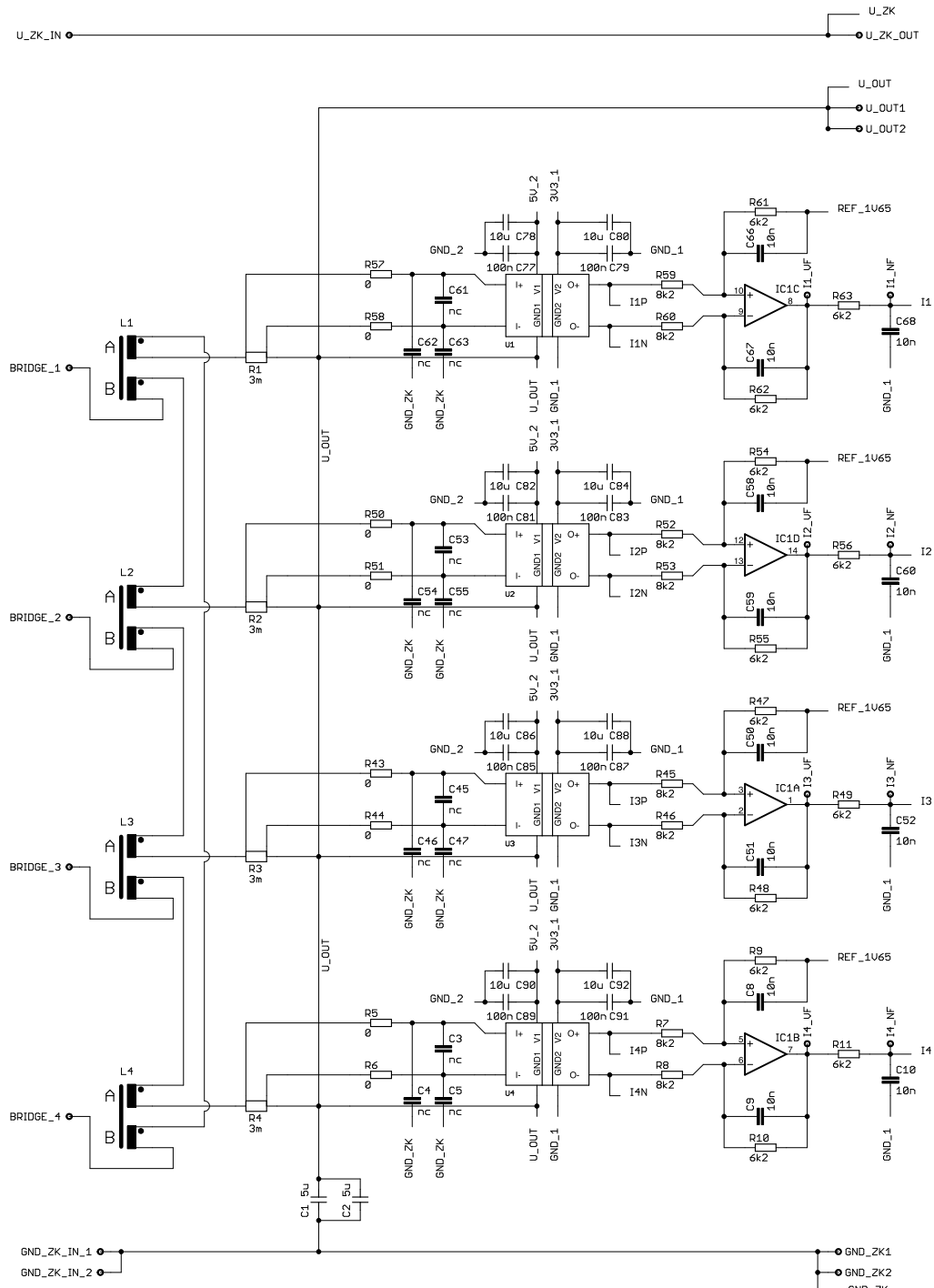
Außerdem könnte das Konzept bei PWM-Umrichtern eingesetzt werden, um die bei SiC-Mosfets oft problematischen steilen Schaltflanken abzuflachen [17]. Dabei wäre die Phasenverschiebung zwischen den N Zweigen nicht T/N sondern sehr klein, und dient gewissermaßen zur Aufteilung einer Schaltflanke in mehrere Stufen, wodurch am Ausgang geringere du/dt -Werte und weniger hochfrequente Harmonische auftreten.

A Schaltpläne

Halbbrücken-Module



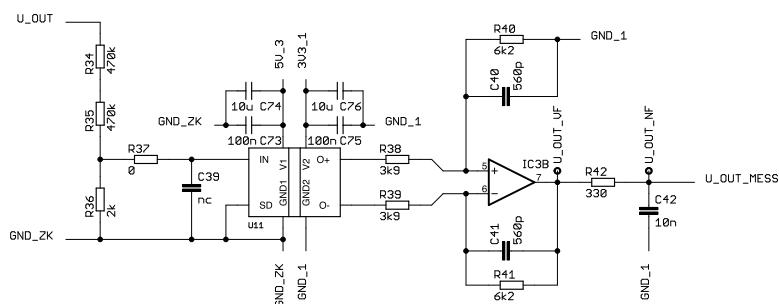
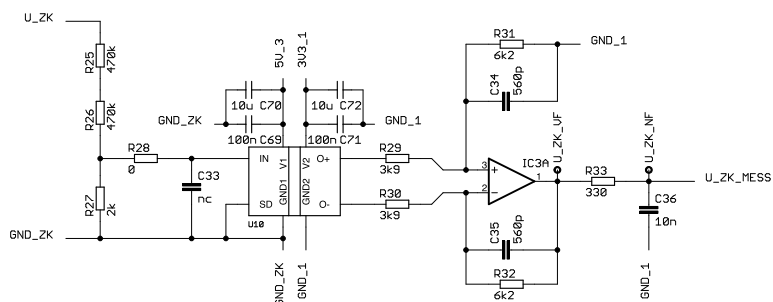
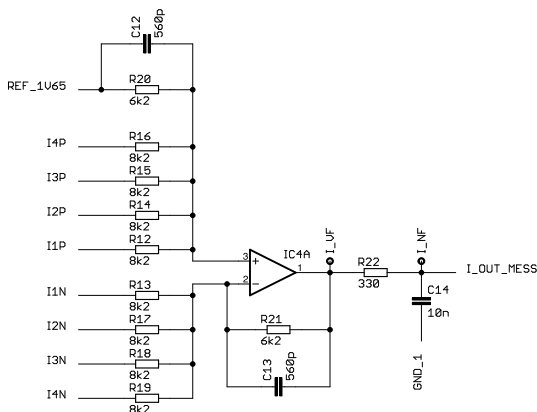
Messplatine Seite 1/3



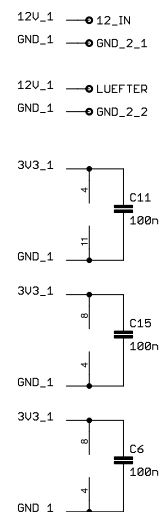
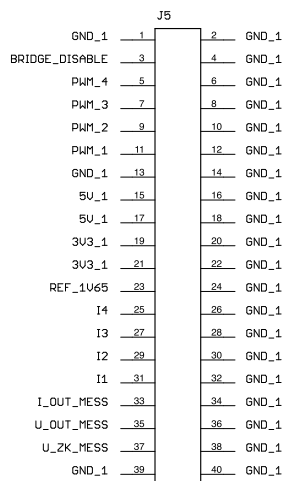
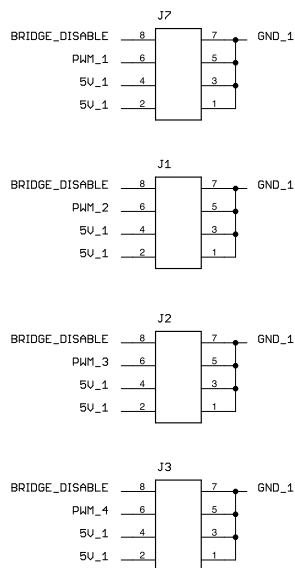
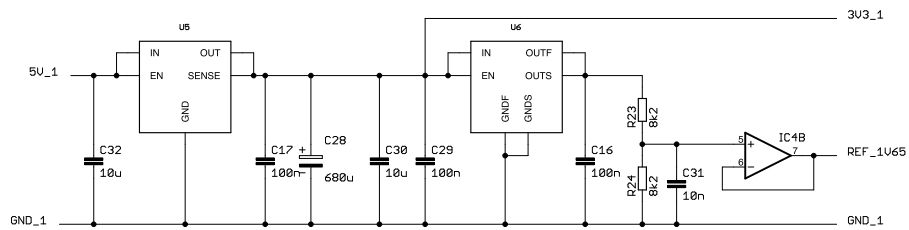
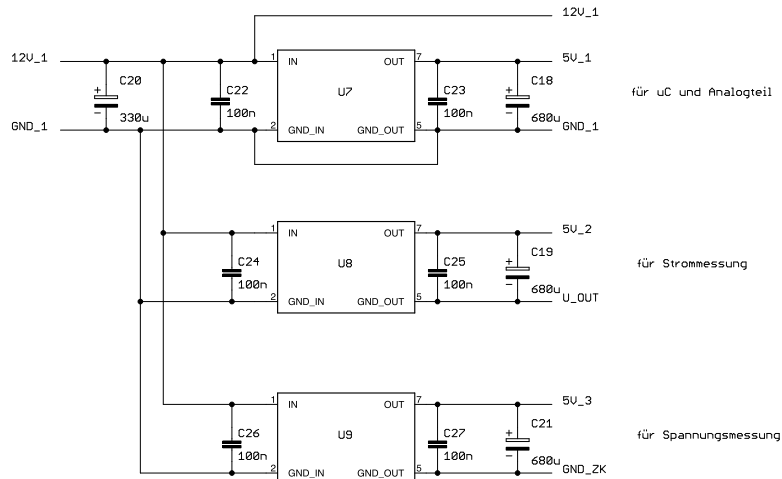
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Messplatine Seite 2/3

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

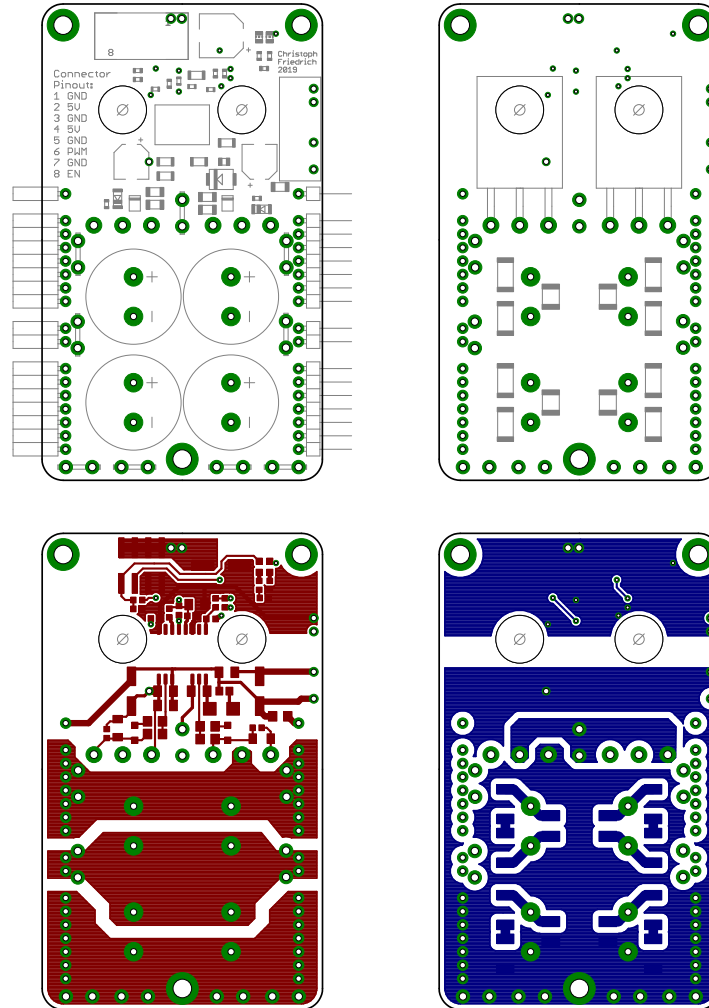


Messplatine Seite 3/3

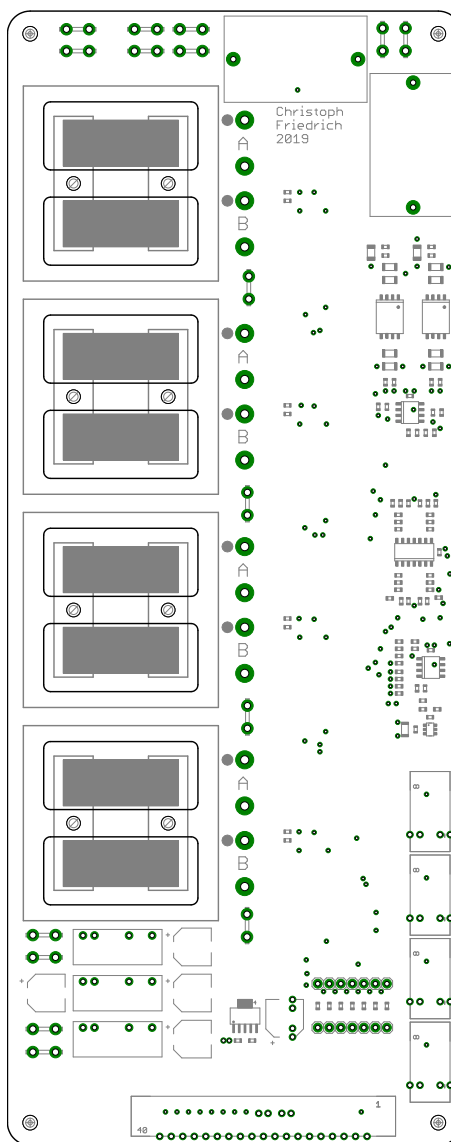


B Platinenlayouts

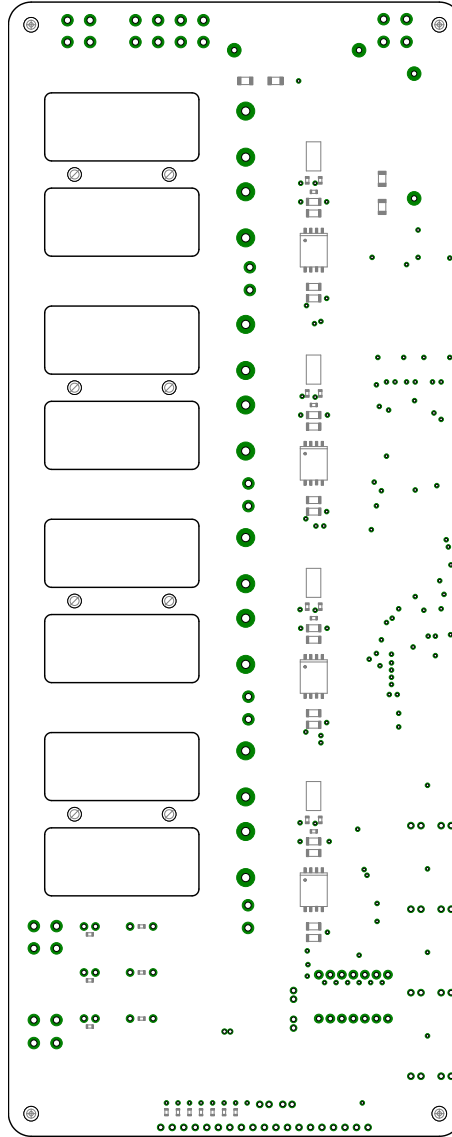
Halbbrücken-Module (Skalierung 70%)



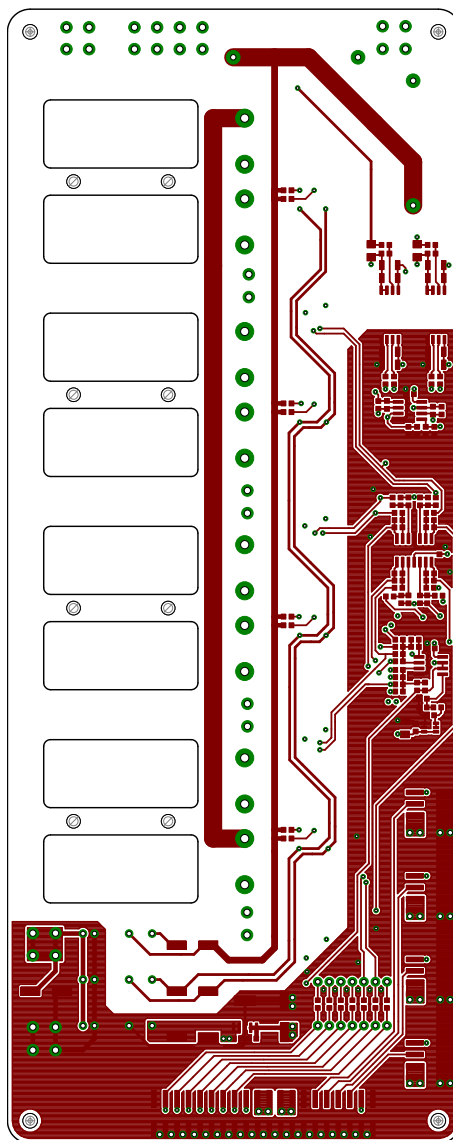
Messplatine (Skalierung 60%), Bauteile Top Layer



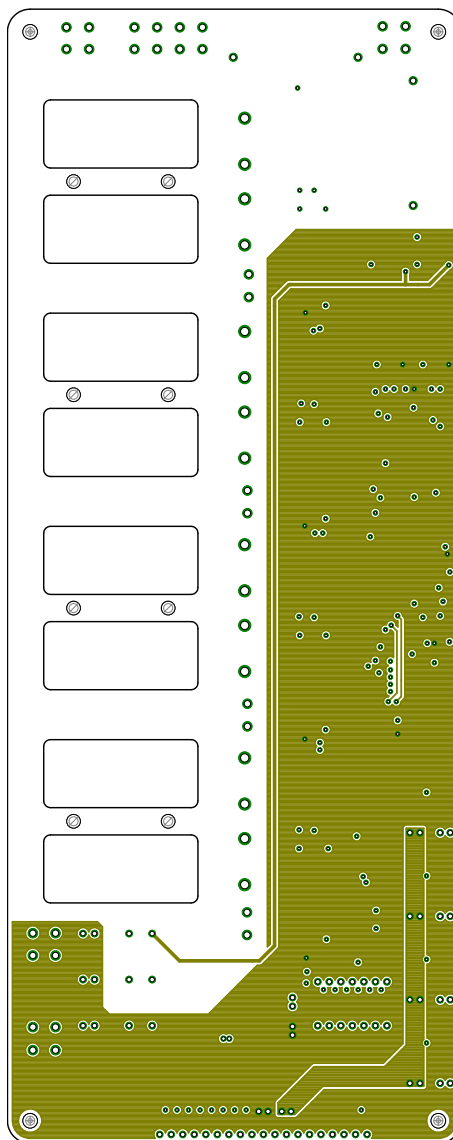
Messplatine (Skalierung 60%), Bauteile Bottom Layer



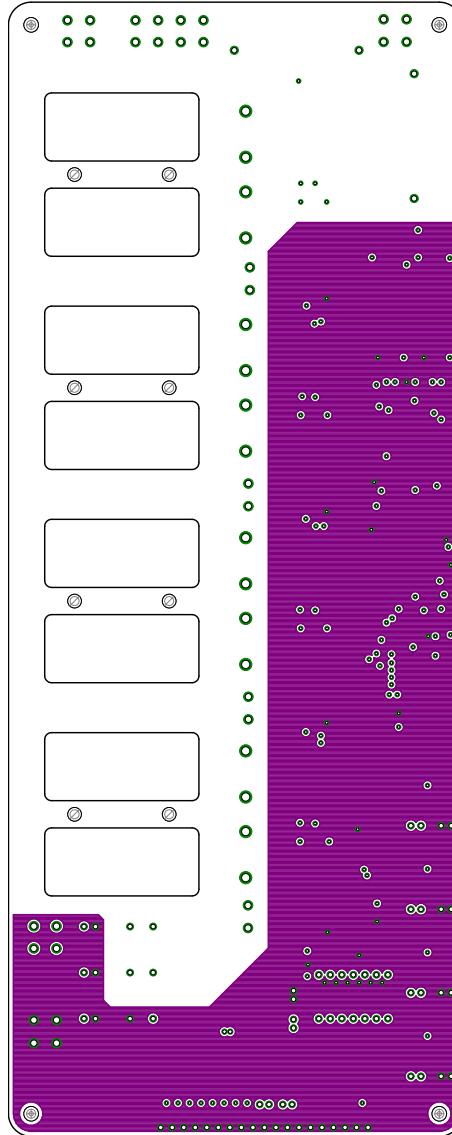
Messplatine (Skalierung 60%), Top Layer



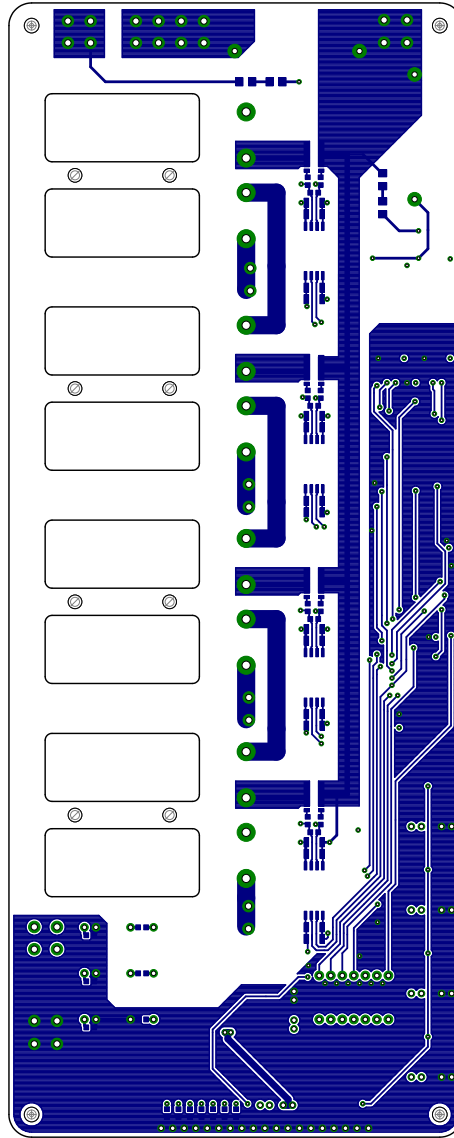
Messplatine (Skalierung 60%), Supply Layer



Messplatine (Skalierung 60%), GND Layer



Messplatine (Skalierung 60%), Bottom Layer



C Bauteillisten

Halbbrücken-Module

Artikel	HerstellerArt. Nr.	Hersteller	Menge pro Platine
IGBT	IKW15N120H3	Infineon	2
Treiber	Si8234BB-D-IS	SiLabs	1
DCDC Wandler 5V->15V, 3kV	NMK0515SAC	Murata	1
C Elko Polymer 330uF, 16V	875075355004	we	1
Abstandshalter Nylon	HTSN-M3-10-6-2	Richco	4
Diode für Ladungspumpe	STTH112U	ST	1
C Elko 220uF, 16V	865060345007	WE	1
C Keramik NP0 0603 100p 50V	C1608C0G1H101J080AA	Kemet	2
C Keramik X7R 0603 100n 50V	C0603C104J5RACTU	Kemet	2
C Keramik X7R 1206 100n 50V	C1206C104J5RACTU	Kemet	1
C Keramik X7R 1206 10u 35V	GMK316AB7106KL-TR	Kemet	3
C Keramik X7T 1u 450V	CKG45NX7T2W105M500JH	TDK	4
R 2512 270k 3W	3522270KFT	TE	8
R 0603 1k	CRG0603F1K0	TE	3
R 0603 100k	CRG0603F100K	TE	2
R 0805 0R	CRCW08050000Z0EA	Vishay	1
R 1206 3R9	ERJP08J3R9V	Panasonic	2
Kühlkörper 250x100x30	132AB1000B	ABL Components	0
Potentiometer	RK09K1130AP5	Alps	0
Buchsenleiste 20 pin gewinkelt	SSQ-120-02-G-S-RA	Samtec	1
R 0603 560R	CPF-A-0603B560RE1	TE	2
IDC Stecker 8x2	MC-254-08-00-ST-SMD	Multicomp	1
C Keramik X7R 1206 270n 50V	C1206C274K5RACTU	Kemet	3
Flachstecker crimp isoliert	3-520408-2	TE	7
Flachstecker pcb	PC250-836LT	Multicomp	7
Elko 120uF, 400V, D18	400KXW120MEFC18X30	Rubycon	4
Stiftleiste 40 pin gewinkelt	61304011021	we	0.5
Ferritkern E42 3C90	E42/21/15-3C90	Ferroxcube	0
Flachstecker male anschraubbar	140814-1	TE	0
Flachbandkabel	535326	3M	0
IDC Stecker 8x2 f. Flachbandk.	MC-254-08-00-00-IDC	Multicomp	0
Lüfter	ME80251V2-0000-A99	Sunon	0
Led 0805 grün	KP-2012CGCK	Kingbright	2

Tabelle C.1: Bauteilliste der Halbbrücken-Module. Mengenangaben pro Modul, es werden vier Module benötigt.

Messplatine

Artikel	HerstellerArt. Nr.	Hersteller	Menge pro Platine
DCDC Wandler	NMK1205SAC	Murata	3
Shunt 3mOhm Kelvin	FC4L64R003FER	Ohmite	4
C Elko Polymer 330uF, 16V	875 075355004	Würth	1
R 0 0603	CR0603-J/-000ELF	Bourns	10
R 8k2 0603 0.1%	ERA3AEB822V	Panasonic	18
R 6k2 0603 0.1%	CPF-A-0603B6K2E1	Panasonic	18
R 3k9 0603 0.1%	CPF-A-0603B3K9E1	Panasonic	4
R 470k 0603 0.1%	CPF0603B470KE	Panasonic	4
R 2k 0603 0.1%	CPF0603B2K0E	Panasonic	2
R 330 0603 0.1%	CPF-A-0603B330RE1	Panasonic	3
C 10n 10V 0603 C0G	C0603C103J3GCAUTO	Kemet	16
C 560p 10V 0603 C0G	C0603C561J5GACTU	Kemet	6
Jumper Wires	MIKROE-511	Mikro El.	0
C 100n 16V 0603 X7R	C0603C104M4RACTU	Kemet	12
C 100n 16V 1206 X7R	VJ1206Y104KXJPW1BC	Vishay	12
C 10u 10V 1206 X7R	CGA5L1X7R1C106K160AC	TDK	14
Elko Polymer 6.3V 680uF	875075155008	Würth	4
C Folie 5u 900V	C4AE0BU4500A11J	Kemet	2
IDC Buchse 8x2 pcb	MC-254-08-00-ST-SMD	Multicomp	4
IDC Buchse 20x2 pcb	61204020621	Würth	1
Flachstecker pcb	PC250-836LT	Multicomp	12
Referenzspg. Quelle 3.3V	REF3433QDBVRQ1	TI	1
Spannungsregler 3.3V	TL1963A-33DCQT	TI	1
OPV 2-fach	OPA2388IDGKT	TI	2
OPV 4-fach	POPA4388IDR	TI	1
OPV iso. AMC1302	AMC1302QDWVRQ1	TI	4
OPV iso. AMC1311	AMC1311BQDWVQ1	TI	2

Tabelle C.2: Bauteilliste der Messplatine. Es wird eine Platine benötigt.

D Software

```
1 /*
2  * TO DO
3  * - Spulenkerne austauschen
4  * - Zusätzlicher Busy Ausgang für Slow Calc
5  * - I1234 Regler ev. auf PI umstellen
6  * - I Regler implementieren
7  * - U Regler implementieren
8  */
9
10 /*
11  * DSP PINOUT
12  *
13  * ADCINA5 I_1
14  * ADCINB5 I_2
15  * ADCINA3 I_3
16  * ADCINB3 I_4
17  * ADCINA4 I_IST
18  * ADCINB7 U_ZK
19  * ADCINB4 U_IST
20  * 20 ADC Status (high=busy, low=idle)
21  * 14 Berechnung Status (high=busy, low=idle)
22  */
23
24 // INCLUDES
25 #include "F2806x_Device.h" // Headerfile Include File
26 #include "F2806x_Examples.h" // Examples Include File
27 #include <math.h>
28 #include "SFO_V6.h"
29
30 // DEFINES
31
32 // Allgemeines
33
34 #define CPU_FREQUENZ_IN_MHZ 90 // CPU Frequenz in MHz, 90MHz ist Maximum
35 #define PWM_FREQ_IN_Hz 25000 // PWM Frequenz für die einzelnen Halbbrücken in Hz (25000 ergibt 100
36 // kHz am Ausgang)
37
38 #define SAMPLING_INTERVALL 10 // 10us (100kHz) Periodendauer für Timer, der ADC-Sampling
39 // aller Kanäle startet, in us
40
41 #define INTERVALL_I_REGLER 20 // 200us (5kHz) Anzahl der Sampling Intervalle für ein I-
42 // Regler Intervall
43 #define INTERVALL_U_REGLER 200 // 2ms (500Hz) Anzahl der Sampling Intervalle für ein U
44 // Regler Intervall
45 #define INTERVALL_I1234_REGLER 200 // 2ms (500Hz) Anzahl der Sampling Intervalle für ein I1234
46 // Regler Intervall
47 #define INTERVALL_SFO 100000 // 1s
48
49 // Fuer Filter
50
51 volatile float filter_display_fg = 50; // Grenzfrequenz in Hz
52 volatile float filter_I_fg = 25000; // Grenzfrequenz in Hz
53 volatile float filter_U_fg = 2500; // Grenzfrequenz in Hz
54 volatile float filter_I1234_fg = 100; // Grenzfrequenz in Hz
55
56 /*
57 volatile float filter_display_fg = 50; // Grenzfrequenz in Hz
58 volatile float filter_I_fg = 50; // Grenzfrequenz in Hz
59 volatile float filter_U_fg = 50; // Grenzfrequenz in Hz
60 volatile float filter_I1234_fg = 50; // Grenzfrequenz in Hz
61 */
62
63 volatile float filter_I1234_alt=0;
64 volatile float filter_I1234_neu=0;
65 volatile float filter_I_alt=0;
66 volatile float filter_I_neu=0;
67 volatile float filter_U_alt=0;
68 volatile float filter_U_neu=0;
69 volatile float filter_display_alt=0;
70 volatile float filter_display_neu=0;
71
72 // Fuer I1234 Regler
```



```

68
69 volatile Uint32 regler_I1234_enable = 0;
70 volatile float regler_I1234_p = 0.001;
71 volatile float regler_I1234_i = 0.000000001;
72 volatile float regler_I1234_limit_max = 0.1;
73 volatile float regler_I1234_limit_min = -0.1;
74 volatile float regler_I1234_Ta = 0;
75
76 volatile float regler_I1_e = 0;
77 volatile float regler_I1_esum = 0;
78 volatile Uint32 regler_I1_integrate = 1;
79 volatile float regler_I2_e = 0;
80 volatile float regler_I2_esum = 0;
81 volatile Uint32 regler_I2_integrate = 1;
82 volatile float regler_I3_e = 0;
83 volatile float regler_I3_esum = 0;
84 volatile Uint32 regler_I3_integrate = 1;
85 volatile float regler_I4_e = 0;
86 volatile float regler_I4_esum = 0;
87 volatile Uint32 regler_I4_integrate = 1;
88
89 volatile float corr1=0;
90 volatile float corr2=0;
91 volatile float corr3=0;
92 volatile float corr4=0;
93 volatile Uint32 pwm1_hr = 0;
94 volatile Uint32 pwm2_hr = 0;
95 volatile Uint32 pwm3_hr = 0;
96 volatile Uint32 pwm4_hr = 0;
97
98 // Fuer I Regler
99 volatile Uint32 regler_I_enable = 0;
100 volatile Uint32 rechteck_I_enable = 0;
101 volatile float Isoll = 0.0;
102 volatile float regler_I_p = 0.00001;
103 volatile float regler_I_i = 0.00001;
104 volatile float regler_I_Ta = 0;
105 volatile float regler_I_esum = 0;
106 volatile Uint32 regler_I_integrate = 1;
107
108 volatile float pwm_reg_min = 0.0;
109 volatile float pwm_reg_max = 0.95;
110
111 // Fuer U Regler
112 volatile Uint32 regler_U_enable = 0;
113 volatile Uint32 rechteck_U_enable = 0;
114 volatile float Usoll = 0.0;
115 volatile float regler_U_p = 0.005;
116 volatile float regler_U_i = 0.000005;
117 volatile float regler_U_Ta = 0;
118 volatile float regler_U_esum = 0;
119 volatile Uint32 regler_U_integrate = 1;
120
121 volatile float Isoll_min = 0.0;
122 volatile float Isoll_max = 20.0;
123
124 // Kurvenformen
125
126 volatile float rechteck_I0 = 1.0;
127 volatile float rechteck_I1 = 2.0;
128 volatile float rechteck_U0 = 10.0;
129 volatile float rechteck_U1 = 30.0;
130 volatile Uint32 rechteck_status = 0;
131 volatile Uint32 count_rechteck = 0;
132
133 // Kalibrierung für Messung
134
135 #define CAL_UZK_0V 0.7
136 #define CAL_UZK_600V 2503.3
137
138 #define CAL_UIST_0V 0.7
139 #define CAL_UIST_600V 2499.5
140
141 #define CAL_I1_0A 2018.2
142 #define CAL_I1_5A 565.9
143
144 #define CAL_I2_0A 2031.2
145 #define CAL_I2_5A 570.8
146
147 #define CAL_I3_0A 2030.4
148 #define CAL_I3_5A 570.8
149
150 #define CAL_I4_0A 2033.1
151 #define CAL_I4_5A 567.9
152
153 #define CAL_IIST_0A 2035.9
154 #define CAL_IIST_5A 569.5
155

```

```

156
157 // Counter und Ablaufsteuerungsvariablen
158 volatile Uint32 TOcount=0;
159 volatile Uint32 counter_I_regler=0;
160 volatile Uint32 counter_U_regler=0;
161 volatile Uint32 counter_I1234_regler=0;
162 volatile Uint32 counter_sfo=0;
163 volatile Uint32 start_I_routine=0;
164 volatile Uint32 start_U_routine=0;
165 volatile Uint32 start_I1234_routine=0;
166 volatile Uint32 start_sfo_routine=0;
167 volatile Uint32 array_writepointer=0;
168 volatile Uint32 U_regler_enable=0;
169 volatile Uint32 I1234_regler_enable=0;
170
171 // Fuer Ausgabe
172 volatile int32 pwm_periode=0;
173 volatile int32 MEP_ScaleFactor_fix = 74;
174 volatile float PWM_fix = 0.5;
175 volatile float pwm_reg = 0;
176
177 volatile int MEP_ScaleFactor;
178 volatile struct EPWM_REGS *ePWM[PWM_CH] = {&EPwm1Regs, &EPwm2Regs, &EPwm3Regs, &EPwm4Regs};
179 volatile int32 sfo_status;
180
181 // Fuer Sampling
182 volatile float I1_adc_array[4];
183 volatile float I2_adc_array[4];
184 volatile float I3_adc_array[4];
185 volatile float I4_adc_array[4];
186
187 volatile float I1_adc=0;
188 volatile float I2_adc=0;
189 volatile float I3_adc=0;
190 volatile float I4_adc=0;
191 volatile float Iist_adc=0;
192 volatile float Uzk_adc=0;
193 volatile float Uist_adc=0;
194
195 volatile float I1_adc_filter = 0;
196 volatile float I2_adc_filter = 0;
197 volatile float I3_adc_filter = 0;
198 volatile float I4_adc_filter = 0;
199 volatile float Iist_adc_filter = 0;
200 volatile float Uzk_adc_filter = 0;
201 volatile float Uist_adc_filter = 0;
202
203
204 // Effektivwerte
205 volatile float Uzk=0;
206 volatile float Uist=0;
207 volatile float I1=0;
208 volatile float I2=0;
209 volatile float I3=0;
210 volatile float I4=0;
211 volatile float Iist=0;
212 volatile float Isum=0;
213 volatile float Iist_neg=0;
214 volatile float Uist_neg=0;
215 volatile float Uzk_neg=0;
216
217
218 // Effektivwerte, sehr stark gefiltert, nur für Anzeige
219 volatile float Uzk_filter = 0;
220 volatile float Uist_filter = 0;
221 volatile float I1_filter = 0;
222 volatile float I2_filter = 0;
223 volatile float I3_filter = 0;
224 volatile float I4_filter = 0;
225 volatile float Iist_filter = 0;
226 volatile float Isum_filter = 0;
227
228
229 // PROTOTYPEN
230
231 void init_gpio(void);
232 void init_timer0(void);
233 void init_adc(void);
234 void init_pwm(void);
235 void pwm(float, float, float, float);
236 float limit(float, float, float, int32*);
237 Uint32 roundToInt(float);
238 void regler_I();
239 void regler_U();
240 void regler_I1234(volatile float *e, volatile float *esum, volatile Uint32 *int_enable, volatile float *
    out);
241
242 // interrupt routines

```

```

243 __interrupt void cpu_timer0_isr(void);
244 __interrupt void adc_isr(void);
245
246
247 // MAIN
248
249 // #####
250
251 int main(void)
252 {
253     // set default values
254     long hilfsvr64 = ((long)CPU_FREQUENZ_IN_MHz*1000000) / (2*(long)PWM_FREQ_IN_Hz); // in cpu steps
255     pwm_periode = (int32)hilfsvr64;
256
257     // initialize System Control (PLL, WatchDog, enable Peripheral Clocks)
258     InitSysCtrl();
259
260     // initialize GPIO:
261     init_gpio();
262
263     // clear all interrupts and initialize PIE vector table, disable CPU interrupts
264     DINT;
265
266     // Initialize PIE control registers to their default state.
267     // The default state is all PIE interrupts disabled and flags are cleared.
268     InitPieCtrl();
269
270     // Disable CPU interrupts and clear all CPU interrupt flags
271     IER = 0x0000;
272     IFR = 0x0000;
273
274     // Initialize the PIE vector table with pointers to the shell Interrupt Service Routines (ISR).
275     // This will populate the entire table, even if the interrupt is not used in this example. This is
276     // useful for debug purposes.
277     InitPieVectTable();
278
279     // Peripherie initialisieren
280     InitCpuTimers();
281     init_timer0();
282     CpuTimer0Regs.TCR.bit.TRB = 1; // timer0 reload
283     CpuTimer0Regs.TCR.bit.TSS = 0; // timer0 start
284     init_adc();
285     init_pwm();
286
287     // Kalibrierung für U und I Messung
288     float cal_I1_k = 5.0/(float)CAL_I1_5A;
289     float cal_I2_k = 5.0/(float)CAL_I2_5A;
290     float cal_I3_k = 5.0/(float)CAL_I3_5A;
291     float cal_I4_k = 5.0/(float)CAL_I4_5A;
292     float cal_Iist_k = 5.0/(float)CAL_IIST_5A;
293     float cal_Uzk_k = 600.0/((float)CAL_UZK_600V-(float)CAL_UZK_0V);
294     float cal_Uist_k = 600.0/((float)CAL_UIST_600V-(float)CAL_UIST_0V);
295
296     // Filterkonstanten ausrechnen
297     filter_I1234_alt = exp((-1)*SAMPLING_INTERVALL*0.000001*2*3.14159265358979323846*filter_I1234_fg);
298     filter_I1234_neu = 1-filter_I1234_alt;
299
300     filter_I_alt = exp((-1)*SAMPLING_INTERVALL*0.000001*2*3.14159265358979323846*filter_I_fg);
301     filter_I_neu = 1-filter_I_alt;
302
303     filter_U_alt = exp((-1)*SAMPLING_INTERVALL*0.000001*2*3.14159265358979323846*filter_U_fg);
304     filter_U_neu = 1-filter_U_alt;
305
306     filter_display_alt = exp((-1)*SAMPLING_INTERVALL*0.000001*2*3.14159265358979323846*filter_display_fg);
307     filter_display_neu = 1-filter_display_alt;
308
309     // Regler Konstanten ausrechnen
310
311     regler_I_Ta = SAMPLING_INTERVALL * INTERVALL_I_REGLER;
312     regler_U_Ta = SAMPLING_INTERVALL * INTERVALL_U_REGLER;
313     regler_I1234_Ta = SAMPLING_INTERVALL * INTERVALL_I1234_REGLER;
314
315     // Sampling Arrays nullsetzen
316     Uint32 i=0;
317     for(i=0; i<4; i++)
318     {
319         I1_adc_array[i]=0;
320         I2_adc_array[i]=0;
321         I3_adc_array[i]=0;
322         I4_adc_array[i]=0;
323     }
324
325     pwm(0.0, 0.0, 0.0, 0.0);
326     GpioDataRegs.GPACLEAR.bit.GPIO21 = 1; // enable drivers
327
328

```

```

329 EINT; // Enable Global interrupt INTM
330 ERTM; // Enable Global realtime interrupt DBGM
331
332 while (SFO() == 0) {}
333
334 while(1)
335 {
336     if(start_I_routine>0)
337     {
338         GpioDataRegs.GPASET.bit.GPIO23 = 1; // testpin, high during calculation
339
340         if(rechteck_I_enable!=0)
341         {
342             count_rechteck++;
343             if(count_rechteck>1000)
344             {
345                 if(rechteck_status==0)
346                 {
347                     rechteck_status=1;
348                     Isoll = rechteck_I0;
349                 }
350                 else
351                 {
352                     rechteck_status=0;
353                     Isoll = rechteck_I1;
354                 }
355                 count_rechteck=0;
356             }
357         }
358
359         Iist = cal_Iist_k * Iist_adc_filter;
360         //Iist = cal_Iist_k * sqrt(Iist_adc_filter);
361         //if(Iist_neg!=0) Iist *= (-1.0);
362
363         if(regler_I_enable!=0)
364         {
365             regler_I();
366             pwm(pwm_reg+corr1, pwm_reg+corr2, pwm_reg+corr3, pwm_reg+corr4);
367         }
368         else
369         {
370             pwm(PWM_fix+corr1, PWM_fix+corr2, PWM_fix+corr3, PWM_fix+corr4);
371         }
372
373         start_I_routine = 0;
374         GpioDataRegs.GPACLEAR.bit.GPIO23 = 1; // testpin, high during calculation
375     }
376
377     if(start_U_routine>0)
378     {
379         GpioDataRegs.GPASET.bit.GPIO14 = 1; // testpin, high during calculation
380
381         Uzk = cal_Uzk_k * sqrt(Uzk_adc_filter);
382         if(Uzk_neg!=0) Uzk *= (-1.0);
383         Uist = cal_Uist_k * sqrt(Uist_adc_filter);
384         if(Uist_neg!=0) Uist *= (-1.0);
385
386         if(regler_U_enable!=0)
387         {
388             regler_U();
389         }
390         if(rechteck_U_enable!=0)
391         {
392             count_rechteck++;
393             if(count_rechteck>300)
394             {
395                 if(rechteck_status==0)
396                 {
397                     rechteck_status=1;
398                     Usoll = rechteck_U0;
399                 }
400                 else
401                 {
402                     rechteck_status=0;
403                     Usoll = rechteck_U1;
404                 }
405                 count_rechteck=0;
406             }
407         }
408
409         // Filterung nur für Anzeige
410         Uzk_filter = Uzk_filter*filter_display_alt + Uzk*filter_display_neu;
411         Uist_filter = Uist_filter*filter_display_alt + Uist*filter_display_neu;
412         I1_filter = I1_filter*filter_display_alt + I1*filter_display_neu;
413         I2_filter = I2_filter*filter_display_alt + I2*filter_display_neu;
414         I3_filter = I3_filter*filter_display_alt + I3*filter_display_neu;
415         I4_filter = I4_filter*filter_display_alt + I4*filter_display_neu;
416         Iist_filter = Iist_filter*filter_display_alt + Iist*filter_display_neu;

```

```

417         Isum_filter = Isum_filter*filter_display_alt + Isum*filter_display_neu;
418
419         start_U_routine = 0;
420         GpioDataRegs.GPACLEAR.bit.GPIO14 = 1; // testpin, high during calculation
421     }
422
423     if(start_I1234_routine>0)
424     {
425         I1 = cal_I1_k * I1_adc_filter;
426         I2 = cal_I2_k * I2_adc_filter;
427         I3 = cal_I3_k * I3_adc_filter;
428         I4 = cal_I4_k * I4_adc_filter;
429
430         Isum = I1 + I2 + I3 + I4;
431
432         if(regler_I1234_enable!=0)
433         {
434             // Funktioniert im Bereich 30V, 300V, nicht aber bei 120V
435             //regler_I1234_p = Uzk_filter * (-0.000045) + 0.01135;
436
437             regler_I1_e = I1-Isum/4;
438             regler_I2_e = I2-Isum/4;
439             regler_I3_e = I3-Isum/4;
440             regler_I4_e = I4-Isum/4;
441
442             regler_I1234(&regler_I1_e, &regler_I1_esum, &regler_I1_integrate, &corr1);
443             regler_I1234(&regler_I2_e, &regler_I2_esum, &regler_I2_integrate, &corr2);
444             regler_I1234(&regler_I3_e, &regler_I3_esum, &regler_I3_integrate, &corr3);
445             regler_I1234(&regler_I4_e, &regler_I4_esum, &regler_I4_integrate, &corr4);
446
447             // Korrekturwert auf 0 normalisieren
448             float corrSum = (corr1+corr2+corr3+corr4)/4;
449             corr1 -= corrSum;
450             corr2 -= corrSum;
451             corr3 -= corrSum;
452             corr4 -= corrSum;
453         }
454         else
455         {
456             corr1=0;
457             corr2=0;
458             corr3=0;
459             corr4=0;
460         }
461
462         start_I1234_routine = 0;
463     }
464
465     if(start_sfo_routine>0)
466     {
467         sfo_status = SFO();
468         if(sfo_status==2)
469         {
470             GpioDataRegs.GPASET.bit.GPIO21 = 1; // disable drivers
471             ESTOP0;
472         }
473         start_sfo_routine = 0;
474     }
475 }
476
477 // -----
478
479
480
481 __interrupt void
482 adc_isr(void)
483 {
484     DINT;
485
486     // Werte aus ADC auslesen
487
488     I1st_adc = (float)AdcResult.ADCRESULT4 - (float)CAL_I1ST_0A;
489     U1st_adc = (float)AdcResult.ADCRESULT6 - (float)CAL_U1ST_0V;
490     Uzk_adc = (float)AdcResult.ADCRESULT5 - (float)CAL_UZK_0V;
491
492     //if(I1st_adc<0) I1st_neg=1; else I1st_neg=0;
493     if(U1st_adc<0) U1st_neg=1; else U1st_neg=0;
494     if(Uzk_adc<0) Uzk_neg=1; else Uzk_neg=0;
495
496     //I1st_adc *= I1st_adc;
497     U1st_adc *= U1st_adc;
498     Uzk_adc *= Uzk_adc;
499
500     I1_adc_array[array_writepointer] = (float)AdcResult.ADCRESULT1 - (float)CAL_I1_0A;
501     I2_adc_array[array_writepointer] = (float)AdcResult.ADCRESULT2 - (float)CAL_I2_0A;
502     I3_adc_array[array_writepointer] = (float)AdcResult.ADCRESULT3 - (float)CAL_I3_0A;
503     I4_adc_array[array_writepointer] = (float)AdcResult.ADCRESULT0 - (float)CAL_I4_0A;
504     array_writepointer++;

```

```

505     if(array_writepointer>3) array_writepointer=0;
506
507     // Einzelströme: Mittelwert über eine Periode (25kHz bzw. 40us) mitteln
508
509     I1_adc = 0;
510     I2_adc = 0;
511     I3_adc = 0;
512     I4_adc = 0;
513     Uint32 i;
514     for(i=0; i<4; i++)
515     {
516         I1_adc += I1_adc_array[i];
517         I2_adc += I2_adc_array[i];
518         I3_adc += I3_adc_array[i];
519         I4_adc += I4_adc_array[i];
520     }
521
522     // alles filtern
523
524     Iist_adc_filter = filter_I_alt * Iist_adc_filter + filter_I_neu * Iist_adc;
525
526     Uist_adc_filter = filter_U_alt * Uist_adc_filter + filter_U_neu * Uist_adc;
527     Uzk_adc_filter  = filter_U_alt * Uzk_adc_filter  + filter_U_neu * Uzk_adc;
528
529     I1_adc_filter = filter_I1234_alt * I1_adc_filter + filter_I1234_neu * I1_adc/4;
530     I2_adc_filter = filter_I1234_alt * I2_adc_filter + filter_I1234_neu * I2_adc/4;
531     I3_adc_filter = filter_I1234_alt * I3_adc_filter + filter_I1234_neu * I3_adc/4;
532     I4_adc_filter = filter_I1234_alt * I4_adc_filter + filter_I1234_neu * I4_adc/4;
533
534     // Regler in main Routine aufrufen
535
536     counter_I_regler++;
537     if(counter_I_regler>INTERVALL_I_REGLER)
538     {
539         start_I_routine=1;
540         counter_I_regler=0;
541     }
542     counter_U_regler++;
543     if(counter_U_regler>INTERVALL_U_REGLER)
544     {
545         start_U_routine=1;
546         counter_U_regler=0;
547     }
548     counter_I1234_regler++;
549     if(counter_I1234_regler>INTERVALL_I1234_REGLER)
550     {
551         start_I1234_routine=1;
552         counter_I1234_regler=0;
553     }
554     counter_sfo++;
555     if(counter_sfo>INTERVALL_SFO)
556     {
557         start_sfo_routine=1;
558         counter_sfo=0;
559     }
560
561     // set round robin pointer to SOC15 -> SOC0 has highest priority now
562     // siehe Datenlatt S.494
563     EALLOW;
564     AdcRegs.SOCPRICL.bit.RRPOINTER = 0x0F;
565     EDIS;
566
567     // Clear ADCINT1 flag reinitialize for next SOC
568     AdcRegs.ADCINTFLGCLR.bit.ADCINT1 = 1;
569     PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;    // Acknowledge interrupt to PIE
570
571     GpioDataRegs.GPACLEAR.bit.GPIO20 = 1;    // set debug Pin low
572
573     EINT;
574 }
575
576 // #####
577
578 void regler_I()
579 {
580     float e = (Isoll-Iist);
581
582     if(regler_I_integrate!=0)
583         regler_I_esum += e;
584     pwm_reg = regler_I_p * e + regler_I_i * regler_I_Ta * regler_I_esum;
585
586     int32 cut;
587     pwm_reg = limit(pwm_reg, pwm_reg_min, pwm_reg_max, &cut);
588
589     // anti wind-up
590     // the integration is stopped when the control variable saturates and the
591     // control error and the control variable have the same sign (i.e., when u · e > 0).
592     if((cut==-1 && e<0) || (cut==1 && e>0))

```

```

593     regler_I_integrate = 0;
594     else
595         regler_I_integrate = 1;
596 }
597
598 void regler_U()
599 {
600     float e = (Usoll-Uist);
601
602     if(regler_U_integrate !=0)
603         regler_U_esum += e;
604     Isoll = regler_U_p * e + regler_U_i * regler_U_Ta * regler_U_esum;
605
606     int32 cut;
607     Isoll = limit(Isoll, Isoll_min, Isoll_max, &cut);
608
609     // anti wind-up
610     // the integration is stopped when the control variable saturates and the
611     // control error and the control variable have the same sign (i.e., when u · e > 0).
612     if((cut==1 && e<0) || (cut==1 && e>0))
613         regler_U_integrate = 0;
614     else
615         regler_U_integrate = 1;
616 }
617
618 void regler_I1234(volatile float *e, volatile float *esum, volatile Uint32 *int_enable, volatile float *
        out)
619 {
620     if(*int_enable !=0)
621         *esum += *e;
622
623     float y = regler_I1234_p * (*e) + regler_I1234_i * regler_I1234_Ta * (*esum);
624
625     int32 cut;
626     y = limit(y, regler_I1234_limit_min, regler_I1234_limit_max, &cut);
627
628     // anti wind-up
629     // the integration is stopped when the control variable saturates and the
630     // control error and the control variable have the same sign (i.e., when u · e > 0).
631     if((cut==1 && *e<0) || (cut==1 && *e>0))
632         *int_enable = 0;
633     else
634         *int_enable = 1;
635
636     *out = y;
637 }
638
639 float limit(float value, float min, float max, int32 *cut)
640 {
641     if(value>max)
642     {
643         *cut=1;
644         return max;
645     }
646     if(value<min)
647     {
648         *cut=-1;
649         return min;
650     }
651     *cut=0;
652     return value;
653 }
654
655 Uint32 roundToInt(float num)
656 {
657     return num < 0 ? num - 0.5 : num + 0.5;
658 }
659
660 void init_gpio(void)
661 {
662     EALLOW;
663
664     // Init EPWM1
665     GpioCtrlRegs.GPAPUD.bit.GPIO0 = 0; // Enable pull-up
666     GpioCtrlRegs.GPAPUD.bit.GPIO1 = 0; // Enable pull-up
667     GpioCtrlRegs.GPAMUX1.bit.GPIO0 = 1; // Configure GPIO40 as EPW7A
668     GpioCtrlRegs.GPAMUX1.bit.GPIO1 = 1; // Configure GPIO44 as EPW7B
669     GpioCtrlRegs.GPADIR.bit.GPIO0 = 1; // Configures the PWM pin as an output
670     GpioCtrlRegs.GPADIR.bit.GPIO1 = 1; // Configures the PWM pin as an output
671
672     // Init EPWM2
673     GpioCtrlRegs.GPAPUD.bit.GPIO2 = 0; // Enable pull-up
674     GpioCtrlRegs.GPAPUD.bit.GPIO3 = 0; // Enable pull-up
675     GpioCtrlRegs.GPAMUX1.bit.GPIO2 = 1; // Configure GPIO42 as EPW8A
676     GpioCtrlRegs.GPAMUX1.bit.GPIO3 = 1; // Configure GPIO43 as EPW8B
677     GpioCtrlRegs.GPADIR.bit.GPIO2 = 1; // Configures the PWM pin as an output
678     GpioCtrlRegs.GPADIR.bit.GPIO3 = 1; // Configures the PWM pin as an output
679 }

```

```

680 // Init EPWM1
681 GpioCtrlRegs.GPAPUD.bit.GPIO4 = 0; // Enable pull-up
682 GpioCtrlRegs.GPAPUD.bit.GPIO5 = 0; // Enable pull-up
683 GpioCtrlRegs.GPAMUX1.bit.GPIO4 = 1; // Configure GPIO40 as EPW7A
684 GpioCtrlRegs.GPAMUX1.bit.GPIO5 = 1; // Configure GPIO44 as EPW7B
685 GpioCtrlRegs.GPADIR.bit.GPIO4 = 1; // Configures the PWM pin as an output
686 GpioCtrlRegs.GPADIR.bit.GPIO5 = 1; // Configures the PWM pin as an output
687
688 // Init EPWM2
689 GpioCtrlRegs.GPAPUD.bit.GPIO6 = 0; // Enable pull-up
690 GpioCtrlRegs.GPAPUD.bit.GPIO7 = 0; // Enable pull-up
691 GpioCtrlRegs.GPAMUX1.bit.GPIO6 = 1; // Configure GPIO42 as EPW8A
692 GpioCtrlRegs.GPAMUX1.bit.GPIO7 = 1; // Configure GPIO43 as EPW8B
693 GpioCtrlRegs.GPADIR.bit.GPIO6 = 1; // Configures the PWM pin as an output
694 GpioCtrlRegs.GPADIR.bit.GPIO7 = 1; // Configures the PWM pin as an output
695
696 // Testpin GPIO20
697 GpioCtrlRegs.GPAMUX2.bit.GPIO20 = 0; // Configure as general purpose I/O
698 GpioDataRegs.GPACLEAR.bit.GPIO20 = 1; // set low
699 GpioCtrlRegs.GPADIR.bit.GPIO20 = 1; // set to output
700
701 // Testpin GPIO23
702 GpioCtrlRegs.GPAMUX2.bit.GPIO23 = 0; // Configure as general purpose I/O
703 GpioDataRegs.GPACLEAR.bit.GPIO23 = 1; // set low
704 GpioCtrlRegs.GPADIR.bit.GPIO23 = 1; // set to output
705
706 // Enable Pin GPIO21: low = enable, high=disable drivers
707 GpioCtrlRegs.GPAMUX2.bit.GPIO21 = 0; // Configure as general purpose I/O
708 GpioDataRegs.GPASET.bit.GPIO21 = 1; // set high
709 GpioCtrlRegs.GPADIR.bit.GPIO21 = 1; // set to output
710
711 // Testpin GPIO20 - high during adc conversion
712 GpioCtrlRegs.GPAMUX2.bit.GPIO20 = 0; // Configure as general purpose I/O
713 GpioDataRegs.GPACLEAR.bit.GPIO20 = 1; // set low
714 GpioCtrlRegs.GPADIR.bit.GPIO20 = 1; // set to output
715
716 // Testpin GPIO14 - high during I1, I2, I3 calculation
717 GpioCtrlRegs.GPAMUX1.bit.GPIO14 = 0; // Configure as general purpose I/O
718 GpioDataRegs.GPACLEAR.bit.GPIO14 = 1; // set low
719 GpioCtrlRegs.GPADIR.bit.GPIO14 = 1; // set to output
720
721 EDIS;
722 }
723
724 void init_pwm(void)
725 {
726     EALLOW;
727     SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC = 0; // Stop all the TB clocks
728
729     EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Up count mode
730     EPwm1Regs.TBPHS.half.TBPHS = 0; // Set Phase register to zero
731     EPwm1Regs.TBCTL.bit.PHSDIR=1; // count up on sync event
732     EPwm1Regs.TBCTL.bit.PHSEN = TB_ENABLE; // Master module
733     EPwm1Regs.TBCTL.bit.PRDL = TB_SHADOW;
734     EPwm1Regs.TBCTL.bit.SYNCSEL = TB_CTR_ZERO;
735     EPwm1Regs.TBCTL.bit.CLKDIV = 0;
736     EPwm1Regs.TBCTL.bit.HSPCLKDIV = 0; // High Speed Time-base Clock Prescale Bits 1
737     EPwm1Regs.TBCTL.bit.FREE_SOFT = 2; // Emulation Mode Bits = Free run
738     EPwm1Regs.AQCTLA.bit.CAU = AQ_CLEAR;
739     EPwm1Regs.AQCTLA.bit.CAD = AQ_SET;
740     EPwm1Regs.AQCTLB.bit.CBU = AQ_CLEAR;
741     EPwm1Regs.AQCTLB.bit.CBD = AQ_SET;
742     EPwm1Regs.DBFED = 0; // Falling Edge Delay
743     EPwm1Regs.DBRED = 0; // Rising Edge Delay
744     EPwm1Regs.DBCTL.bit.POLSEL = 0;
745     EPwm1Regs.DBCTL.bit.OUT_MODE = 0;
746     EPwm1Regs.CMPCTL.bit.SHADOWMODE = CC_SHADOW;
747     EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
748     EPwm1Regs.CMPCTL.bit.LOADMODE = CC_CTR_ZERO; // load on TBCTR = Zero
749     EPwm1Regs.CMPCTL.bit.LOADEMODE = CC_CTR_ZERO; // load on TBCTR = Zero
750     EPwm1Regs.TBPRD = (unsigned int) pwm_periode;
751
752     EPwm1Regs.HRCNFG.all = 0x0;
753     EPwm1Regs.HRCNFG.bit.EDGMODE = HR_BEP;
754     EPwm1Regs.HRCNFG.bit.CTLMODE = HR_CMP;
755     EPwm1Regs.HRCNFG.bit.HRLOAD = HR_CTR_ZERO_PRD;
756     EPwm1Regs.HRCNFG.bit.AUTOCNV = 1; // Enable autoconversion for HR period
757     EPwm1Regs.HRPCTL.bit.HRPE = 1; // Turn on high-resolution period control.
758     EPwm1Regs.HRPCTL.bit.PHSEN = 1; // Turn on high-resolution period control.
759     EPwm1Regs.HRPCTL.bit.TBPHSHRLOADE = 1; // Enable TBPHSHR sync (required for updown count HR control)
760
761     EPwm2Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Up count mode
762     EPwm2Regs.TBPHS.half.TBPHS = (int32)(pwm_periode/2); // Set Phase register
763     EPwm2Regs.TBCTL.bit.PHSDIR=0; // count down on sync event
764     EPwm2Regs.TBCTL.bit.PHSEN = TB_ENABLE; // Master module
765     EPwm2Regs.TBCTL.bit.PRDL = TB_SHADOW;
766

```



```

768 EPwm2Regs.TBCTL.bit.SYNCSEL = TB_SYNC_IN;
769 EPwm2Regs.TBCTL.bit.CLKDIV = 0;
770 EPwm2Regs.TBCTL.bit.HSPCLKDIV = 0; // High Speed Time-base Clock Prescale Bits 1
771 EPwm2Regs.TBCTL.bit.FREE_SOFT = 2; // Emulation Mode Bits = Free run
772 EPwm2Regs.AQCTLA.bit.CAU = AQ_CLEAR; // set actions for EPWM2A
773 EPwm2Regs.AQCTLA.bit.CAD = AQ_SET;
774 EPwm2Regs.AQCTLB.bit.CBU = AQ_CLEAR; // set actions for EPWM2B
775 EPwm2Regs.AQCTLB.bit.CBD = AQ_SET;
776 EPwm2Regs.DBFED = 0; // Falling Edge Delay
777 EPwm2Regs.DBRED = 0; // Rising Edge Delay
778 EPwm2Regs.DBCTL.bit.POLSEL = 0;
779 EPwm2Regs.DBCTL.bit.OUT_MODE = 0;
780 EPwm2Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
781 EPwm2Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
782 EPwm2Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on TBCTR = Zero
783 EPwm2Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on TBCTR = Zero
784 EPwm2Regs.TBPRD = (unsigned int) pwm_periode;
785
786 EPwm2Regs.HRCNFG.all = 0x0;
787 EPwm2Regs.HRCNFG.bit.EDGMODE = HR_BEP;
788 EPwm2Regs.HRCNFG.bit.CTLMODE = HR_CMP;
789 EPwm2Regs.HRCNFG.bit.HRLOAD = HR_CTR_ZERO_PRD;
790 EPwm2Regs.HRCNFG.bit.AUTOCNV = 1; // Enable autoconversion for HR period
791 EPwm2Regs.HRPCTL.bit.HRPE = 1; // Turn on high-resolution period control.
792 EPwm2Regs.TBCTL.bit.PHSEN = 1; // Turn on high-resolution period control.
793 EPwm2Regs.HRPCTL.bit.TBPHSHRLOADE = 1; // Enable TBPHSHR sync (required for updown count HR control)
794
795 EPwm3Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Up count mode
796 EPwm3Regs.TBPHS.half.TBPHS = (int32)(pwm_periode); // Set Phase register to zero
797 EPwm3Regs.TBCTL.bit.PHSDIR=0; // count down on sync event
798 EPwm3Regs.TBCTL.bit.PHSEN = TB_ENABLE; // Master module
799 EPwm3Regs.TBCTL.bit.PRDL = TB_SHADOW;
800 EPwm3Regs.TBCTL.bit.SYNCSEL = TB_SYNC_IN;
801 EPwm3Regs.TBCTL.bit.CLKDIV = 0;
802 EPwm3Regs.TBCTL.bit.HSPCLKDIV = 0; // High Speed Time-base Clock Prescale Bits 1
803 EPwm3Regs.TBCTL.bit.FREE_SOFT = 2; // Emulation Mode Bits = Free run
804 EPwm3Regs.AQCTLA.bit.CAU = AQ_CLEAR; // set actions for EPWM2A
805 EPwm3Regs.AQCTLA.bit.CAD = AQ_SET;
806 EPwm3Regs.AQCTLB.bit.CBU = AQ_CLEAR; // set actions for EPWM2B
807 EPwm3Regs.AQCTLB.bit.CBD = AQ_SET;
808 EPwm3Regs.DBFED = 0; // Falling Edge Delay
809 EPwm3Regs.DBRED = 0; // Rising Edge Delay
810 EPwm3Regs.DBCTL.bit.POLSEL = 0;
811 EPwm3Regs.DBCTL.bit.OUT_MODE = 0;
812 EPwm3Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
813 EPwm3Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
814 EPwm3Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on TBCTR = Zero
815 EPwm3Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on TBCTR = Zero
816 EPwm3Regs.TBPRD = (unsigned int) pwm_periode;
817
818 EPwm3Regs.HRCNFG.all = 0x0;
819 EPwm3Regs.HRCNFG.bit.EDGMODE = HR_BEP;
820 EPwm3Regs.HRCNFG.bit.CTLMODE = HR_CMP;
821 EPwm3Regs.HRCNFG.bit.HRLOAD = HR_CTR_ZERO_PRD;
822 EPwm3Regs.HRCNFG.bit.AUTOCNV = 1; // Enable autoconversion for HR period
823 EPwm3Regs.HRPCTL.bit.HRPE = 1; // Turn on high-resolution period control.
824 EPwm3Regs.TBCTL.bit.PHSEN = 1; // Turn on high-resolution period control.
825 EPwm3Regs.HRPCTL.bit.TBPHSHRLOADE = 1; // Enable TBPHSHR sync (required for updown count HR control)
826
827 EPwm4Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Up count mode
828 EPwm4Regs.TBPHS.half.TBPHS = (int32)(pwm_periode/2); // Set Phase register to
zero
829 EPwm4Regs.TBCTL.bit.PHSDIR=1; // count up on sync event
830 EPwm4Regs.TBCTL.bit.PHSEN = TB_ENABLE; // Master module
831 EPwm4Regs.TBCTL.bit.PRDL = TB_SHADOW;
832 EPwm4Regs.TBCTL.bit.SYNCSEL = TB_SYNC_IN;
833 EPwm4Regs.TBCTL.bit.CLKDIV = 0;
834 EPwm4Regs.TBCTL.bit.HSPCLKDIV = 0; // High Speed Time-base Clock Prescale Bits 1
835 EPwm4Regs.TBCTL.bit.FREE_SOFT = 2; // Emulation Mode Bits = Free run
836 EPwm4Regs.AQCTLA.bit.CAU = AQ_CLEAR; // set actions for EPWM2A
837 EPwm4Regs.AQCTLA.bit.CAD = AQ_SET;
838 EPwm4Regs.AQCTLB.bit.CBU = AQ_CLEAR; // set actions for EPWM2B
839 EPwm4Regs.AQCTLB.bit.CBD = AQ_SET;
840 EPwm4Regs.DBFED = 0; // Falling Edge Delay
841 EPwm4Regs.DBRED = 0; // Rising Edge Delay
842 EPwm4Regs.DBCTL.bit.POLSEL = 0;
843 EPwm4Regs.DBCTL.bit.OUT_MODE = 0;
844 EPwm4Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
845 EPwm4Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
846 EPwm4Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on TBCTR = Zero
847 EPwm4Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on TBCTR = Zero
848 EPwm4Regs.TBPRD = (unsigned int) pwm_periode;
849
850 EPwm4Regs.HRCNFG.all = 0x0;
851 EPwm4Regs.HRCNFG.bit.EDGMODE = HR_BEP;
852 EPwm4Regs.HRCNFG.bit.CTLMODE = HR_CMP;
853 EPwm4Regs.HRCNFG.bit.HRLOAD = HR_CTR_ZERO_PRD;
854 EPwm4Regs.HRCNFG.bit.AUTOCNV = 1; // Enable autoconversion for HR period

```

```

855     EPwm4Regs.HRPCTL.bit.HRPE = 1;           // Turn on high-resolution period control.
856     EPwm4Regs.TBCTL.bit.PHSEN = 1;         // Turn on high-resolution period control.
857     EPwm4Regs.HRPCTL.bit.TBPHSHRLOADE = 1; // Enable TBPHSHR sync (required for updown count HR control)
858
859     SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC = 1; // Start all the timers synced
860     EDIS;
861 }
862
863
864 void pwm(float pwm1, float pwm2, float pwm3, float pwm4)
865 {
866     if(pwm1>1.0) pwm1=1.0;
867     if(pwm1<0.0) pwm1=0.0;
868
869     if(pwm2>1.0) pwm2=1.0;
870     if(pwm2<0.0) pwm2=0.0;
871
872     if(pwm3>1.0) pwm3=1.0;
873     if(pwm3<0.0) pwm3=0.0;
874
875     if(pwm4>1.0) pwm4=1.0;
876     if(pwm4<0.0) pwm4=0.0;
877
878     // If you load CMPA/CMPB on zero, then use CMPA/CMPB values greater than or equal to 1.
879     // If you load CMPA/CMPB on period, then use CMPA/CMPB values less than or equal to TBPRD-1.
880     // This means there will always be a pulse of at least one TBCLK cycle in a PWM period which, when
881     // very short, tend to be ignored by the system.
882
883     // calculate coarse and fine values for high resolution pwm
884     Uint32 pwm1_coarse = (Uint32)((float)pwm1*(float)pwm_periode);
885     Uint32 pwm1_fine = (Uint32) (((float)pwm1*(float)pwm_periode-(float)pwm1_coarse)*(float)
886     MEP_ScaleFactor+1.5));
887
888     Uint32 pwm2_coarse = (Uint32)((float)pwm2*(float)pwm_periode);
889     Uint32 pwm2_fine = (Uint32) (((float)pwm2*(float)pwm_periode-(float)pwm2_coarse)*(float)
890     MEP_ScaleFactor+1.5));
891
892     Uint32 pwm3_coarse = (Uint32)((float)pwm3*(float)pwm_periode);
893     Uint32 pwm3_fine = (Uint32) (((float)pwm3*(float)pwm_periode-(float)pwm3_coarse)*(float)
894     MEP_ScaleFactor+1.5));
895
896     Uint32 pwm4_coarse = (Uint32)((float)pwm4*(float)pwm_periode);
897     Uint32 pwm4_fine = (Uint32) (((float)pwm4*(float)pwm_periode-(float)pwm4_coarse)*(float)
898     MEP_ScaleFactor+1.5));
899
900     pwm1_hr = (pwm1_coarse*256 + pwm1_fine);
901     pwm2_hr = (pwm2_coarse*256 + pwm2_fine);
902     pwm3_hr = (pwm3_coarse*256 + pwm3_fine);
903     pwm4_hr = (pwm4_coarse*256 + pwm4_fine);
904
905     EPwm1Regs.CMPA.all = (Uint32) pwm2_hr*256;
906     EPwm2Regs.CMPA.all = (Uint32) pwm3_hr*256;
907     EPwm3Regs.CMPA.all = (Uint32) pwm4_hr*256;
908     EPwm4Regs.CMPA.all = (Uint32) pwm1_hr*256;
909 }
910
911 void init_adc(void)
912 {
913     EALLOW; // This is needed to write to EALLOW protected register
914     PieVectTable.ADCINT1 = &adc_isr;
915     EDIS; // This is needed to disable write to EALLOW protected registers
916
917     // initialize adc
918     InitAdc();
919     AdcOffsetSelfCal();
920
921     // folgende siehe Datenblatt ab S.489
922
923     // Enable ADCINT1 in PIE
924     PieCtrlRegs.PIEIER1.bit.INTx1 = 1; // Enable INT 1.1 in the PIE
925     IER |= M_INT1; // Enable CPU Interrupt 1
926
927     // Configure ADC
928     EALLOW;
929
930     // enable ADC
931     AdcRegs.ADCCTL1.bit.ADCENABLE = 1;
932     AdcRegs.ADCCTL2.bit.ADCNONOVERLAP = 1; // Enable non-overlap mode
933
934     // generell: eoc (end of conversion) interrupt wird ausgelöst wenn results abholbereit
935     AdcRegs.ADCCTL1.bit.INTPULSEPOS = 1;
936
937     // ADCINT1 aktivieren
938     AdcRegs.INTSEL1N2.bit.INT1E = 1; // Enabled ADCINT1
939     AdcRegs.INTSEL1N2.bit.INT1CONT = 0; // Disable ADCINT1 Continuous mode
940
941     // ADCINT1 dem SOC6 zuweisen (d.h. nachdem SOC6 fertig ist, wird interrupt ADCINT1 ausgelöst)
942     AdcRegs.INTSEL1N2.bit.INT1SEL = 6;

```

```

938
939 // beim ADC start wird SOC0, SOC1, etc. nacheinander abgearbeitet
940 // Alle SOCs haben den Selben Trigger (CPU Timer 0), wenn der ausgelöst wird werden alle SOCs ausgelö
941 // Gestartet wird immer beim niedrigsten SOC, bei uns SOC0, danach alle nacheinander (round-robin
942 // Prinzip)
943 // siehe Datenlatt S.515
944
945 // SOC0 konfigurieren:
946 AdcRegs.ADCSOC0CTL.bit.CHSEL = 0x05; // set SOC0 channel select to ADCINA5 (I_1)
947 AdcRegs.ADCSOC0CTL.bit.TRIGSEL = 1; // set SOC0 start trigger to CPU Timer 0
948 AdcRegs.ADCSOC0CTL.bit.ACQPS = 6; // set SOC0 S/H Window to 7 ADC Clock Cycles, (6 ACQPS plus
949 // 1) - ist minimum
950
951 // SOC1 konfigurieren:
952 AdcRegs.ADCSOC1CTL.bit.CHSEL = 0x0D; // set SOC1 channel select to ADCINB5 (I_2)
953 AdcRegs.ADCSOC1CTL.bit.TRIGSEL = 1;
954 AdcRegs.ADCSOC1CTL.bit.ACQPS = 6;
955
956 // SOC2 konfigurieren:
957 AdcRegs.ADCSOC2CTL.bit.CHSEL = 0x03; // set SOC2 channel select to ADCINA3 (I_3)
958 AdcRegs.ADCSOC2CTL.bit.TRIGSEL = 1;
959 AdcRegs.ADCSOC2CTL.bit.ACQPS = 6;
960
961 // SOC3 konfigurieren:
962 AdcRegs.ADCSOC3CTL.bit.CHSEL = 0x0B; // set SOC3 channel select to ADCINB3 (I_4)
963 AdcRegs.ADCSOC3CTL.bit.TRIGSEL = 1;
964 AdcRegs.ADCSOC3CTL.bit.ACQPS = 6;
965
966 // SOC6 konfigurieren:
967 AdcRegs.ADCSOC4CTL.bit.CHSEL = 0x04; // set SOC6 channel select to ADCINA4 (I_IST)
968 AdcRegs.ADCSOC4CTL.bit.TRIGSEL = 1;
969 AdcRegs.ADCSOC4CTL.bit.ACQPS = 6;
970
971 // SOC4 konfigurieren:
972 AdcRegs.ADCSOC5CTL.bit.CHSEL = 0x0F; // set SOC4 channel select to ADCINB7 (U_ZK)
973 AdcRegs.ADCSOC5CTL.bit.TRIGSEL = 1;
974 AdcRegs.ADCSOC5CTL.bit.ACQPS = 6;
975
976 // SOC5 konfigurieren:
977 AdcRegs.ADCSOC6CTL.bit.CHSEL = 0x0C; // set SOC5 channel select to ADCINB4 (U_IST)
978 AdcRegs.ADCSOC6CTL.bit.TRIGSEL = 1;
979 AdcRegs.ADCSOC6CTL.bit.ACQPS = 6;
980
981 // set SOC0-SOC3 to high priority, SOC4-SOC15 are in normal mode
982 //AdcRegs.SOCPRICL.bit.SOCPRRIORITY = 0x04;
983
984 EDIS;
985 }
986
987
988 void init_timer0(void)
989 {
990     EALLOW; // This is needed to write to EALLOW protected registers
991     PieVectTable.TINT0 = &cpu_timer0_isr;
992     EDIS;
993
994     // Configure CPU-Timer 0 to interrupt every 500 milliseconds
995     // (set prescaler and timer value)
996     // 80MHz CPU Freq, 500 millisecond Period (in uSeconds)
997     // ConfigCpuTimer(&CpuTimer0, 80, 500000);
998     ConfigCpuTimer(&CpuTimer0, 90, SAMPLING_INTERVALL);
999
1000     // Set Timer0 config to default value
1001     CpuTimer0Regs.TCR.all = 0x0000;
1002
1003     // Enable Timer0 Interrupt
1004     CpuTimer0Regs.TCR.bit.TIE = 1;
1005
1006     // Stop Timer0
1007     CpuTimer0Regs.TCR.bit.TSS = 1;
1008
1009     // Load Timer0 with timer value and prescaler
1010     // The timer counts down from this value and triggers an interrupt
1011     // if it reaches 0
1012     CpuTimer0Regs.TCR.bit.TRB = 1;
1013
1014     IER |= M_INT1; // Enable CPU Interrupt 1
1015
1016     PieCtrlRegs.PIEIER1.bit.INTx7 = 1;
1017 }
1018
1019 // timer interrupt routines
1020 __interrupt void cpu_timer0_isr(void)
1021 {
1022     /*

```

```
1023     * Hier muss nichts gemacht werden.  
1024     * ADC wird später parallel zum Timer 0 Interrupt getriggert  
1025     * -> diese ISR kann später entfernt werden  
1026     */  
1027  
1028     T0count++;  
1029  
1030     GpioDataRegs.GPASET.bit.GPIO20 = 1;        // set debug Pin high  
1031  
1032     PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;  
1033 }
```

Literatur

- [1] F. Maislinger, H. Ertl, G. Stojcic und F. Holzner, „Efficiency and Motor-Performance Improvement Using WBG-Inverters with Observer-based Actively Damped LC-Sine Wave Filters“, in *PCIM Europe 2019; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management*, Mai 2019, S. 1–9.
- [2] L. A. Saunders, G. L. Skibinski, S. T. Evon und D. L. Kempkes, „Riding the reflected wave-IGBT drive technology demands new motor and cable considerations“, in *Proceedings of 1996 IAS Petroleum and Chemical Industry Technical Conference*, Sep. 1996, S. 75–84.
- [3] P. Nussbaumer, C. Zoeller, T. M. Wolbank und M. A. Vogelsberger, „Transient distribution of voltages in induction machine stator windings resulting from switching of power electronics“, in *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, Nov. 2013, S. 3189–3194.
- [4] I. G. Park und S. I. Kim, „Modeling and analysis of multi-interphase transformers for connecting power converters in parallel“, eng, in *PESC97. Record 28th Annual IEEE Power Electronics Specialists Conference. Power Processing and Electronic Specialists Conference 1997*, Bd. 2, IEEE, 1997, 1164–1170 vol.2, ISBN: 0780338405.
- [5] J. Ertl, *Skript Leistungselektronik und Stromrichtertechnik*. 2012.
- [6] T. Fuchslueger, M. Vogelsberger und H. Ertl, „Reducing the dv/dt of Motor Inverters by a Two Leg Resonant Switching Cell“, in *PCIM Europe 2018; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management*, Juni 2018, S. 1–8.
- [7] J. Specovius, *Grundkurs Leistungselektronik*, 7., aktualisierte u. überarb. Aufl. 2015. Wiesbaden : Springer Fachmedien Wiesbaden : Imprint: Springer Vieweg, ISBN: 3-658-03309-6.
- [8] Infineon, *Datenblatt IKW15N120H3*, Dez. 2014. Adresse: https://www.infineon.com/dgdl/Infineon-IKW15N120H3-DataSheet-v02_01-EN.pdf (besucht am 03.12.2019).
- [9] Silicon Laboratories Inc., *Datenblatt SI8234*, Sep. 2019. Adresse: <https://www.silabs.com/documents/public/data-sheets/Si823x.pdf> (besucht am 03.12.2019).

- [10] TDK, *Datenblatt E42/21/20 Ferrite Core*, Mai 2013. Adresse: https://www.tdk-electronics.tdk.com/inf/80/db/fer/e_42_21_20.pdf (besucht am 03.12.2019).
- [11] Texas Instruments, *Datenblatt TMS320F28069*, Mai 2018. Adresse: <http://www.ti.com/lit/ds/symlink/tms320f28069.pdf> (besucht am 03.12.2019).
- [12] Texas Instruments, *LAUNCHXL-F28069M Overview*, März 2019. Adresse: <http://www.ti.com/lit/ug/sprui11b/sprui11b.pdf> (besucht am 03.12.2019).
- [13] Texas Instruments, *TMS320x280x, 2801x, 2804x High Resolution Pulse Width Modulator (HRPWM) Reference Guide*, Aug. 2011. Adresse: <https://www.ti.com/lit/ug/spru924f/spru924f.pdf> (besucht am 03.12.2019).
- [14] Texas Instruments, *Datenblatt AMC1302*, Sep. 2018. Adresse: <http://www.ti.com/lit/ds/symlink/amc1302.pdf> (besucht am 03.12.2019).
- [15] Texas Instruments, *Datenblatt AMC1311*, Sep. 2018. Adresse: <http://www.ti.com/lit/ds/symlink/amc1311.pdf> (besucht am 03.12.2019).
- [16] G. J. Capella, J. Pou, S. Ceballos, J. Zaragoza und V. G. Agelidis, „Current-Balancing Technique for Interleaved Voltage Source Inverters With Magnetically Coupled Legs Connected in Parallel“, *IEEE Transactions on Industrial Electronics*, Jg. 62, Nr. 3, S. 1335–1344, März 2015, ISSN: 0278-0046.
- [17] T. Fuchslueger, H. Ertl und M. A. Vogelsberger, „Reducing dv/dt of Motor Inverters by Staggered-Edge Switching of Multiple Parallel SiC Half-Bridge Cells“, in *PCIM Europe 2017; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management*, Mai 2017, S. 1–8.

Eidesstattliche Erklärung

Hiermit erkläre ich, dass die vorliegende Arbeit gemäß dem Code of Conduct – Regeln zur Sicherung guter wissenschaftlicher Praxis (in der aktuellen Fassung des jeweiligen Mitteilungsblattes der TU Wien), insbesondere ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel, angefertigt wurde. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder in ähnlicher Form in anderen Prüfungsverfahren vorgelegt.

Wien, im Dezember 2019

Christoph Friedrich