



Modellierung der Umrichternichtlinearität bei Spannungszwischenkreisumrichtern

Masterarbeit

durchgeführt von

Klaus Erhart

Institut für Elektrische Antriebstechnik und Maschinen
der Technischen Universität Graz

Leiterin: Univ.-Prof. Dr.-Ing. Annette Mütze

Betreuer

Ass.-Prof. Dipl.-Ing. Dr.techn. Roland Seebacher

A-8010 Graz, Inffeldgasse 18/1
Tel.: +43 (0) 316 873 – 7241
Fax: +43 (0) 316 873 – 7244
<http://www.eam.tugraz.at>
<http://www.tugraz.at>

Graz, im Juni 2014

Für Rudolf und Elfriede

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am 4. Juni 2014

A handwritten signature in blue ink, reading "Klaus Eberhart". The signature is written in a cursive style with a prominent loop at the end of the last name.

.....
(Unterschrift)

Kurzfassung

In der feldorientierten Regelung kommt es aufgrund des Schätzfehlers im Flussbeobachter, hervorgerufen durch die Umrichternichtlinearität des Spannungszwischenkreisumrichters, zu einem suboptimalen Regelungsverhalten. Ziel der vorliegenden Arbeit ist es, diese Umrichternichtlinearität in einem MATLAB/Simulink-Modell zu implementieren. Durch die schrittweise erfolgende Einbindung von Verriegelungszeit, Spannungsabfällen an den leistungselektronischen Bauteilen (IGBT und Freilaufdiode) sowie der parasitären Kapazität am Ausgang des Spannungszwischenkreisumrichters, kann dieses Ziel erreicht werden. Die einzelnen Modellierungsschritte wurden mit Messungen, durchgeführt bei unterschiedlichen Zwischenkreisspannungen, validiert. Dadurch wird anhand von diesem Modell eine nützliche Basis für die Kompensation des nichtlinearen Spannungsabfalles am Spannungszwischenkreisumrichter geschaffen.

Abstract

In field-oriented control, the nonlinearity of the voltage source inverter leads to an estimation error in the flux observer. In turn, this results in suboptimal control behavior. The goal of this Master thesis is to implement the aforementioned inverter nonlinearity in a MATLAB/Simulink model. This implementation is attained by gradual integration of deadtime, voltage drop in power electronic components (IGBT and flyback diode), and parasitic capacitance at the output of the voltage source inverter. In order to validate the individual modelling increments, measurements were performed at different DC link voltages. This model should, therefore, provide a useful basis for the compensation of the nonlinear voltage drop occurring in voltage source inverters and give further insights into the mechanisms involved.

Inhaltsverzeichnis

Eidesstattliche Erklärung	i
Kurzfassung.....	ii
Abstract.....	iii
1 Einleitung.....	1
2 Der Spannungszwischenkreisumrichter und die Messung der Umrichternichtlinearität.....	3
2.1 Aufbau des Spannungszwischenkreisumrichters.....	3
2.1.1 Pulsweitenmodulation.....	4
2.1.2 Grundlegende Eigenschaften.....	5
2.1.3 Insulated Gate Bipolar Transistor (IGBT)	5
2.1.4 Der Pulswechselrichter	7
2.2 Messaufbau.....	7
2.2.1 Messungen mit dem Signalprozessorsystem DS 1005	10
2.2.2 Messungen mit dem Leistungsmessgerät N5000	10
2.2.3 Messschaltung	11
2.3 Messablauf.....	11
2.4 Messergebnisse für die Umrichternichtlinearität.....	12
3 Ideales Modell des Spannungszwischenkreisumrichters.....	14
3.1 Funktionsweise einer Halbbrücke zur Erzeugung eines idealen PWM-Signals ...	14
3.1.1 Positiver Strom	15
3.1.2 Negativer Strom.....	17
3.1.3 Diskussion zur Abhängigkeit vom Stromvorzeichen	18
3.2 Modellierung eines idealen PWM-Signals.....	19
3.2.1 Implementierung mit S-Function.....	20
3.2.2 Eingangs- bzw. Ausgangssignale der S-Function	21
3.2.3 Simulationsablauf der S-Function	22
3.2.4 Diskussion der Simulationsergebnisse.....	23
3.3 Berechnung des Mittelwerts (Gleichwert)	26
3.3.1 mean-Funktion.....	26
3.3.2 MATLAB/Simulink-Modell	26
3.3.3 Validierung der Mittelwertmessung	27
3.4 Ohmsch-induktive Last	31
3.4.1 Mathematisches Modell der Last.....	32
3.4.2 Validierung des idealen Modells	34
3.4.3 Diskussion der Validierung	34
3.5 Vergleich mit Messung	37

4	Modell mit Verriegelungszeit.....	38
4.1	Verriegelungszeit.....	38
4.1.1	Strom positiv.....	39
4.1.2	Strom negativ.....	41
4.1.3	Strom null.....	43
4.1.4	Diskussion zur Abhängigkeit vom Stromvorzeichen.....	43
4.2	Modellerweiterung für die Berücksichtigung der Verriegelungszeit.....	44
4.2.1	Eingangs- bzw. Ausgangssignale der S-Function.....	44
4.2.2	Simulationsablauf in der S-Function.....	45
4.2.3	Funktionstest des erweiterten Modells.....	47
4.2.4	Testsimulation für die Umrichternichtlinearität.....	48
4.2.5	Analyse der Simulationsprobleme.....	50
4.3	Modellmodifizierung zur Berücksichtigung des Stromlückens.....	57
4.3.1	Änderungen in der S-Function.....	57
4.3.2	Ausgangsspannung u_1	58
4.3.3	Testsimulation für die Umrichternichtlinearität.....	60
4.4	Validierung des Modells mit Verriegelungszeit.....	61
5	Modell mit Spannungsabfällen an IGBT und Freilaufdiode.....	63
5.1	Spannungsabfälle an den elektronischen Bauelementen.....	63
5.1.1	Strom positiv.....	64
5.1.2	Strom negativ.....	66
5.1.3	Strom null.....	68
5.2	Kennlinien von IGBT und Freilaufdiode.....	69
5.2.1	Diskussion über die Berücksichtigung des Sperrverhaltens von IGBT und Freilaufdiode.....	70
5.2.2	Ermittlung der Kennlinie für die Freilaufdiode.....	75
5.2.3	Ermittlung der Kennlinie für den IGBT.....	78
5.2.4	Zusammensetzung der beiden Kennlinien.....	80
5.3	Modellerweiterung für die Berücksichtigung der Spannungsabfälle an den elektronischen Bauelementen.....	83
5.3.1	Eingangs- bzw. Ausgangssignale der S-Function.....	84
5.3.2	Simulationsablauf in der S-Function.....	85
5.3.3	Ausgangsspannung u_1	85
5.4	Validierung des Modells mit Spannungsabfällen an IGBT und Freilaufdiode.....	86
6	Modell unter Berücksichtigung der parasitärer Kapazität.....	89
6.1	Die parasitäre Kapazität.....	89
6.1.1	Strom positiv.....	90
6.1.2	Strom negativ.....	94
6.1.3	Strom null.....	97
6.2	Modellerweiterung für die Berücksichtigung der parasitären Kapazität.....	98
6.2.1	Mathematisches Modell mit der parasitären Kapazität.....	98

6.2.2	Implementierung in MATLAB/Simulink	101
6.2.3	Untersuchung der langen Simulationsdauer.....	106
6.3	Testen des Modells mit parasitärer Kapazität	107
6.3.1	Strom positiv	107
6.3.2	Strom negativ.....	109
6.3.3	Strom null.....	110
6.4	Validierung des Modells mit parasitärer Kapazität	111
6.4.1	Größenordnung der parasitäre Kapazität	111
6.4.2	Vergleich der Simulation mit der Messung.....	112
7	Resümee und Ausblick.....	114
	Literaturverzeichnis	121
A	Anhang	123
A.1	MATLAB-Funktion „varhit2ideal.m“	123
A.2	MATLAB-Funktion „varhit4tvi.m“	128
A.3	MATLAB-Funktion „varhit4tvii.m“	132
A.4	MATLAB-Funktion „varhit4spabf.m“	136
A.5	MATLAB-Funktion „varhit4pk.m“	140
B	Datenblatt (IGBT, Freilaufdiode)	144

1 Einleitung

Stromrichter sind in der modernen elektrischen Antriebstechnik als Stellglieder nicht mehr wegzudenken. Sowohl für Gleichstromantriebe als auch für Drehstromantriebe sind sie seit Jahrzehnten beinahe flächendeckend im Einsatz. Die feldorientierte Regelung (FOR) einer Asynchronmaschine im Labor verwendet einen Spannungszwischenkreisumrichter als Stellglied. In der FOR werden der Betrag des Rotorflusses und dessen Winkel im Rotorkoordinatensystem mit einem Flussbeobachter geschätzt. Ungenauigkeiten in der Schätzung führen zu einem suboptimalen Regelverhalten. Dafür mitverantwortlich ist die Umrichternichtlinearität im Spannungszwischenkreisumrichter. Ersichtlich wird dies im zeitlichen Verlauf des momentbildenden Stromes $i_{s,q}$ bei geringer Drehzahl, dargestellt im Bild 1.1. In dieser Abbildung sind die Phasenströme, sowie der momentbildende Strom $i_{s,q}$ zu sehen. Dabei ist ersichtlich, dass bei den Nulldurchgängen der Phasenströme im Funktionsverlauf von $i_{s,q}$ „Kerben“ auftreten. Diese werden durch die Umrichternichtlinearität verursacht.

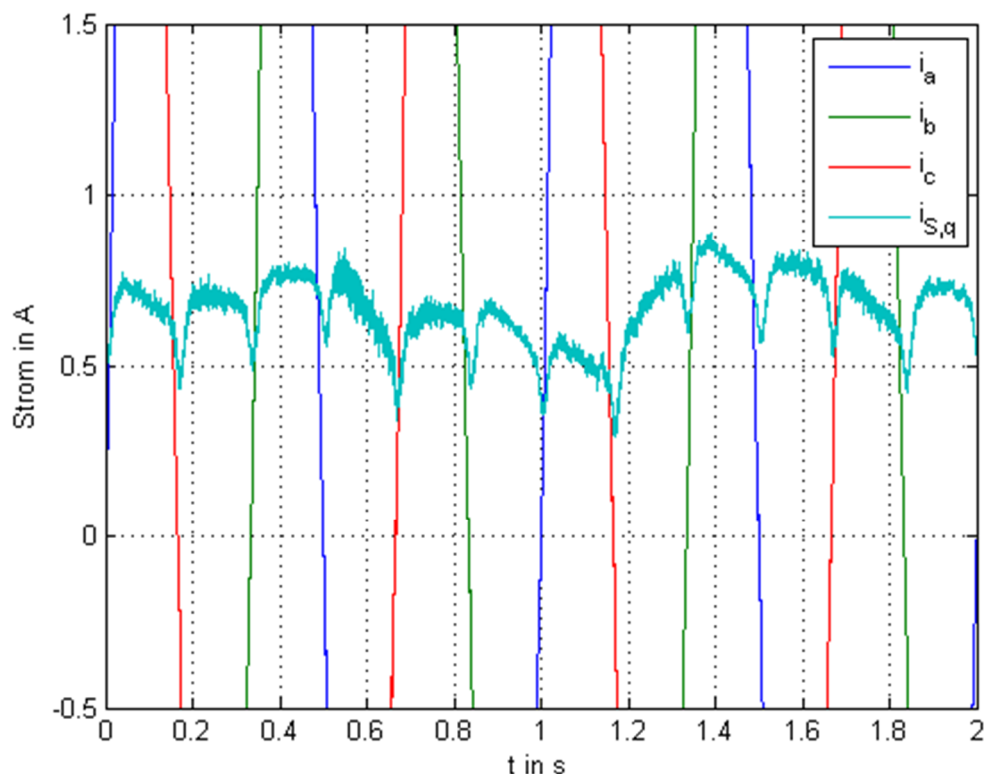


Bild 1.1: Auswirkungen der Umrichternichtlinearität auf den momentbildenden Strom $i_{s,q}$

Aufgrund des suboptimalen Regelverhaltens entstand die Intention, ein Simulationsmodell für Spannungszwischenkreisumrichter zu entwickeln, um die einflussreichsten Ursachen

der Umrichternichtlinearität erstens aufzudecken und im darauffolgenden Schritt mittels Kompensation ausgleichen zu können.

Um die Umrichternichtlinearität kompensieren zu können, wird in dieser Arbeit ein Modell in MATLAB/Simulink entwickelt, das möglichst einfach in der Struktur ist, die Realität jedoch gut approximiert. In der Literatur [1], [2] und [3] sind vor allem drei Faktoren für den nichtlinearen Spannungsabfall am Spannungszwischenkreisumrichter ausschlaggebend. Diese sind die Verriegelungszeit, die Spannungsabfälle an den leistungselektronischen Bauelementen (IGBT bzw. Freilaufdiode) und die parasitäre Kapazität am Ausgang des Zwischenkreisumrichters. Ausgehend von einem vorhandenen idealen Modell für den Spannungszwischenkreisumrichter werden im Verlauf dieser Arbeit die vorhin genannten Einflussfaktoren der Reihe nach berücksichtigt und in das Modell implementiert.

Um die Modellierungsschritte auf ihre Genauigkeit überprüfen zu können, wird die Umrichternichtlinearität vermessen und mit den Simulationsergebnissen verglichen. Dabei wird unter anderem auch der Wert der Zwischenkreisspannung variiert und dessen Auswirkung auf das Modell untersucht.

Dies führt zu folgenden Aufgabenstellungen

- Vermessung der Umrichternichtlinearität bei Variation der Zwischenkreisspannung
- Untersuchung des Verhaltens einer Halbbrücke mit dem idealen Modell
- Diskussion des Verhaltens von Strom und Spannung in einer Pulsperiode bei Berücksichtigung der Verriegelungszeit
- Implementierung der Verriegelungszeit in das vorhandene ideale Modell
- Untersuchung des Verhaltens der Halbbrücke bei Berücksichtigung der Spannungsabfälle an IGBT und der Freilaufdiode
- Ermittlung der Durchlasskennlinien von IGBT und der Freilaufdiode
- Implementierung der Spannungsabfälle von IGBT und der Freilaufdiode im Modell
- Untersuchung des Verhaltens von Strom und Spannung in der Halbbrücke bei Berücksichtigung der parasitären Kapazität
- Implementierung der parasitären Kapazität im Modell
- Validierung der Simulationsergebnisse mit den gemessenen Kennlinien der Umrichternichtlinearität

2 Der Spannungszwischenkreisumrichter und die Messung der Umrichternichtlinearität

In diesem einführenden Kapitel wird der verwendete Spannungszwischenkreisumrichter beschrieben. Dabei werden der prinzipielle Aufbau, die Pulsweitenmodulation und die abschaltbaren leistungselektronischen Schalter (IGBT) kurz erklärt. Danach folgt eine Beschreibung des Messaufbaus im Labor und der Messschaltung, mit deren Hilfe die Umrichternichtlinearität gemessen werden kann. Zum Schluss werden die erhaltenen Messergebnisse vorgestellt.

2.1 Aufbau des Spannungszwischenkreisumrichters

Spannungszwischenkreisumrichter (engl. Voltage Source Inverter, VSI) ist eine mögliche Technologie für Umrichter in der elektrischen Antriebstechnik. In der Fachliteratur sind auch die Bezeichnungen „Wechselrichter mit eingepprägter Gleichspannung“, „Umrichter mit Gleichspannungszwischenkreis“ oder kurz „U-Umrichter“ gebräuchlich. Bei Zwischenkreisumrichtern erfolgt im Gegensatz zum Direktumrichter die Umwandlung der Energie in zwei Schritten. Zuerst geschieht das Gleichrichten, im zweiten Schritt das Wechselrichten. Zwischen den beiden Umwandlungsschritten passiert die Energiespeicherung. Im Zwischenkreisumrichter mit Gleichspannungszwischenkreis erfolgt die Energiespeicherung in einem Kondensator. Durch diese große Zwischenkreis-Kapazität wird dem Wechselrichter eine quasi lastunabhängige Gleichspannung eingepragt. Der Kondensator im Zwischenkreis entkoppelt die beiden fremdfrequenten Ein- und Ausgangsnetze. Das Eingangsnetz ist das öffentliche Netz. Dieses weist eine konstante Spannung und Frequenz auf, während das Ausgangsnetz mit variabler Frequenz und variabler Spannung arbeitet. Im Bild 2.1 ist die prinzipielle Struktur eines Spannungszwischenkreisumrichters dargestellt.

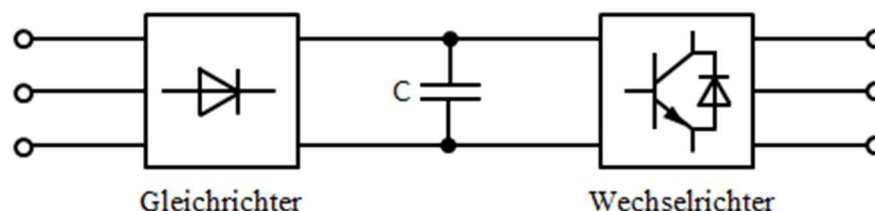


Bild 2.1: Prinzipieller Aufbau eines Spannungszwischenkreisumrichters

Der Gleichspannungszwischenkreis wird in diesem Fall mit einem ungesteuerten Brückengleichrichter (B6) gespeist. Der ungesteuerte Eingangsstromrichter sorgt für eine konstante Zwischenkreisspannung U_{zk} . Mittels „Pulsen“ von U_{zk} wird die gewünschte

Frequenz bzw. Spannung am Ausgang des Spannungszwischenkreisumrichters erzeugt. Somit ist der Wechselrichter in diesem Fall ein Pulswechselrichter mit Pulsweitenmodulation (engl. Pulse-Width Modulation, PWM).

2.1.1 Pulsweitenmodulation

Bei der Pulsweitenmodulation wird mit dem Tastverhältnis (engl. Duty cycle) eine gewünschte Spannungszeitfläche eingestellt. Da wir in dieser Arbeit für die Untersuchungen zur Modellbildung der Umrichternichtlinearität bei U-Umrichtern nur konstante Spannungen verwenden, kann über das Tastverhältnis, die geforderte Gleichspannung am Umrichterausgang, eingestellt werden. Das Tastverhältnis d wird auch als Tastrate bezeichnet und liegt im Wertebereich von $0 \leq d \leq 1$. Die Fälle $d = 0$ und $d = 1$ sind keine Pulse, da kein Ein- bzw. Ausschalten stattfindet. Somit ist für $d = 0$ die Ausgangsspannung am Pulswechselrichter über die ganze Pulsperiode null. Für $d = 1$ hingegen immer auf dem Wert der Zwischenkreisspannung U_{zk} . Für alle anderen Fälle der Tastrate d ist für eine Zeitspanne t_{aus} die Spannung null und für t_{ein} ist die Spannung auf U_{zk} . Im Bild 2.2 ist dies graphisch dargestellt. Daraus ergibt sich für das Tastverhältnis

$$d = \frac{t_{ein}}{T_a} \quad (2.1)$$

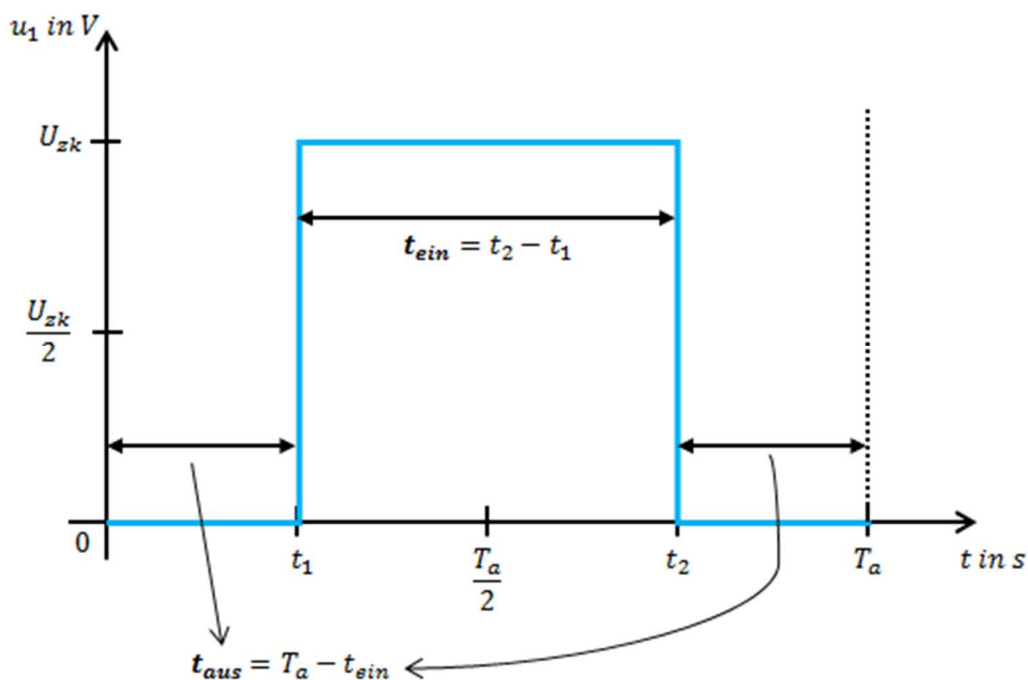


Bild 2.2: PWM-Puls

Im verwendeten Pulsverfahren ist der PWM-Puls symmetrisch um die halbe Pulsperiode angeordnet. Im Bild 2.3 ist das Pulsmuster für die Tastverhältnisse 0, 0,25, 0,5, 0,75 und 1 abgebildet.

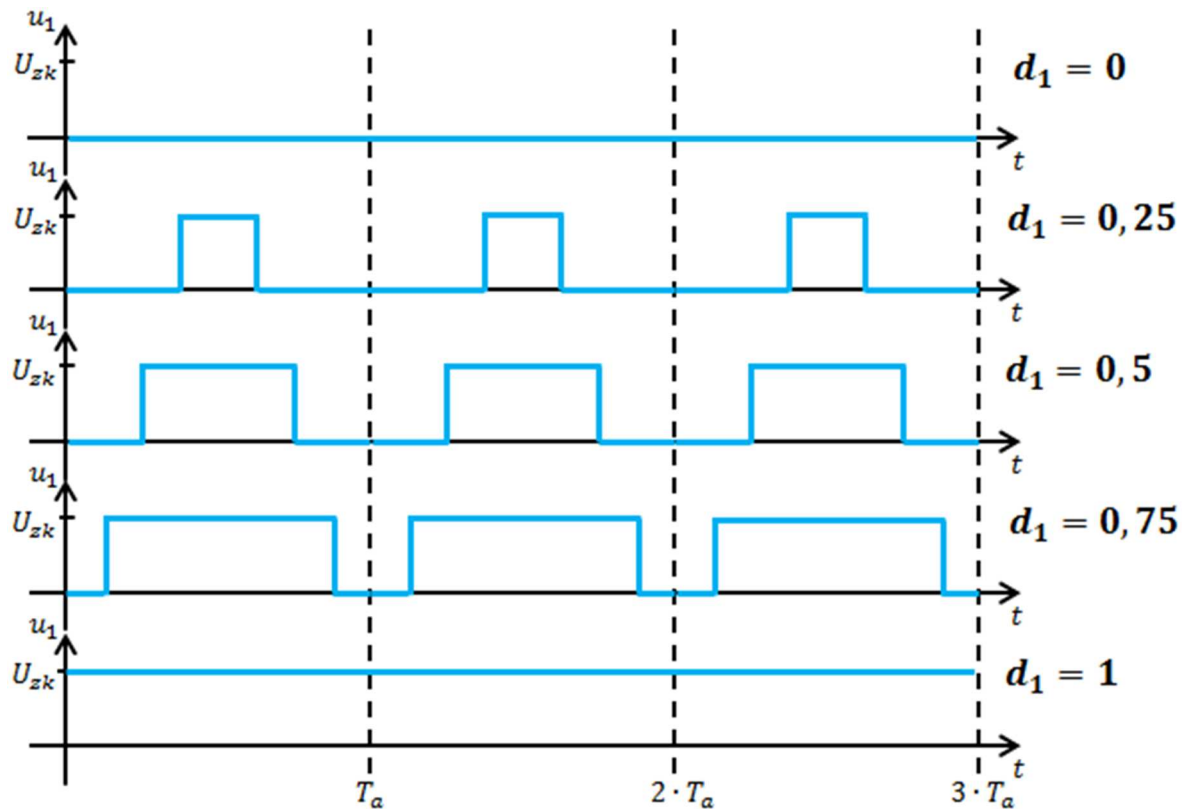


Bild 2.3: Pulsmuster für verschiedene Tastverhältnisse d_1

Für weitere Pulsverfahren und Details wird an dieser Stelle auf die einschlägige Fachliteratur verwiesen [4], [5], [6], [7], [8] und [9].

2.1.2 Grundlegende Eigenschaften

Der verwendete Spannungszwischenkreisumrichter hat einen Zweipunkt-Wechselrichter (engl. 2-level inverter). Dies bedeutet, dass der ausgegebene Spannungsverlauf am Ausgang aus zwei Spannungswerten zusammengesetzt ist. In diesem Fall schaltet jede Halbbrücke zwischen 0 V und U_{zk} .

Ein selbstgeführter Wechselrichter hat ein- und ausschaltbare leistungselektronische Schalter die in dieser Arbeit IGBT's (Insulated Gate Bipolar Transistor) sind. D.h. das Öffnen und Schließen der Ventile erfolgt zu gewünschten Zeitpunkten und ist unabhängig von Netz bzw. Last.

2.1.3 Insulated Gate Bipolar Transistor (IGBT)

Der Insulated Gate Bipolar Transistor (IGBT) vereint die Vorteile des bipolaren Transistors mit denen des Feldeffekttransistors. Der Bipolartransistor sorgt für geringe Durchlassverluste. Der Feldeffekttransistor (FET) liefert den hochohmigen Eingangssteuerteil. Der IGBT hat die im Bild 2.4 dargestellte Halbleiterstruktur. Prinzipiell

entsteht der IGBT aus einem MOSFET indem er die Dotierung einer Schicht verändert. Die Schicht n mit dem Drain-Kontakt wird zu p mit dem Kollektor-Kontakt.

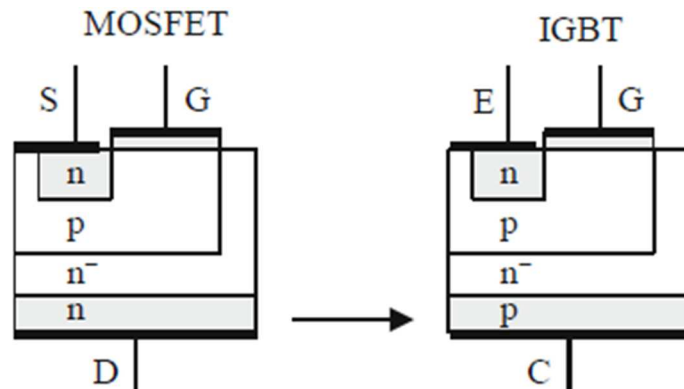


Bild 2.4: Ableitung der IGBT-Struktur vom MOSFET aus [6]

Das Ersatzschaltbild des IGBT ist im Bild 2.5 abgebildet. Wie aus der Halbleiterstruktur ersichtlich, setzt sich der IGBT aus einem Bipolartransistor T2 und einem MOSFET T1 zusammen. Für den normalen Betrieb ist dieses vereinfachte Ersatzschaltbild ausreichend.

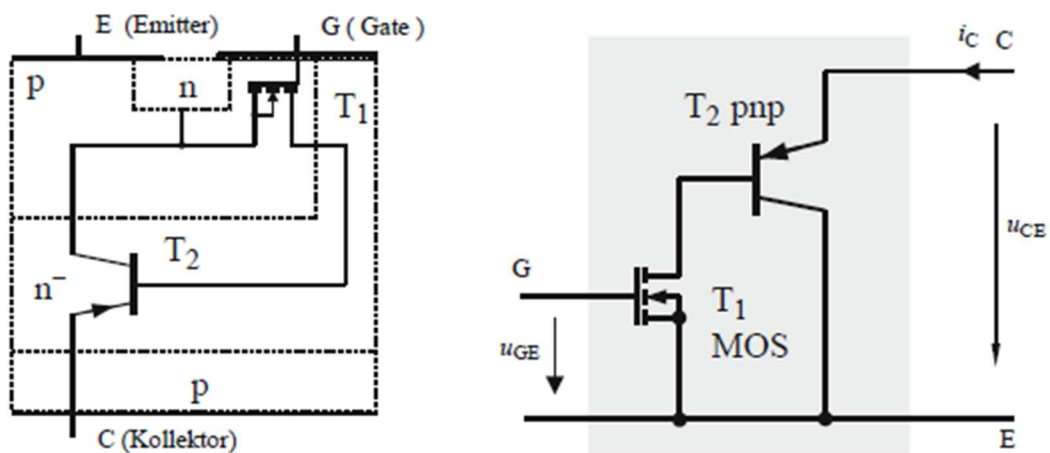


Bild 2.5: Ersatzschaltbild des IGBT aus [6]

Das Schaltsymbol für den IGBT repräsentiert auch die Vermischung von Bipolartransistor und Feldeffekttransistor und ist im Bild 2.6 zu sehen.

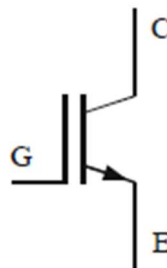


Bild 2.6: Schaltsymbol des IGBT aus [6]

2.1.4 Der Pulswechselrichter

Im prinzipiellen Aufbau des Spannungszwischenkreisumrichters im Bild 2.1 wird nun der Wechselrichter genauer im Bild 2.7 dargestellt. Dabei sind drei Halbbrücken mit jeweils zwei IGBT's zu sehen. Jedes elektronische Ventil hat eine antiparallele Diode, da der IGBT den Strom nur in eine Richtung leiten kann. Die Stromrichtung die der IGBT führen kann, wird durch den Pfeil im Schaltsymbol angegeben. Bei einem Wechsel des Stromvorzeichens in einer Halbbrücke übernimmt die antiparallele Diode die Stromführung.

Für ohmsch-induktive Belastung des Pulswechselrichters übernimmt die Diode dann die Stromführung, wenn der IGBT den Strom leitet und dabei geöffnet wird. Da sich bei einer Induktivität der Strom nicht sprunghaft verändert, ändert sich die Spannung sprunghaft, so dass die dadurch veränderte Ausgangsspannung der Halbbrücke den Strom über die Diode treiben kann. Die Diode wird dadurch auch als Freilaufdiode (engl. Flyback diode) bezeichnet. Eine detaillierte Betrachtung der Schaltvorgänge und der dadurch bedingte Wechsel der Strompfade werden in den nachfolgenden Kapiteln vorgestellt.

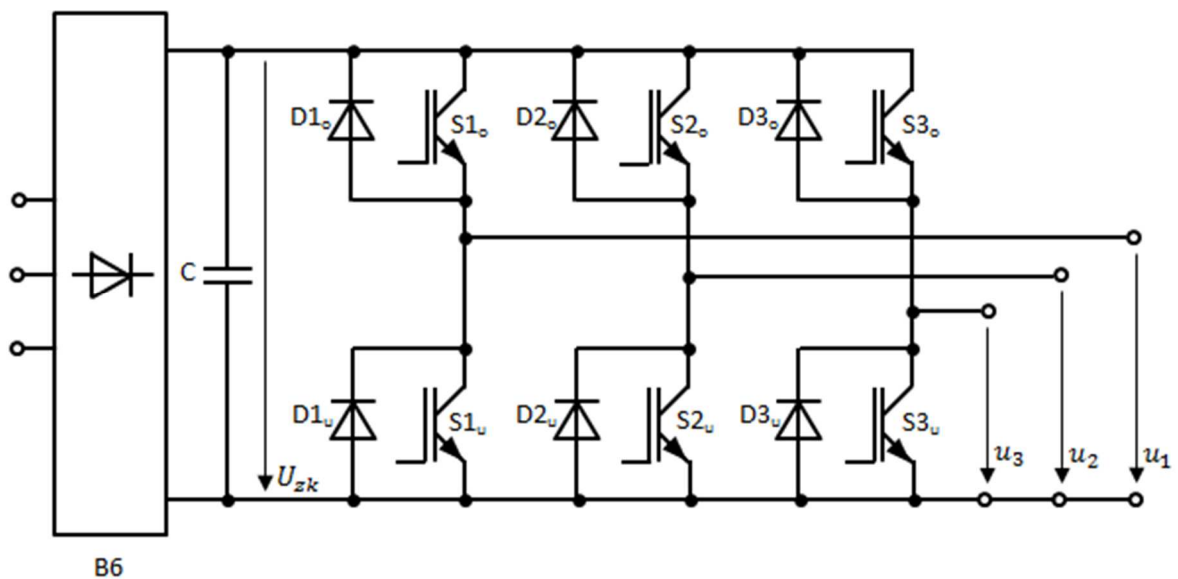


Bild 2.7: Spannungszwischenkreisumrichter mit detaillierter Schaltung des Pulswechselrichters

Die Umrichternichtlinearität und deren verschiedenen Einflussgrößen treten im Pulswechselrichter zu Tage. Damit konzentrieren sich die nachfolgenden Untersuchungen auf den Pulswechselrichter.

2.2 Messaufbau

Der prinzipielle Messaufbau im Labor ist im Bild 2.8 dargestellt. Der Gleichrichter (GR) mit dem ungesteuerten Brückengleichrichter (B6) wird aus dem öffentlichen Netz versorgt. Dazwischen befindet sich noch ein Stelltransformator, mit dem die

Zwischenkreisspannung eingestellt werden kann. Im Zwischenkreis befindet sich die Ladeeinrichtung für den Zwischenkreiskondensator. Der Zwischenkreis mit annähernd konstanter Zwischenkreisspannung U_{zk} speist den Pulswechselrichter (PWR). Die drei Ausgänge des Pulswechselrichters führen über eine Strommessung, die für die Feldorientierte Regelung (FOR) vorhanden ist. Danach folgt die Messschaltung die an den ersten beiden Halbbrücken hängt. Diese ist im Bild 2.10 abgebildet.

Mit dem Signalprozessorsystem DS 1005 werden die Pulsmuster des PWR gesteuert. Ein PC ist mit dem Signalprozessor mittels Netzwerk verbunden. Somit können vom Rechner aus die gewünschten Tastverhältnisse vorgegeben werden. Auch die Pulsperiode T_a wird über diesen Weg eingestellt. Mit dem Signalprozessorsystem werden sowohl der Pulswechselrichter gesteuert, als auch Messungen durchgeführt. Zusätzlich ist das Leistungsmessgerät N5000 vorhanden und über Ethernet mit dem PC verbunden.

Am Signalprozessorsystem DS 1005 und dem Leistungsmessgerät N5000 sind im Bild 2.8 die jeweiligen Messgrößen eingezeichnet. Augenblickswertverläufe können nur mit dem Leistungsmessgerät N5000 gemessen werden. Diese werden auch in dieser Arbeit wie üblich mit Kleinbuchstaben bezeichnet (u_1 , i_1). Mittelwerte werden mit beiden Messsystemen gemessen. Beim Mittelwert handelt es sich um den sogenannten Gleichwert bzw. linearen (arithmetischen) Mittelwert

$$\bar{i}_1 = \frac{1}{T} \int_0^T i_1(t) \cdot dt. \quad (2.2)$$

Der Gleichwert wird im weiteren Verlauf der Arbeit mit überstrichenen Kleinbuchstaben gekennzeichnet (\bar{u}_1 , \bar{i}_1). Wird im Folgenden vom Mittelwert gesprochen, ist damit der Gleichwert gemeint.

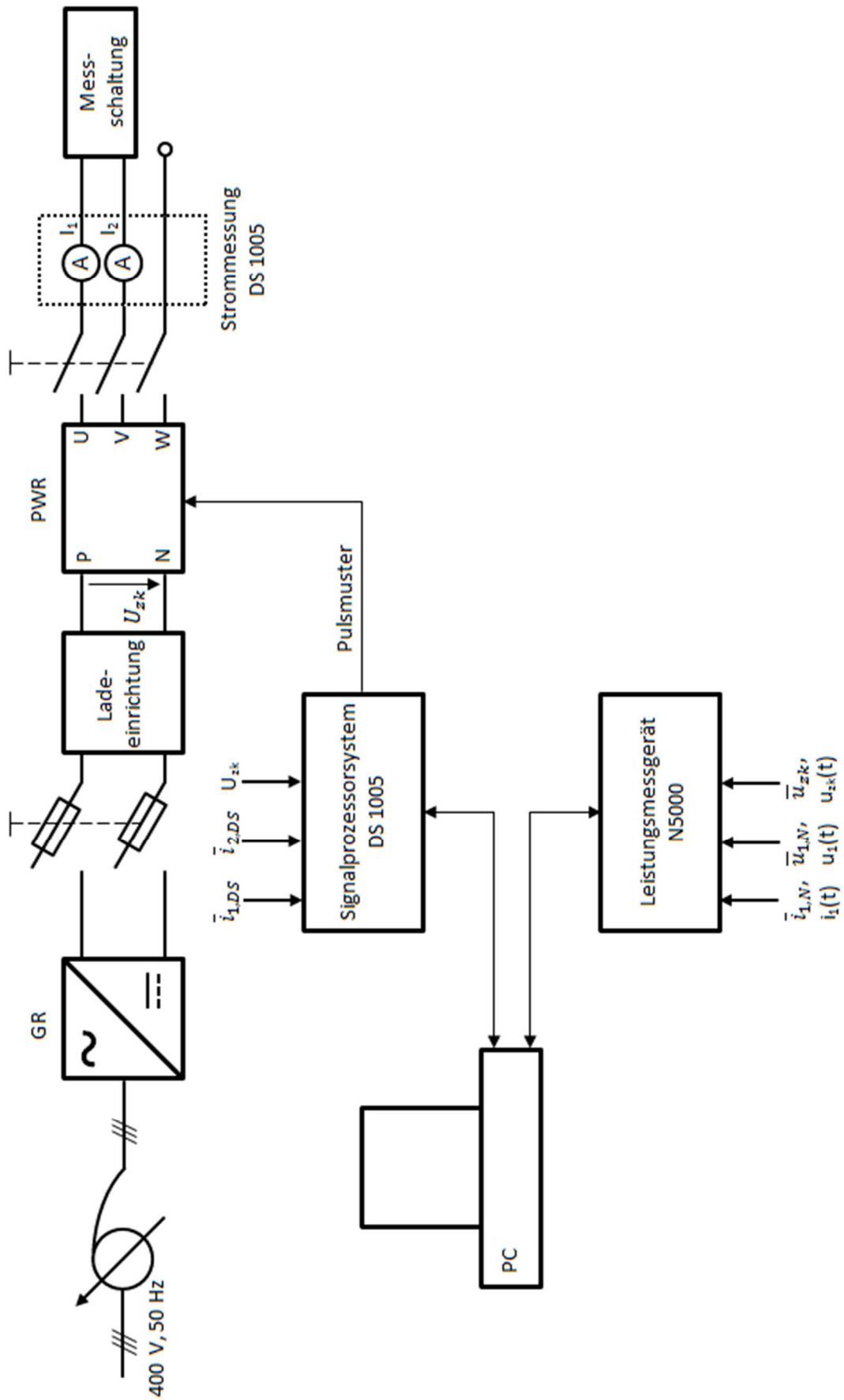


Bild 2.8: Messaufbau im Labor

2.2.1 Messungen mit dem Signalprozessorsystem DS 1005

Gemessen werden die Gleichwerte (linearen Mittelwerte) der Ströme $\bar{i}_{1,DS}$ und $\bar{i}_{2,DS}$. Die Erfassung erfolgt über einen Stromwandler. Mittels Messsignalverarbeitung wird das Messsignal für den A/D-Input des Signalprozessorsystems aufbereitet.

Die Zwischenkreisspannung wird auch gemessen und die Anpassung des Messsignals an das Signalprozessorsystem (A/D-Input) erfolgt dabei über eine „Differential Probe“.

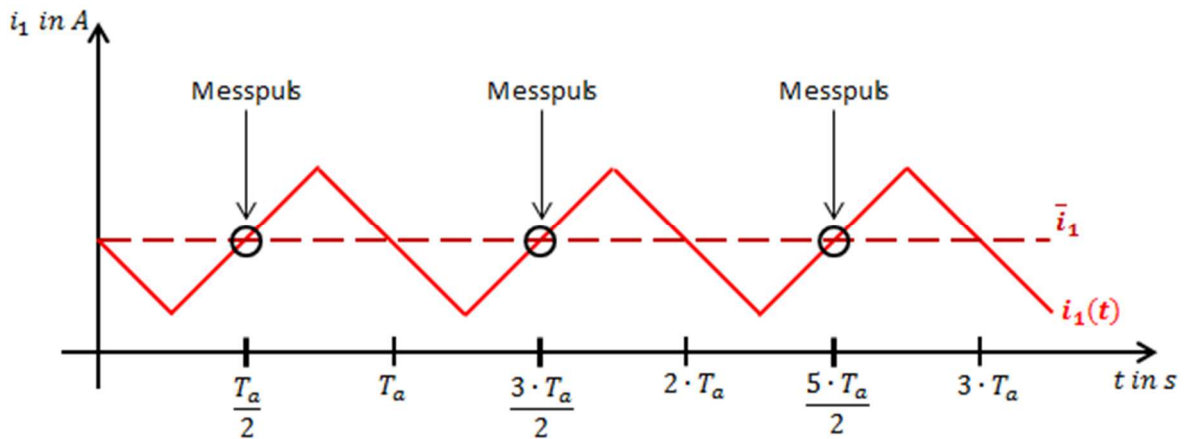


Bild 2.9: Messung des Strommittelwertes mit dem Signalprozessorsystem DS 1005

Im Bild 2.9 ist die Mittelwertmessung mit dem Signalprozessorsystem dargestellt. Daraus ist ersichtlich, dass zu bestimmten Zeitpunkten der Momentanwert des Stromes i_1 und der Mittelwert \bar{i}_1 gleich sind. Aufgrund der ohmsch-induktiven Last hat der Strom diesen Funktionsverlauf. Damit ist diese einfache Messwerterfassung überhaupt möglich. Da das Signalprozessorsystem die Pulsperiode T_a vorgibt, kann jeweils in der Mitte der Pulsperiode ein Messpuls ausgegeben werden. Dabei wird der Augenblickswert des Stromes i_1 als Mittelwert übernommen.

2.2.2 Messungen mit dem Leistungsmessgerät N5000

Zur Messung der Spannung am Ausgang des Pulswechselrichters wird dieses Messgerät benötigt. Dabei werden sowohl der Mittelwert $\bar{u}_{1,N}$ als auch der Momentanwert u_1 gemessen. Zusätzlich erfolgt die Messung des Stroms über die erste Halbbrücke. Auch hier werden der Mittelwert $\bar{i}_{1,N}$ und der Momentanwert i_1 erfasst. Für die Strommessung wird ein externer Shunt verwendet.

Auch mit diesem Messgerät wird die Zwischenkreisspannung gemessen ($u_{zk,N}$ bzw. $u_{zk}(t)$).

Im Gegensatz zum Signalprozessorsystem können mit diesem Messgerät zeitlichen Funktionsverläufe mit einem Megasample/s aufgezeichnet werden. Die Mittelwerte werden über frei wählbare Mittelungsperioden im Messgerät berechnet.

2.2.3 Messschaltung

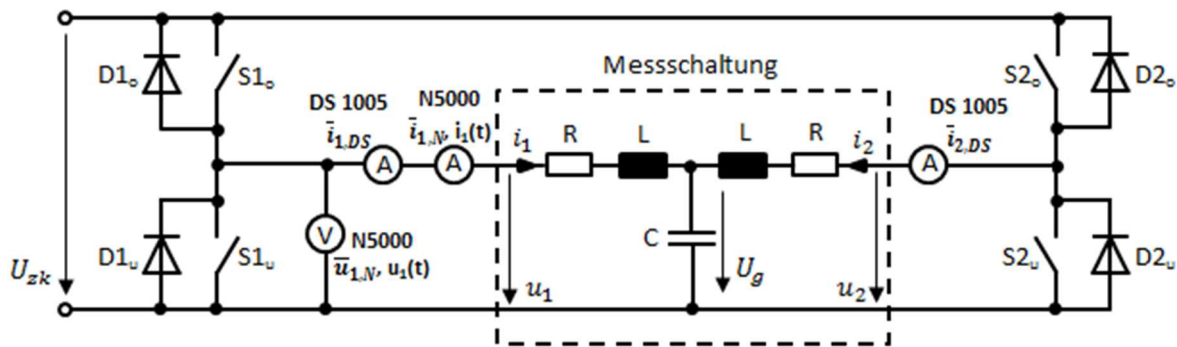


Bild 2.10: Messschaltung zur Erfassung der Umrichternichtlinearität

Die Messschaltung ist eine T-Schaltung, wobei die waagrechten Zweige aus einem ohmschen Widerstand R und einer Induktivität L bestehen (siehe Bild 2.10). Dadurch ist jede der beiden Halbbrücken des Spannungszwischenkreisumrichters mit einer ohmsch-induktiven Last beaufschlagt. Eine Leistungsdrossel ist für diese beiden Parameter verantwortlich. Der Wicklungswiderstand beträgt hierbei $R = 0,3 \Omega$. Die Induktivität der Drossel hat den Wert $L = 10 \text{ mH}$. Den senkrechten Zweig der T-Schaltung bildet ein Kondensator mit der Kapazität von $C = 314 \mu\text{F}$. Da die zweite Halbbrücke mit der Ausgangsspannung u_2 auch ein hochfrequentes PWM-Signal ist, wird mit C die Gegenspannung U_g geglättet. Für die spätere Simulation kann U_g als konstante Gleichspannungsquelle interpretiert werden.

Das Konzept für die Messung der Umrichternichtlinearität besteht darin, dass in diesem Fall die erste Halbbrücke mit einem konstanten Tastverhältnis beaufschlagt wird. Durch das Tastverhältnis d_1 ergibt sich die geforderte Ausgangsspannung des Spannungszwischenkreisumrichters $\bar{u}_1 = U_{zk} \cdot d_1$. Mit der im Bild 2.10 dargestellten Schaltung kann in der ersten Halbbrücke der Strom \bar{i}_1 mit Hilfe von d_2 der zweiten Halbbrücke eingestellt werden. Da in der Realität die gemessene Spannung am Ausgang des U-Umrichters vom Strom abhängt, fällt am Umrichter eine Spannung $\bar{u}_{1,NL}$ ab.

2.3 Messablauf

Für die Messung der Umrichternichtlinearität wird das Tastverhältnis auf $d_1 = 0,5$ festgelegt. Mit dem Tastverhältnis d_2 der zweiten Halbbrücke wird der Strom \bar{i}_1 von -20 A bis 20 A schrittweise verändert. Im Strombereich, in dem die Nichtlinearität stärker ausgeprägt ist, werden die Schritte kleiner gewählt.

Mit dem N5000 werden pro Messpunkt ca. 10 Mittelwertmessungen vom Strom \bar{i}_1 , der Spannung \bar{u}_1 und der Zwischenkreisspannung über die Dauer von $t_m = 0,2 \text{ s}$ aufgezeichnet. Für die stationären Messpunkte wurde mit dem N5000 auch der zeitliche Funktionsverlauf

von u_1 , i_1 und u_{zk} über einen Zeitspanne von 0,2 s gemessen.

Mit dem Signalprozessorsystem werden zudem die Mittelwerte des Stromes $\bar{i}_{1,DS}$ in einem File abgelegt.

Da die Zwischenkreisspannung mit steigendem Strom absinkt, wird diese bei jedem Messpunkt nachgeregelt und mit dem Stelltransformator auf den gewünschten Wert eingestellt.

Der verwendete U-Umrichter kann laut Datenblatt mit einer maximalen Zwischenkreisspannung von 900 V betrieben werden. Dabei sind Ströme bis zu maximal 300 A über die Halbbrücken möglich. Im Labor werden mit diesem Spannungszwischenkreisumrichter Maschinen im Strombereich von 20 A betrieben. Dabei ist die Verkabelung dem Amperebereich der Maschinen angepasst bzw. auf einen Maximalstrom von 30 A ausgelegt. Die Einschränkungen die sich dadurch ergeben sind relativ gering, da der Strombereich in dem die Umrichternichtlinearität am markantesten ist, gut abgedeckt ist. Die Zwischenkreisspannung wird bis 300 V eingestellt. Dies ist ausreichend um den Einfluss der Zwischenkreisspannung auf die Umrichternichtlinearität aufzuzeigen. Dabei bildet die Spannungsfestigkeit der verwendeten Bauteile (Drosseln und Kondensatoren) die Grenze.

2.4 Messergebnisse für die Umrichternichtlinearität

Aus den Messungen kann nun die Umrichternichtlinearität abgeleitet werden. Ein Tastverhältnis von $d_1 = 0,5$ ergibt mit der Zwischenkreisspannung U_{zk} eine mittlere Spannung am Ausgang des Pulswechselrichters mit $\bar{u}_{1,soll} = U_{zk} \cdot d_1$. Durch die gemessene Zwischenkreisspannung erhalten wir den tatsächlichen Wert für $\bar{u}_{1,ist}$. Der Vergleich der beiden Spannungsmittelwerte führt zu einer Differenzspannung, die einen nichtlinearen Funktionsverlauf hat.

$$\bar{u}_{1,NL} = \bar{u}_{1,soll} - \bar{u}_{1,ist} \quad (2.3)$$

Wird der berechnete Spannungsabfall an der Umrichternichtlinearität $\bar{u}_{1,NL}$ in Abhängigkeit von Strommittelwert \bar{i}_1 in einem Diagramm aufgetragen, ergeben sich die im Bild 2.11 dargestellten Kennlinien.

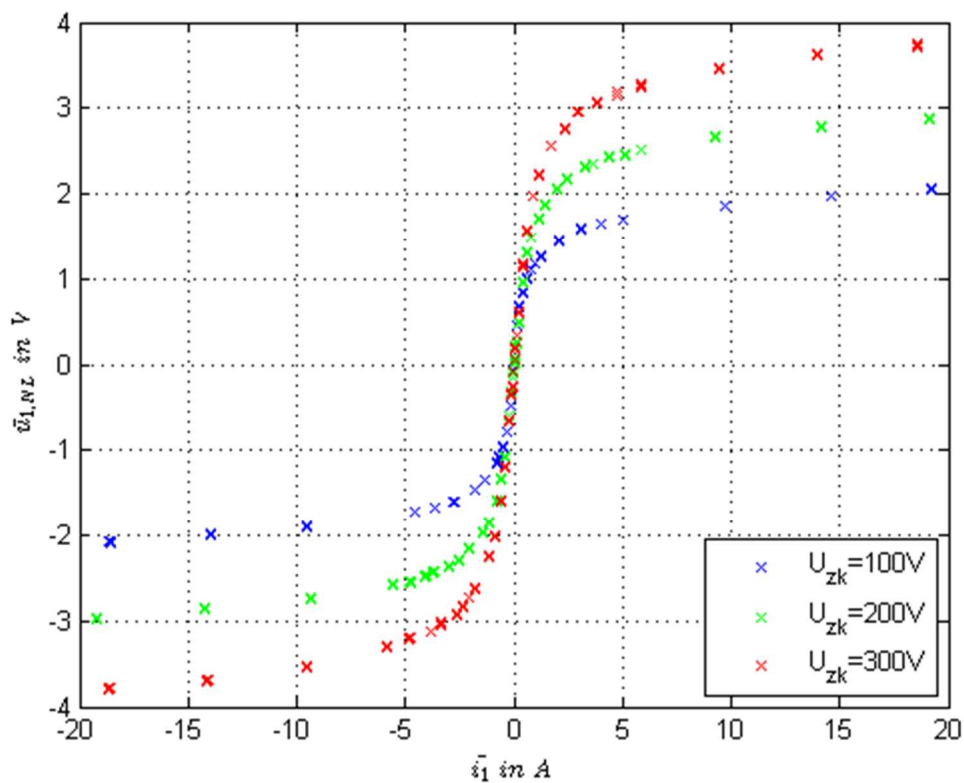


Bild 2.11: Gemessene Umrichternichtlinearität für verschiedene Zwischenkreisspannungen

In den folgenden Kapiteln wird nun ein mathematisches Modell entwickelt, um die gemessene Umrichternichtlinearität möglichst genau nachbilden zu können. Hierfür starten wir mit dem idealen Modell eines Spannungszwischenkreisumrichters. Dann wird die Berücksichtigung der Verriegelungszeit im Modell realisiert. Danach erfolgt die Modellierung der Spannungsabfälle an IGBT und Freilaufdiode. Den Abschluss bildet die Modellerweiterung mit der parasitären Kapazität.

3 Ideales Modell des Spannungszwischenkreisumrichters

In diesem Kapitel wird die Erzeugung eines idealen PWM-Signals beschrieben, das eine Halbbrücke erzeugt. Ein dreiphasiges ideales Modell wurde mir von meinem Betreuer zur Verfügung gestellt. Die Modellierung in dieser Arbeit wird jedoch einphasig realisiert. Als erstes folgt die prinzipielle Funktionserklärung einer Halbbrücke, danach die Modellrealisierung in MATLAB/Simulink. Dabei wird auch die Mittelwertberechnung im Simulink-Modell und das mathematische Modell der Last vorgestellt. Anschließend kommt ein Vergleich des Simulationsmodells mit der Messung. Wichtig für diesen ersten Schritt der Modellierung ist, dass noch keine Korrekturen für die Umrichternichtlinearität implementiert werden. Dieses Modell dient als Basis für die weiteren Modifizierungsstufen unter Berücksichtigung der Umrichternichtlinearität.

3.1 Funktionsweise einer Halbbrücke zur Erzeugung eines idealen PWM-Signals

Der Einstieg in die Modellierung der Umrichternichtlinearität bei Spannungszwischenkreisumrichtern führt über die Nachbildung einer idealen PWM in MATLAB/Simulink. Es werden nun anhand des zeitlichen Verlaufs am Ausgang der ersten Halbbrücke, die nacheinander ablaufenden Schaltzustände pro Pulsperiode erläutert. Diese Betrachtungen sind essentiell. Sie bilden die Grundlage für ein sehr einfaches Modell eines U-Umrichters.

Der im Bild 3.1 dargestellte ideale PWM-Verlauf über eine Pulsperiode wird für die folgende Untersuchung in fünf Zeitintervalle bzw. Zeitpunkte aufgeteilt.

- a) $0 \leq t < t_1$
- b) $t = t_1$
- c) $t_1 < t < t_2$
- d) $t = t_2$
- e) $t_2 < t < T_a$

Diesen fünf zeitlichen Unterteilungen werden die im Bild 3.2 bis Bild 3.6 dargestellten Schaltzustände der Halbbrücke zugeordnet. Es wird angenommen, dass die Schaltzeiten ideal sind und es zu keinen Verzögerungen beim Schalten kommt. Zusätzlich können die Schalter gleichzeitig Öffnen und Schließen, ohne dass dabei der Zwischenkreis kurzgeschlossen wird. Je nach Stromvorzeichen ergeben sich unterschiedliche Strompfade, d.h. der Strom fließt in Abhängigkeit seiner Richtung über die Freilaufdiode oder den

IGBT. Im Folgenden werden die Schaltzustände bei positivem bzw. negativem Strom diskutiert.

3.1.1 Positiver Strom

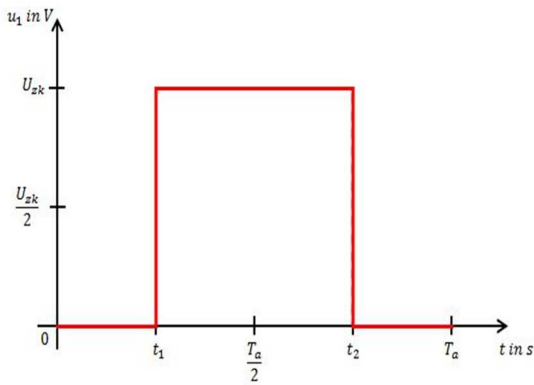


Bild 3.1: PWM ideal bei $i_1 > 0$

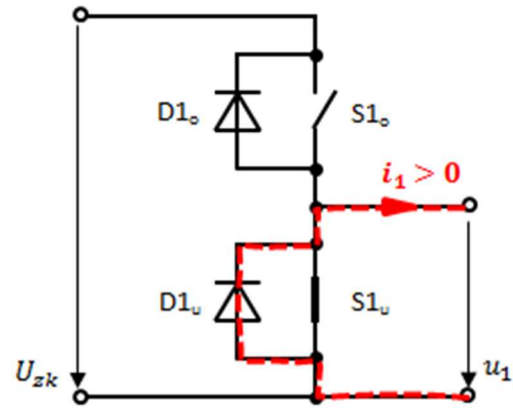


Bild 3.2: $0 \leq t < t_1$

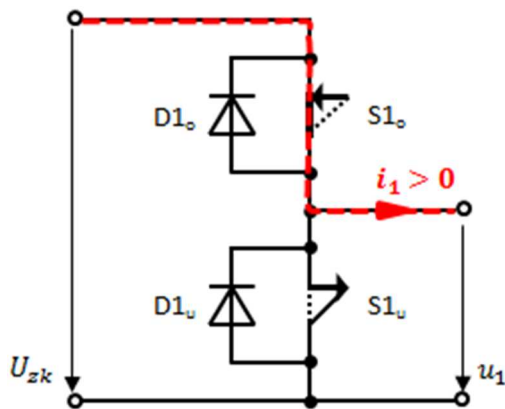


Bild 3.3: t_1

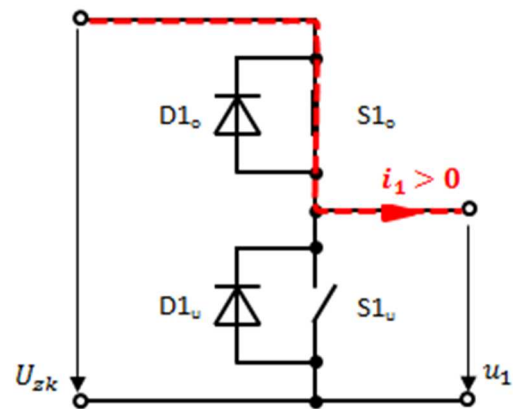


Bild 3.4: $t_1 < t < t_2$

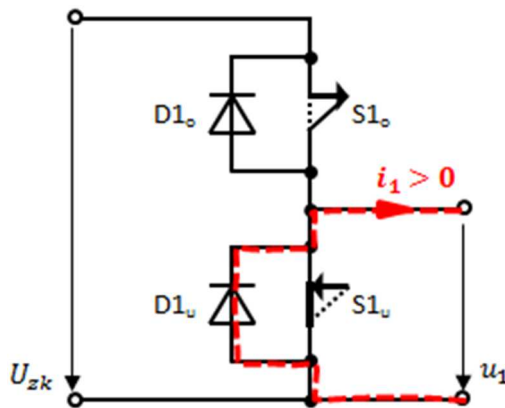


Bild 3.5: t_2

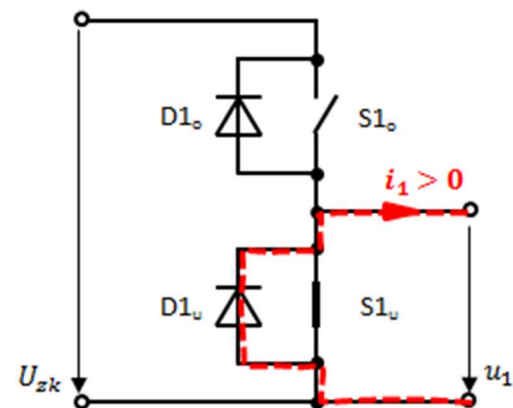


Bild 3.6: $t_2 < t < T_a$

a) Schaltzustand während dem Zeitintervall $0 \leq t < t_1$ (siehe Bild 3.2)

Schalter $S1_u$ ist geschlossen, somit ist $S1_o$ offen und der Ausgang des Spannungszwischenkreisumrichters liegt auf dem Potential 0 V. Ein Strom mit positivem Vorzeichen fließt über die untere Freilaufdiode ($D1_u$). Der IGBT führt keinen Strom, da er wie im vorigen Kapitel erklärt, nicht „rückstromfähig“ ist.

b) Schaltzustand zum Zeitpunkt t_1 (siehe Bild 3.3)

Schalter $S1_u$ öffnet sich, gleichzeitig schließt sich $S1_o$ ohne dabei die Halbbrücke kurzzuschließen. Das Ausgangspotential des Pulswechselrichters springt von 0 V auf den Wert der Zwischenkreisspannung. Ein positiver Strom wechselt seinen Pfad von der unteren Freilaufdiode ($D1_u$) zum oberen IGBT ($S1_o$).

c) Schaltzustand während dem Zeitintervall $t_1 < t < t_2$ (siehe Bild 3.4)

Der Schalter $S1_o$ ist geschlossen, $S1_u$ ist somit offen. Am Ausgang des Umrichters liegt nun die Spannung U_{zk} an. Der Strom mit positivem Vorzeichen fließt über den oberen IGBT ($S1_o$).

d) Schaltzustand zum Zeitpunkt t_2 (siehe Bild 3.5)

Dieser Fall ist ähnlich zum Schaltzustand b). Schalter $S1_o$ öffnet sich, im selben Augenblick schließt $S1_u$. In diesem Fall wird wieder angenommen, dass die Halbbrücke dabei nicht kurzgeschlossen wird. Damit springt das Potential am Ausgang der Halbbrücke von U_{zk} zurück auf 0 V. Der positive Strom wechselt vom oberen IGBT ($S1_o$) zur Freilaufdiode unten ($D1_u$).

e) Schaltzustand während dem Zeitintervall $t_2 < t < T_a$ (siehe Bild 3.6)

Dieser Schaltzustand ist äquivalent zum Zeitintervall in Punkt a). Beim Durchlauf der nachfolgenden Pulsperioden werden die obigen Schritte erneut durchlaufen.

3.1.2 Negativer Strom

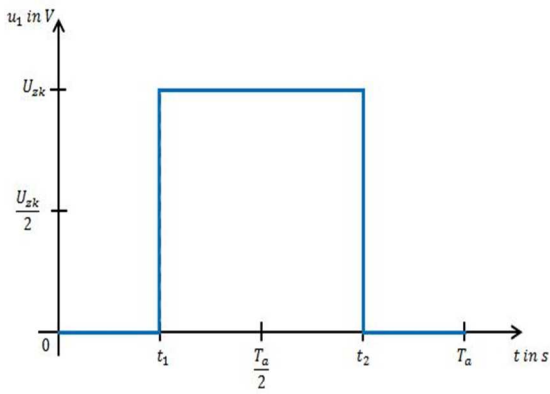


Bild 3.7: PWM ideal bei $i_1 < 0$

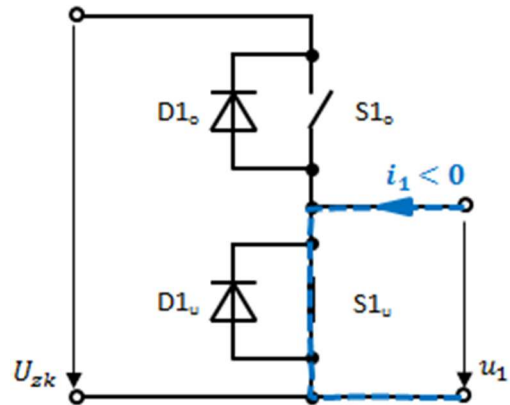


Bild 3.8: $0 \leq t < t_1$

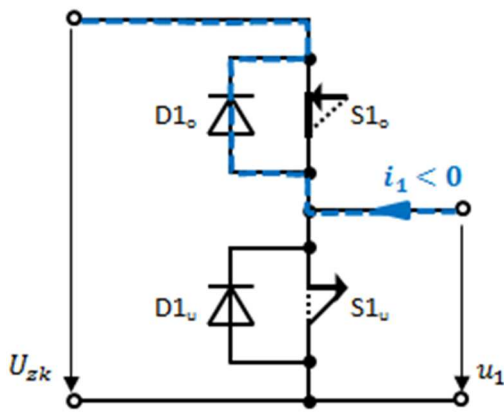


Bild 3.9: t_1

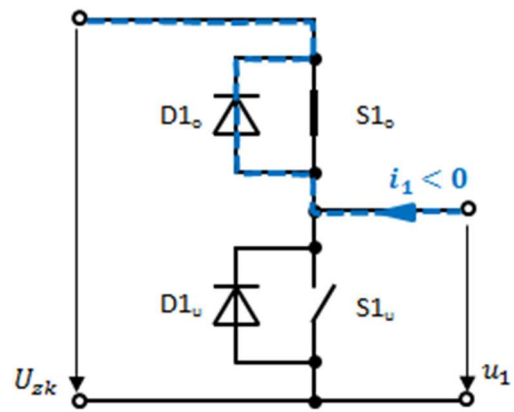


Bild 3.10: $t_1 < t < t_2$

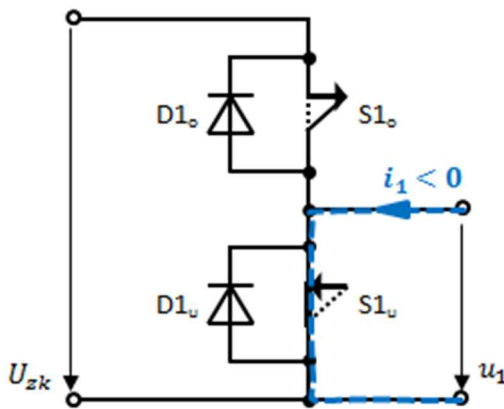


Bild 3.11: t_2

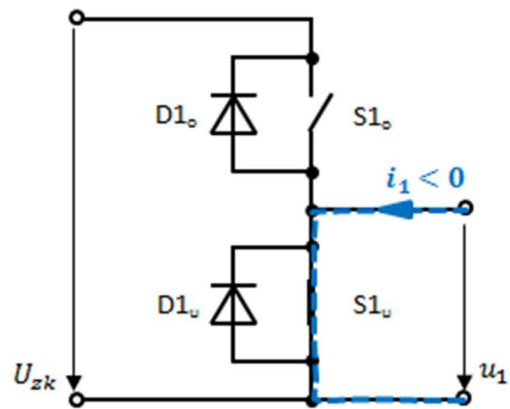


Bild 3.12: $t_2 < t < T_a$

- a) Schaltzustand während dem Zeitintervall $0 \leq t < t_1$ (siehe Bild 3.8)

Schalter $S1_u$ ist geschlossen, somit ist $S1_o$ offen und der Ausgang des Spannungszwischenkreisumrichters liegt auf dem Potential 0 V. Ein Strom mit negativem Vorzeichen fließt über den unteren IGBT ($S1_u$).

- b) Schaltzustand zum Zeitpunkt t_1 (siehe Bild 3.9)

Schalter $S1_u$ öffnet und gleichzeitig schließt sich $S1_o$, ohne dabei die Halbbrücke kurzzuschließen. Das Ausgangspotential des Pulswechselrichters springt von 0 V auf den Wert der Zwischenkreisspannung. Ein negativer Strom wechselt seinen Pfad vom unteren IGBT ($S1_u$) zur oberen Freilaufdiode ($D1_o$).

- c) Schaltzustand während dem Zeitintervall $t_1 < t < t_2$ (siehe Bild 3.10)

Der Schalter $S1_o$ ist geschlossen und damit ist $S1_u$ offen. Am Ausgang des Umrichters liegt nun die Spannung U_{zk} an. Der Strom mit negativem Vorzeichen fließt über die obere Freilaufdiode ($D1_o$).

- d) Schaltzustand zum Zeitpunkt t_2 (siehe Bild 3.11)

Dieser Fall ist ähnlich zum Schaltzustand b) im Bild 3.9. Schalter $S1_o$ öffnet sich, im selben Augenblick schließt $S1_u$. Es wird angenommen, dass dabei kein Kurzschluss über die Halbbrücke entsteht. Damit springt das Potential am Ausgang der Halbbrücke von U_{zk} zurück auf 0 V. Der negative Strom wechselt von der oberen Freilaufdiode ($D1_o$) zum IGBT unten ($S1_u$).

- e) Schaltzustand während dem Zeitintervall $t_2 < t < T_a$ (siehe Bild 3.12)

Dieser Schaltzustand ist äquivalent zum Zeitintervall in a).

3.1.3 Diskussion zur Abhängigkeit vom Stromvorzeichen

Aus Bild 3.1 und Bild 3.7 ist ersichtlich, dass die beiden Pulse identisch sind. Somit ist das ideale PWM-Signal unabhängig vom Vorzeichen des Stromes. Im Bild 3.13 ist der Unterschied beim Stromvorzeichen veranschaulicht. Je nach Stromrichtung, fließt der Strom über bestimmte Pfade bzw. leistungselektronische Bauteile.

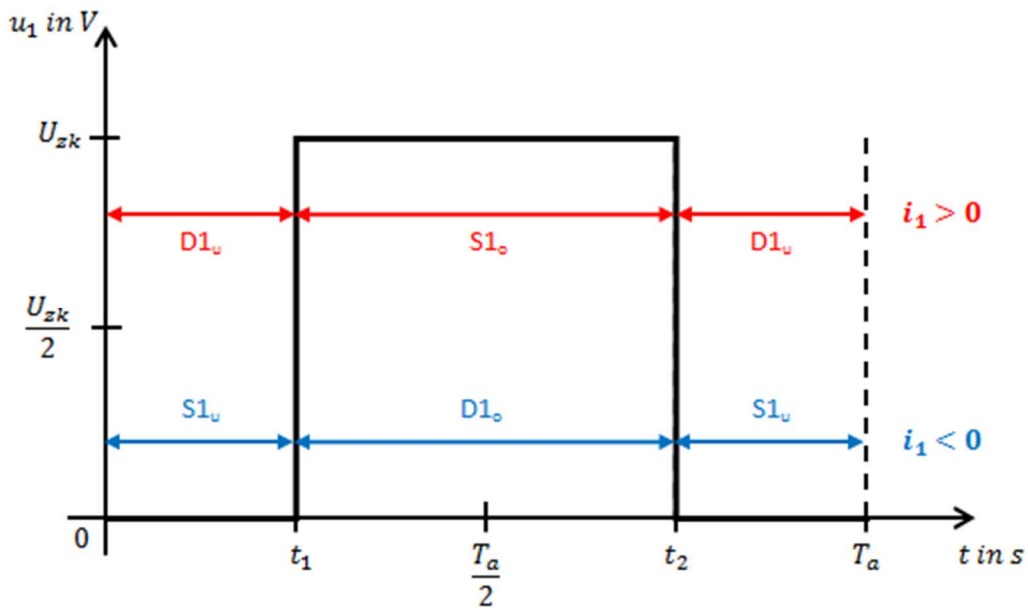


Bild 3.13: Stromführung abhängig von Stromvorzeichen

Nun ist klar wie der Spannungszwischenkreisumrichter prinzipiell funktioniert. Im nächsten Abschnitt wird die vorhandene Implementierung des vorhin veranschaulichten Konzepts der Erzeugung eines idealen PWM-Signals mittels MATLAB/Simulink erläutert.

3.2 Modellierung eines idealen PWM-Signals

Nun wird ein idealer Spannungszwischenkreisumrichter nachgebildet und ein MATLAB/Simulink-Modell vorgestellt, das diesen Spannungsverlauf erzeugt (siehe Bild 3.14). Dabei handelt es sich um ein PWM-Signal mit dem konstanten Tastverhältnis von $d_1 = 0,5$. Damit erhält man mit einer Zwischenkreisspannung von $U_{zk} = 100 \text{ V}$ einen Spannungsmittelwert von $\bar{u}_1 = 50 \text{ V}$ über eine Pulsperiode T_a .

Es sind mehrere Möglichkeiten vorhanden dies zu realisieren. Die einfachste Lösung wäre eine vorhandene Realisierung aus der Simulink-Library. Dies sind z.B. „Pulse Generator“, „Signal-Builder“, „Step“ und „Signal-Generator“. Der Nachteil dieser Blöcke besteht darin, dass sie für variable Tastverhältnisse ungeeignet sind. Die ideale Realisierung geht zwar einfach und schnell, aber die geplante Erweiterung für die Berücksichtigung der Verriegelungszeit ist kompliziert und unpraktisch. Aus diesem Grund wird das Modell mit einer sogenannten S-Function implementiert.

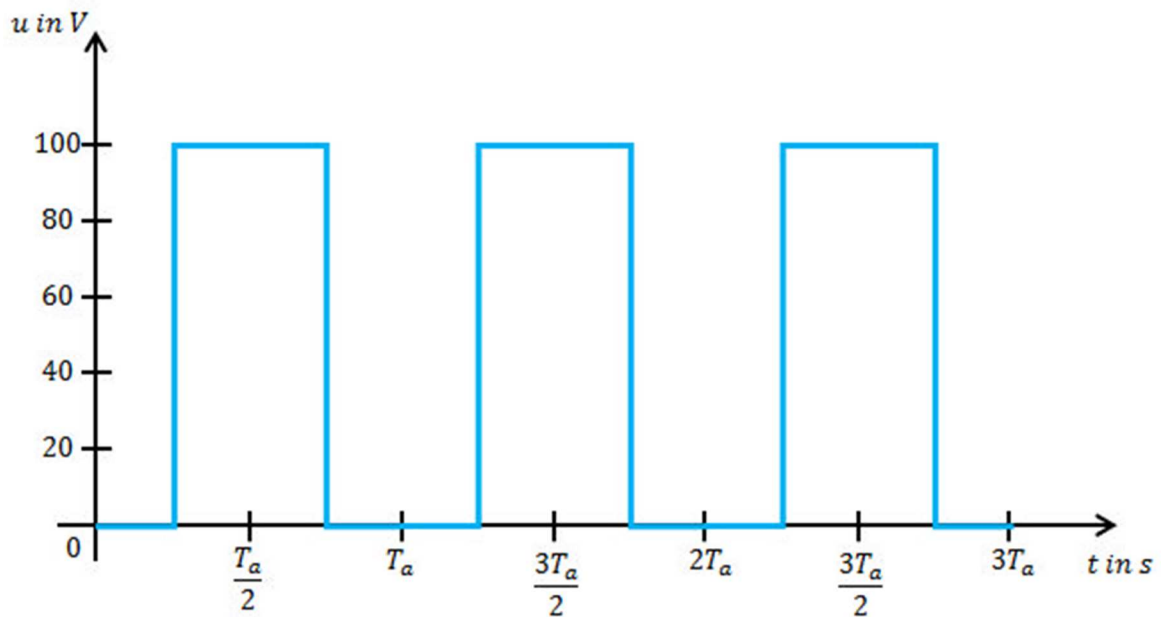


Bild 3.14: Ideales PWM-Signal mit drei Pulsperioden

3.2.1 Implementierung mit S-Function

Die vorhandene Implementierung wurde mit einer „S-Function“ aus der Library Simulink/User-Defined Functions programmiert. Der Begriff S-Function steht als Abkürzung für „system-functions“. Der Vorteil dabei ist, dass ein Simulink-Block gezielt nach den benötigten Funktionen mit einer m-Funktion (MATLAB-Code) programmiert werden kann. Dadurch wird der Eingriff auf den Blockausgang der S-Function zu einem gewünschten Zeitpunkt ermöglicht. Bei Verwendung eines „Variable Step Solvers“ kann zu gewünschten Zeitpunkten ein Berechnungsschritt durchgeführt werden.

Für das gewünschte Signal aus Bild 3.14 kann in Abhängigkeit vom Tastverhältnis die Spannung auf 0 V bzw. auf $U_{zk} = 100$ V gesetzt werden. Für eine detailliertere Behandlung der S-Function verweise ich auf die MATLAB-Hilfe in [10] und die zugehörigen Demos.

Um den Kern der Modellierung, die S-Function mit der m-Function „varhit2ideal.m“, wurde ein Koppelplan aufgebaut, um den im Bild 3.14 dargestellten Funktionsverlauf zu erzeugen.

Koppelplan des MATLAB/Simulink-Modells

Im Bild 3.15 ist der Simulink-Koppelplan abgebildet. Anhand dieser Grafik wird die Erzeugung einer idealen PWM erklärt.

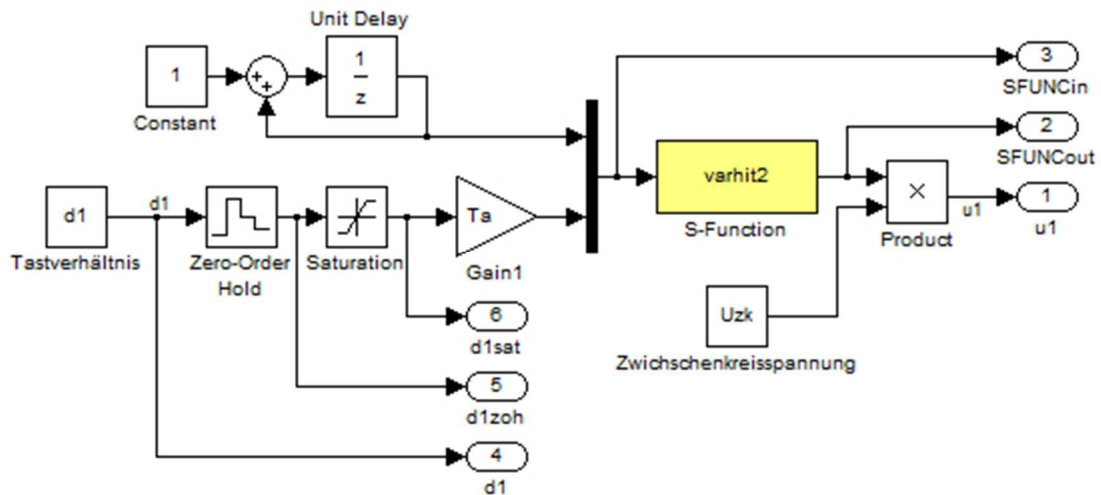


Bild 3.15: Simulink-Koppelplan für ideale PWM Erzeugung

Als erstes werden die beiden Eingangssignale für die S-Function betrachtet. Danach folgt die S-Function mit „varhit2ideal.m“ und das damit erhaltene Ausgangssignal der S-Function. Die Parameterwerte $T_a = 200 \mu\text{s}$ und $U_{zk} = 100 \text{ V}$ wurden für die beiden Testversuche festgelegt.

3.2.2 Eingangs- bzw. Ausgangssignale der S-Function

Das erste Eingangssignal ist ein Zähler. Der Unit Delay-Block hat einen Anfangswert von 0. Die Sample Time ist eine Pulsperiode T_a , d.h. er verzögert sein Eingangssignal um eine Pulsperiode. Zu Beginn der zweiten Pulsperiode springt der Wert am Ausgang des Unit Delay-Blocks auf 1 und nach jeder weiteren Pulsperiode erhöht sich dieses Signal um 1.

Das zweite Eingangssignal für die S-Function liefert die Information, wie lange U_{zk} eingeschaltet sein soll. Dies wird über das sogenannte Tastverhältnis d_1 vorgegeben. Es ist notwendig, das Tastverhältnis im Modell zu begrenzen und für die gesamte Pulsperiode konstant zu halten. Dies geschieht wie im Bild 3.15 dargestellt mit einem Saturation-Block und dem Zero-Order Hold-Block. Durch die Multiplikation mit der Pulsperiode T_a erhält man die Einschaltzeit t_{ein} (siehe Gleichung (2.1)).

Das Ausgangssignal ist ein Pulssignal mit den Werten 0 und 1. Dieses Signal multipliziert mit der Zwischenkreisspannung ergibt das gewünschte PWM-Signal.

3.2.3 Simulationsablauf der S-Function

In diesem Abschnitt wird die Verarbeitung der vorhin behandelten Eingangssignale der S-Function erläutert bzw. die wichtigsten Eigenschaften im Simulationsablauf. Für die Erklärung wird das Testexperiment mit dem konstanten Tastverhältnis $d_1 = 0,5$ für die Dauer von drei Pulsperioden verwendet.

Der Programmcode in der S-Function ist eine MATLAB-Funktion und besitzt folgende Deklaration:

```
function [sys,x0,str,ts] = varhit2ideal(t,x,u,flag,Ta)
```

Inputs:

- t...Simulationszeitpunkt
- x...Zustandsvektor für kontinuierliche und diskrete Zustände
- u...Eingangsvektor
- flag... Variable für die switch-Anweisung
- Pulsperiode T_a ...S-function parameter

Über die Dialogbox „Function Block Parameters: S-Function“ wird die Pulsperiode T_a als zusätzlicher Parameter definiert.

Outputs:

- sys...Rückgabewerte der jeweiligen Fälle aus dem switch-Konstrukt
- x0...Anfangszustand
- str...generell leere Matrix
- ts...Sample Time Information

Fallunterscheidung der switch-Anweisung

- case 0: Initialization
- case 1: Derivative (wird nicht benötigt)
- case 2: Update
- case 3: Outputs
- case 4: GetTimeOfNextVarHit
- case 9: Terminate

Während der Simulation werden die Fälle folgendermaßen durchlaufen. Nach dem Start der Simulation erfolgt die Initialisierung. Danach werden die drei Fälle GetTimeOfNextVarHit, Outputs und Update solange sequentiell durchlaufen bis $flag = 9$ die Simulation terminiert. Es wird nun erklärt was in den jeweiligen Fällen prinzipiell passiert.

Initialisierung

In der Initialisierung werden im Rückgabeparameter *sys* die Dimension des Zustandsvektors und die Anzahl der Ein- und Ausgänge definiert. Das hier beschriebene Modell hat einen diskreten Zustand, besitzt zwei Eingänge und einen Ausgang. Der Anfangszustand wurde auf $x_0 = -1$ festgelegt. Die Sample Time Information $ts = [-2 \ 0]$. Damit wird eine variable Schrittweite ermöglicht, wobei *flag* = 4 verwendet wird um den nächsten Berechnungszeitpunkt zu fixieren.

GetTimeOfNextVarHit

Nach der Initialisierung ist die Variable *flag* = 4 und es wird auf *GetTimeOfNextVarHit* geschaltet. In diesem Fall wird nun der nächste Berechnungszeitpunkt an dem die S-Function wieder aufgerufen wird bestimmt und im Rückgabeparameter *sys* abgelegt.

Outputs

Nun ist *flag* = 3 und es wird der Fall *Outputs* ausgeführt. Dabei wird für den momentanen Zeitpunkt der Ausgang auf 0 oder 1 gesetzt und im Rückgabeparameter *sys* gespeichert.

Update

Nach der Bestimmung des Ausgangswertes für die S-Function ist *flag* = 2. Im *Update* wird nun der diskrete Zustand neu berechnet. Dieser wird benötigt, um festzustellen, in welcher Pulsperiode man sich gerade befindet. Er wird für die Berechnung des nächsten Zeitpunktes und dessen *Outputs* verwendet.

Terminate

Wie oben schon erwähnt, wird bei *flag* = 9 die Simulation beendet.

Für eine detailliertere Beschreibung der Abläufe wird auf den Quellcode im Anhang A.1 bzw. [10] verwiesen.

3.2.4 Diskussion der Simulationsergebnisse

In Tabelle 1 sind die für das Verständnis wichtigen Inputs, Outputs und Zustandsgrößen für einen Testversuch mit konstantem Tastverhältnis zusammengefasst. Bei den Zahlenwerten die in der Tabelle angegeben sind, handelt es sich um die Werte nach der Ausführung des jeweiligen Schrittes. In der Spalte für den zweiten Eingang u_2 ändert sich der Wert nie, da immer dasselbe Tastverhältnis $d_1 = 0,5$ anliegt. Für jeden Zeitpunkt an dem die S-Function aufgerufen wird, werden die Fälle *NextHit* (abgekürzt für

GetTimeOfNextVarHit), Output und Update durchlaufen. Zu erkennen ist, dass die *sys*-Werte von NextHit und Update die Werte *t* und *x* im nächsten Zeitpunkt sind.

Tabelle 1: Simulationsablauf für Testexperiment mit konstantem Tastverhältnis

		t	x	u1	u2	sys	flag
	<i>Initialization</i>	0	-1	0	0.0001		4
1. Pulsperiode	NextHit	0	-1	0	0.0001	0.00005	3
	Output	0	-1	0	0.0001	0	2
	Update	0	-1	0	0.0001	0	4
	NextHit	0.00005	0	0	0.0001	0.00015	3
	Output	0.00005	0	0	0.0001	1	2
	Update	0.00005	0	0	0.0001	0	4
	NextHit	0.00015	0	0	0.0001	0.0002	3
	Output	0.00015	0	0	0.0001	0	2
	Update	0.00015	0	0	0.0001	0	4
2. Pulsperiode	NextHit	0.0002	0	1	0.0001	0.00025	3
	Output	0.0002	0	1	0.0001	0	2
	Update	0.0002	0	1	0.0001	1	4
	NextHit	0.00025	1	1	0.0001	0.00035	3
	Output	0.00025	1	1	0.0001	1	2
	Update	0.00025	1	1	0.0001	1	4
	NextHit	0.00035	1	1	0.0001	0.0004	3
	Output	0.00035	1	1	0.0001	0	2
	Update	0.00035	1	1	0.0001	1	4
3. Pulsperiode	NextHit	0.0004	1	2	0.0001	0.00045	3
	Output	0.0004	1	2	0.0001	0	2
	Update	0.0004	1	2	0.0001	2	4
	NextHit	0.00045	2	2	0.0001	0.00055	3
	Output	0.00045	2	2	0.0001	1	2
	Update	0.00045	2	2	0.0001	2	4
	NextHit	0.00055	2	2	0.0001	0.0006	3
	Output	0.00055	2	2	0.0001	0	2
	Update	0.00055	2	2	0.0001	2	4
...	NextHit	0.0006	2	3	0.0001	0.00065	3
	Output	0.0006	2	3	0.0001	0	2
	Update	0.0006	2	3	0.0001	3	9
	<i>Terminate</i>						

Die fett geschriebenen Einträge der Zeitspalte *t* sowie in der Spalte mit *sys* (Output), ergeben den im Bild 3.16 dargestellten Funktionsverlauf für die Spannung. Dabei ist zu beachten, dass der Output in der Variable *sys* noch mit der Zwischenkreisspannung U_{zk} zu multiplizieren ist. Der gewünschten Verlauf aus Bild 3.14 kann mit diesem Modell realisiert werden.

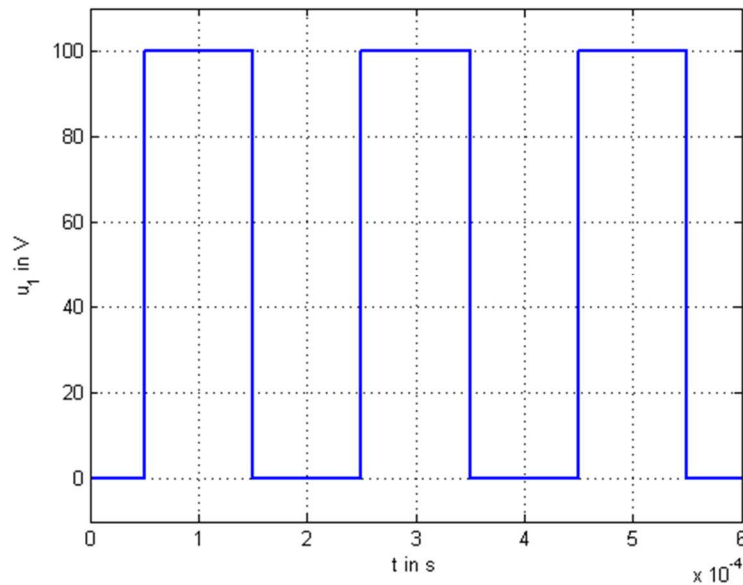


Bild 3.16: Ausgangssignal des U-Umrichters für konstantes d_1 mit drei T_a

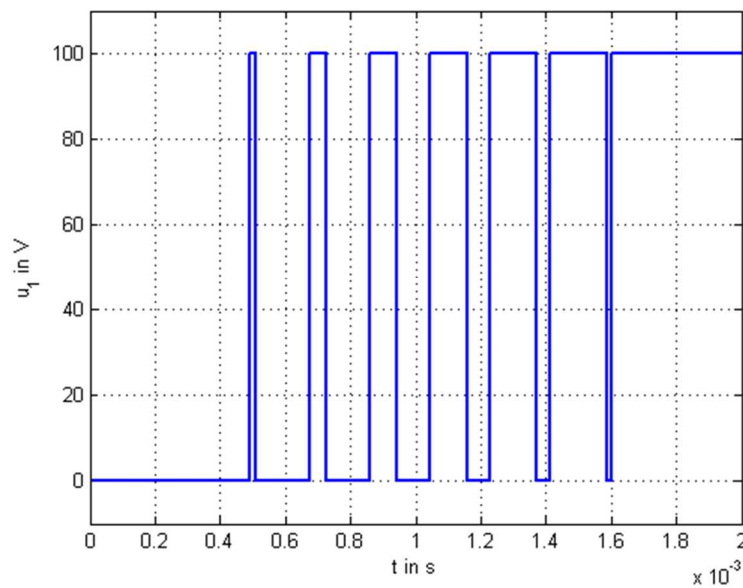


Bild 3.17: Ausgangssignal U-Umrichters für variables d_1 mit zehn T_a

Für das Testsignal mit variabler Tastrate d_1 ist der Ausgang im Bild 3.17 abgebildet. In den ersten beiden Pulsperioden bleibt die Spannung auf 0 V. Die darauffolgenden Pulse werden immer breiter und für die letzten beiden Pulsperioden bleibt der Ausgang auf $U_{zk} = 100$ V.

Mit der vorhin durchgeführten Untersuchung wurde ein Einblick in die prinzipielle Ausführung der Simulation mit einer S-Function am Beispiel der idealen PWM gezeigt. Im nächsten Abschnitt wird die Berechnung des Mittelwerts im Simulationsmodell vorgestellt.

3.3 Berechnung des Mittelwerts (Gleichwert)

Die Berechnung des Mittelwertes (Gleichwert) ist notwendig, da der Vergleich Simulation mit der Mittelwertmessung, einfach und anschaulich, den Einfluss der Umrichternichtlinearität zeigt.

Prinzipiell gibt es zwei Möglichkeiten aus den Zeitverläufen der Simulation den linearen Mittelwert zu berechnen. Die erste Möglichkeit bietet die mean-Funktion. Dabei wird nach dem Simulationsdurchlauf, mit Hilfe der erhaltenen Daten, die mean-Funktion angewendet und der Mittelwert über eine Pulsperiode bzw. mehrere Pulsperioden berechnet. Bei der zweiten Möglichkeit erfolgt die Berechnung des Mittelwertes während der Simulation. Dabei kann auch die Anzahl der Pulsperioden gewählt werden, über die gemittelt werden soll.

3.3.1 mean-Funktion

Die mean-Funktion funktioniert einwandfrei, wenn die Simulationsergebnisse äquidistante Stützstellen haben. Dies ist bei Verwendung eines Fixed-Step-Solvers gewährleistet. Hier wird aber wie vorher schon erwähnt, ein Variable-Step-Solver verwendet. Dieser liefert im Allgemeinen keine Daten mit konstanten Zeitintervallen zwischen den Berechnungsstellen. Die Ausnahme ist, wenn die maximale Schrittweite sehr klein gewählt wird und damit ein quasi-äquidistantes Signal erzeugt wird. Da eine kleine maximale Schrittweite lange Simulationszeiten und große Datenmengen zu Folge hat, wird die Möglichkeit der mean-Funktion nicht weiter verfolgt.

3.3.2 MATLAB/Simulink-Modell

Für die Mittelwertberechnung wurde ein eigenes Simulink-Modell aufgebaut, dessen Koppelplan im Bild 3.18 dargestellt ist. Damit wird eine Mittelwertbestimmung über die Pulsperiode T_a , sowie eine Mittelungsdauer t_m über mehrere Pulsperioden ermöglicht. Dieser Algorithmus wird mit

$$\bar{u} = \frac{1}{t_m} \int_{t_m} u(t) dt \quad (3.1)$$

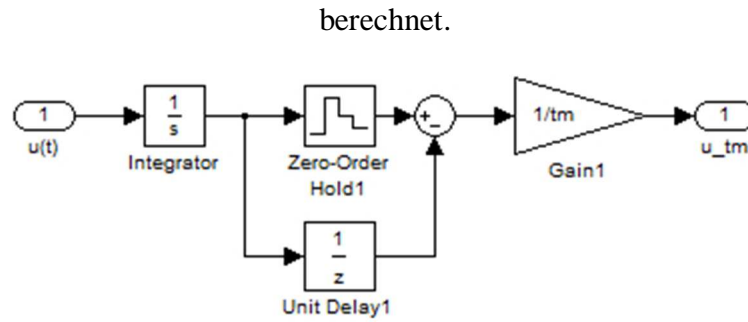


Bild 3.18: Simulink-Koppelplan für Mittelwertmessung (N5000)

Modellbeschreibung

Für das zu mittelnde Signal folgt als erstes die Integration, d.h. die Berechnung der Fläche unter dem jeweiligen Funktionsverlauf. Nun wird einerseits mit einem Zero-Order Hold-Block der Integratorausgang beim Simulationsstart und zu den darauffolgenden Mittelungsperioden abgetastet, andererseits verzögert der Unit Delay-Block den Ausgang des Integrators um eine Mittelungsdauer. Anschließend wird der Integratorausgang der vorhergehenden Mittelungsperiode von dem der momentanen subtrahiert. Durch die darauffolgende Division mit der Mittelungsdauer ergibt sich der Mittelwert. Diese Methode berechnet somit den Gleichwert bzw. den linearen (arithmetischer) Mittelwert.

3.3.3 Validierung der Mittelwertmessung

Die beiden Testexperimente mit konstantem und variablem Tastverhältnis dienen zur Veranschaulichung und Überprüfung der Funktionstüchtigkeit. Das MATLAB/Simulink-Modell zur Validierung ist im Bild 3.19 abgebildet. Darin werden die speziell für die Mittelwertberechnung interessanten Signale untersucht.

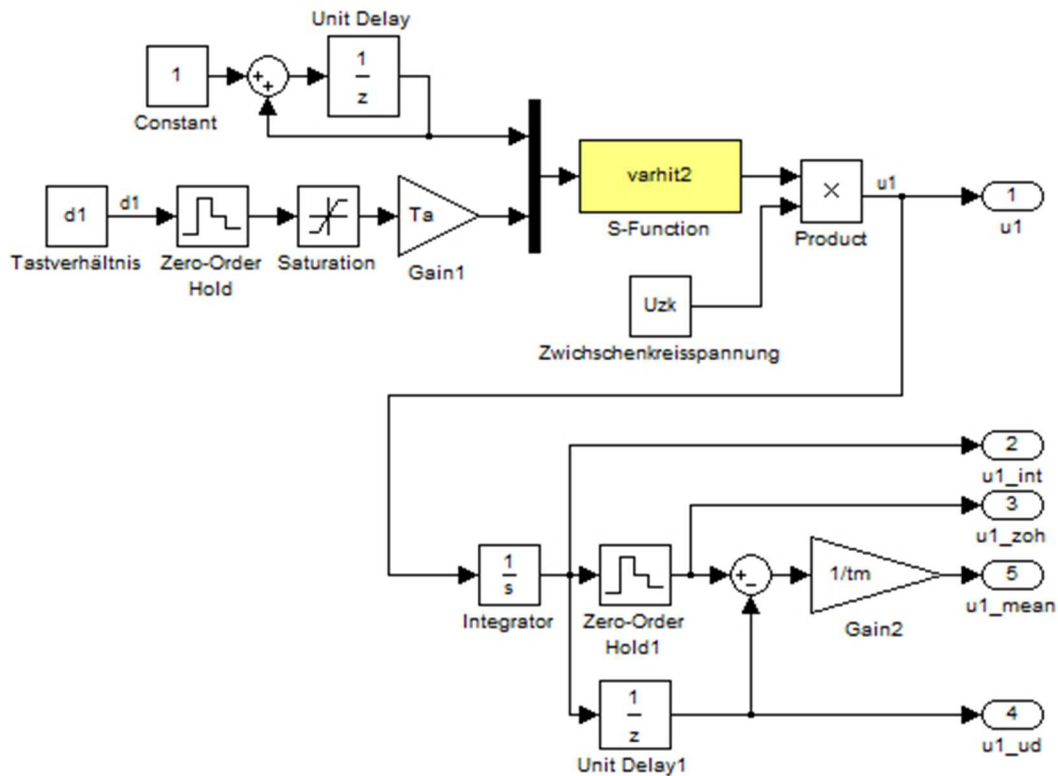


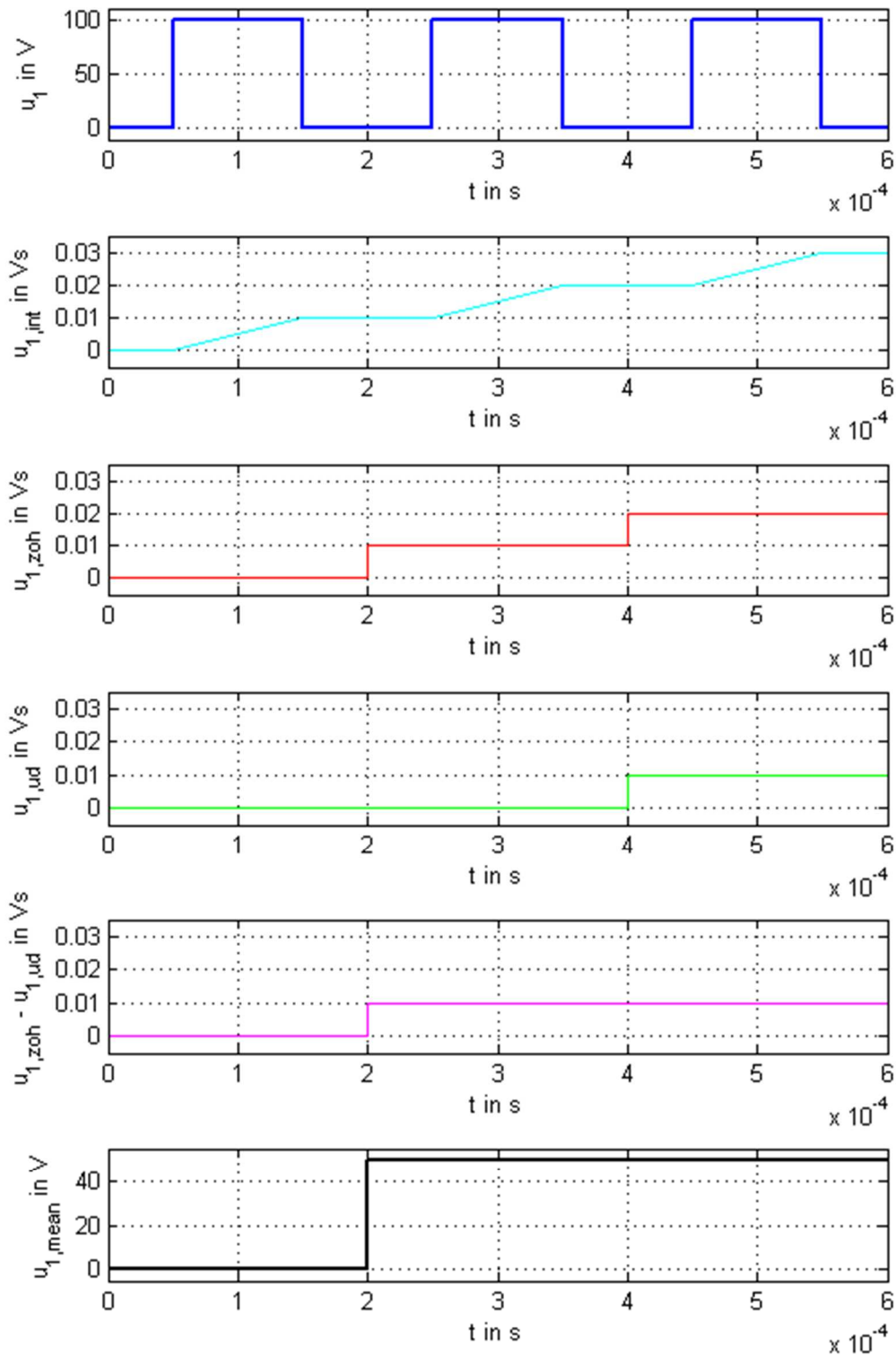
Bild 3.19: Simulink-Koppelplan zur Validierung der Mittelwertmessung

Validierung mit konstantem Tastverhältnis

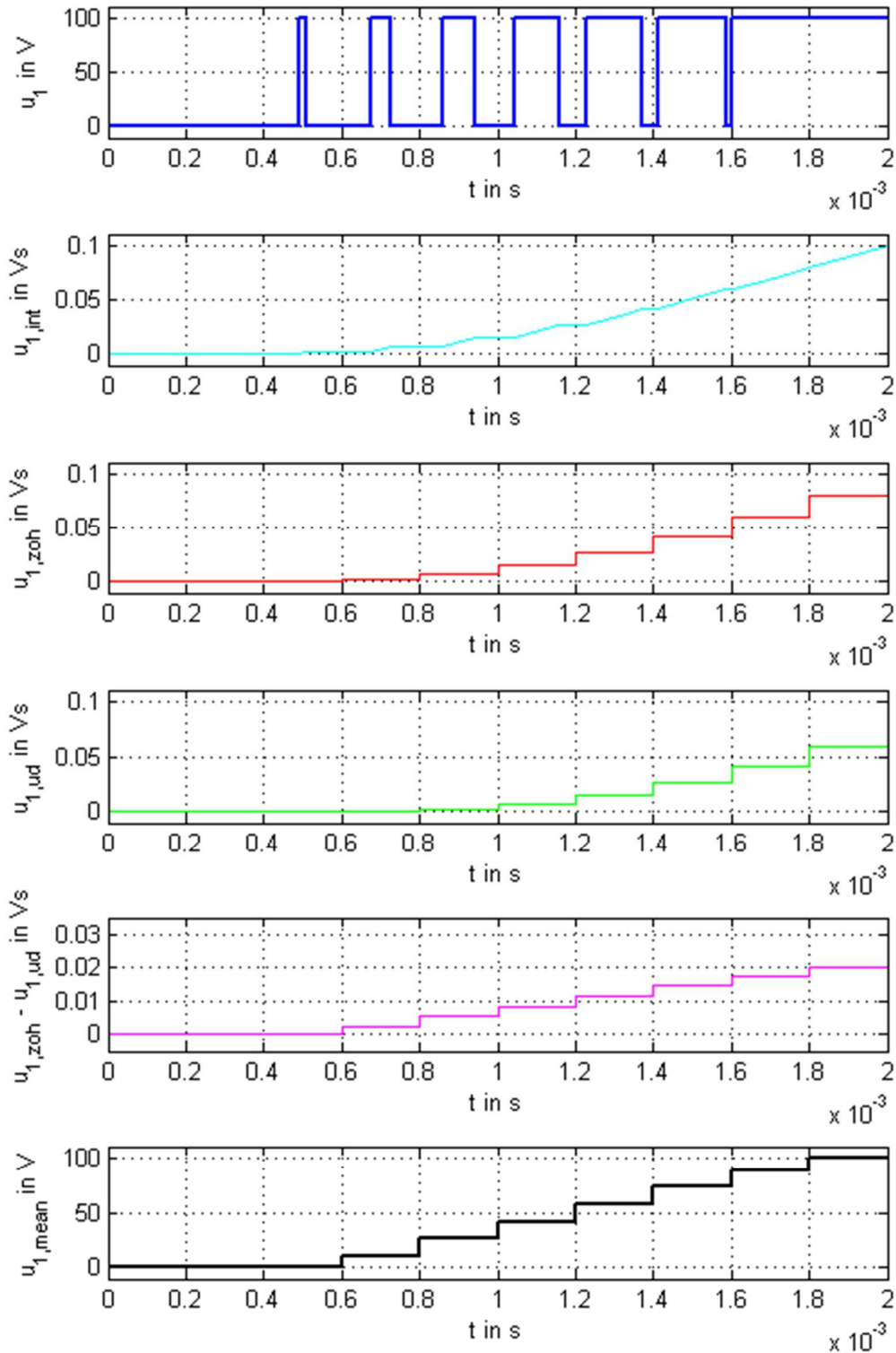
Im Bild 3.20 sind die Signale für die Berechnung des arithmetischen Mittelwertes dargestellt. Ausgehend vom PWM-Signal u_1 , folgt der Signalverlauf nach dem Integrator $u_{1,int}$. Dabei wird die Fläche unter dem Rechtecksignal u_1 berechnet. Dies ergibt für die erste Pulsperiode $0,01 \text{ Vs}$ und lässt sich in diesem Fall mit

$$U_{zk} \cdot d_1 \cdot T_a = 100V \cdot 0,5 \cdot 0,0002s = 0,01Vs \quad (3.2)$$

einfach nachrechnen. Nach jeder weiteren Pulsperiode erhöht sich der Wert um $0,01 \text{ Vs}$, da das Tastverhältnis d_1 konstant ist. Der Integratorausgang wird nun mit $t_m = T_a$ abgetastet (Zero-Order Hold) und ergibt den Funktionsverlauf $u_{1,zoh}$. Das Ausgangssignal des Integrators wird auch um die gewünschte Mittelungsdauer $t_m = T_a$ verzögert (Unit Delay) und ergibt das Signal $u_{1,ud}$. Im nächsten Subplot ist $u_{1,zoh} - u_{1,ud}$ abgebildet. Dieses Signal, dividiert durch die Mittelungsdauer $t_m = T_a$, ergibt den Mittelwert $u_{1,mean} = 50 \text{ V}$. Dabei ist ersichtlich, dass der Mittelwert um $t_m = T_a$ verzögert ausgegeben wird und erst am Ende der gewählten Mittelungsdauer zur Verfügung steht.

Bild 3.20: Mittelwertberechnung mit konstantem Tastverhältnis d_1

Validierung mit variabler Tastrate

Bild 3.21: Mittelwertberechnung mit variablem Tastverhältnis d_1

Für die Simulation mit variabler Tastrate sind die Ergebnisse im Bild 3.21 abgebildet. Im Vergleich mit dem konstanten Tastverhältnis im Bild 3.20 fällt auf, dass das Signal $u_{1,int}$ während der Pulsperioden unterschiedlich stark ansteigt. Da das Tastverhältnis immer größer wird, trifft dies auch auf die Pulsbreite zu, sowie für die Flächenberechnung unter dem Funktionsverlauf den der Integrator durchführt. Im letzten Subplot ist besonders gut sichtbar, dass der Mittelwert für die jeweilige Pulsperiode um die gewünschte Mittelungsdauer $t_m = T_a$ verzögert ist. Mit dieser Methode kann die Mittelwertmessung des Leistungsmessgerätes N5000 nachgebildet werden. Für das Signalprozessorsystem DS 1005 wird wie in 2.2.1 eine alternative Messmethode durchgeführt.

Für das Signalprozessorsystem DS 1005 wurde in MATLAB/Simulink der Koppelplan im Bild 3.22 implementiert. Dieser besteht aus zwei Zero-Order Hold-Blöcken mit denen pro Pulsperiode ein Abtastwert festgehalten wird. Der Unterschied zwischen den beiden besteht darin, dass der erste um die halbe Pulsperiode verschoben ist. Damit wird wie im Bild 2.9 zu sehen ist, der Augenblickswert als Mittelwert übernommen. Dieser Mittelwert wird nun eine Pulsperiode lang gehalten. Zum Zeitpunkt $T_a/2$ der nächsten Pulsperiode erfolgt die erneute Abtastung. Mit dem zweiten Zero-Order Hold wird das um $T_a/2$ verschobene Mittelwertsignal pro Pulsperiode abgetastet. Daraus erhält man das, um eine Pulsperiode verschobene Mittelwertsignal.



Bild 3.22: Simulink-Koppelplan für Mittelwertmessung mit Signalprozessorsystem DS 1005

In diesem Abschnitt wurde die Berechnung des Mittelwertes behandelt und präsentiert eine Möglichkeit, dies effektiv zu bewerkstelligen. Der Koppelplan wurde mit zwei Testversuchen geprüft und garantiert somit die richtigen Ergebnisse. Die Mittelwertberechnung ist für den Vergleich mit den Messungen zur Untersuchung der Umrichternichtlinearität wichtig und hiermit einfach möglich.

Im nächsten Absatz wird nun der Ausgang des Wechselrichters mit einem ohmsch-induktiven Verbraucher belastet. Damit fließt nun auch ein Strom i_1 und es kann die Simulation mit dem in 3.1 diskutierten theoretischen Verhalten des U-Umrichters verglichen werden.

3.4 Ohmsch-induktive Last

Bis jetzt wurde der Spannungszwischenkreisumrichter ohne Last betrachtet. Nun wird eine ohmsch-induktive Last hinzugefügt und das ideale Modell damit getestet. Das Ersatzschaltbild dafür ist im Bild 3.23 dargestellt. Im Vergleich mit der Messschaltung im

Bild 2.10 werden für die Simulation die zweite Halbbrücke, Widerstand, Induktivität und Kapazität zu einer Gleichstromquelle (Spannung U_g) zusammengefasst.

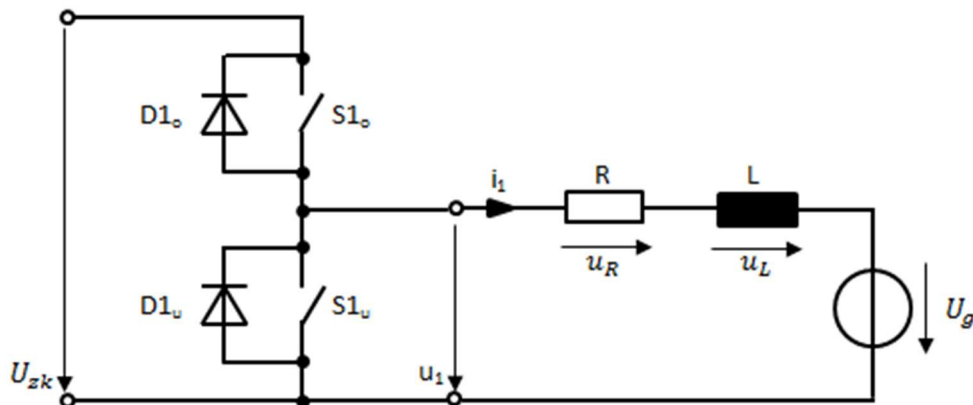


Bild 3.23: Ersatzschaltbild des Spannungszwischenkreisumrichters mit ohmsch-induktiver Last

Die Gegenspannung U_g ist notwendig, da sonst nur positive Ströme über die Halbbrücke möglich sind. Wird $U_g = U_{zk}/2$ festgelegt, kann mit dem Tastverhältnis d_1 der Halbbrücke ein positiver und ein negativer Strom geführt werden. Umgekehrt ist es auch möglich, das Tastverhältnis d_1 zu fixieren und die Gegenspannung zu variieren. In der Messung am Laborstand wurde das zweite Konzept verfolgt, da damit die Umrichternichtlinearität besser veranschaulicht werden kann. Mit $d_1 = 0.5$ und $U_g = U_{zk}/2$ fließt kein Strom, d.h. der Strommittelwert ist Null. Ist $U_g < U_{zk}/2$ fließt ein Strom mit positivem Mittelwert. Bei einer Gegenspannung von $U_g > U_{zk}/2$ fließt ein Strom mit negativem Mittelwert.

3.4.1 Mathematisches Modell der Last

Die Zustandsdifferentialgleichung des Ersatzschaltbildes wird folgendermaßen abgeleitet.

$$u_1 = u_R + u_L + U_g \quad (3.3)$$

Mit der Bauteilgleichung an der Induktivität

$$u_L = L \cdot \frac{di_1}{dt} \quad (3.4)$$

wird (3.3) zu

$$u_1 = i_1 \cdot R + L \cdot \frac{di_1}{dt} + U_g. \quad (3.5)$$

Durch Umformung erhält man aus (3.5) die Differentialgleichung

$$\frac{di_1}{dt} = \frac{1}{L} \cdot (-i_1 \cdot R + u_1 - U_g) \quad (3.6)$$

mit dem Strom i_1 als Zustandsgröße.

Mit der Differentialgleichung (3.6) ergibt sich folgender MATLAB/Simulink Koppelplan. Dies ist eine gewöhnliche Differentialgleichung 1. Ordnung mit konstanten Koeffizienten. Der elektrische Widerstand $R = 0,3 \Omega$ und die Induktivität $L = 10 \text{ mH}$.

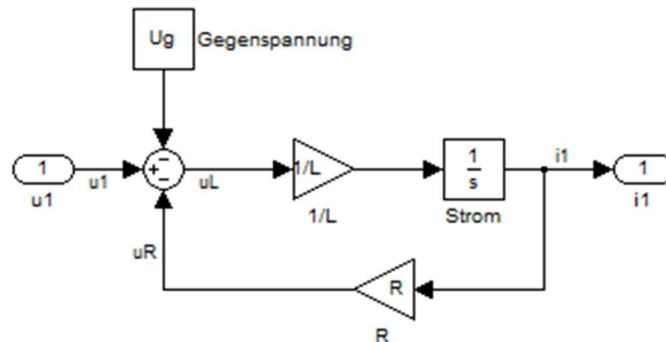


Bild 3.24: Simulink-Koppelplan der ohmsch-induktiven Last mit konstanter Gegenspannung

Das gesamte MATLAB/Simulink-Modell mit Spannungszwischenkreisumrichter, Last und Mittelwertmessung ergibt die im Bild 3.25 abgebildete Struktur.

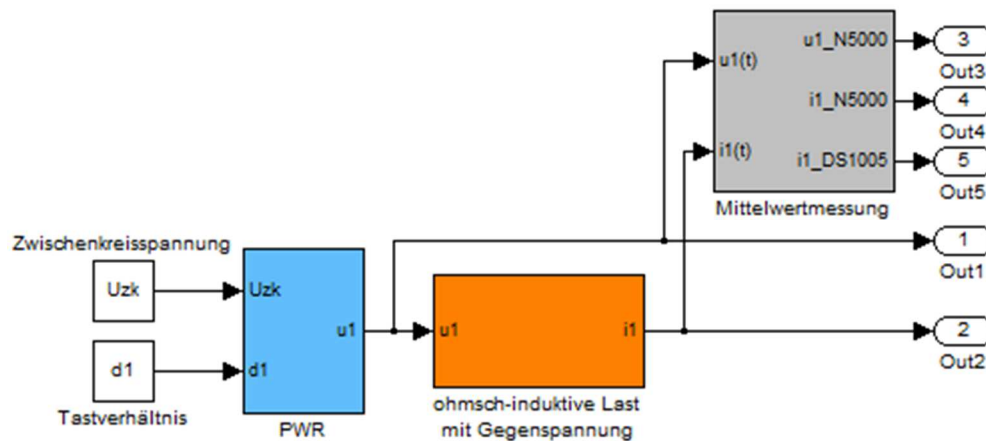


Bild 3.25: Ideales MATLAB/Simulink-Modell des Spannungszwischenkreisumrichters

Der Block „ohmsch-induktive Last mit Gegenspannung“ im Bild 3.25 enthält den Koppelplan aus Bild 3.24. Im Block „Mittelwertmessung“ wird für den Strom i_1 und die Spannung u_1 mit dem Koppelplan aus Bild 3.18 der Mittelwert ermittelt. Zusätzlich erfolgt darin die Strommittelwertmessung des Signalprozessorsystems DS 1005 (siehe Bild 3.19). Der Pulswechselrichter im Block „PWR“ hat den im Bild 3.26 dargestellten Koppelplan.

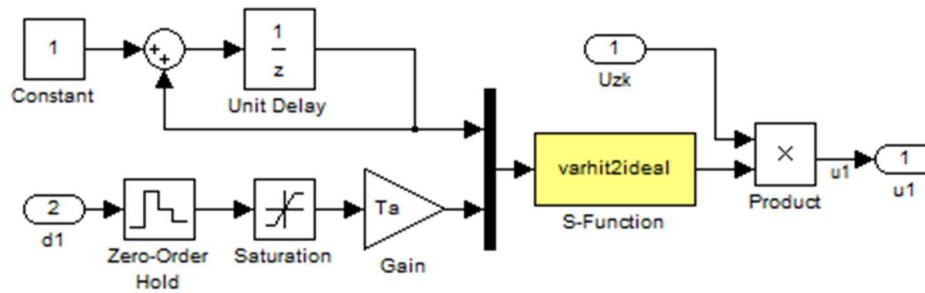


Bild 3.26: Koppelplan des idealen Spannungszwischenkreisumrichters im Block „PWR“

3.4.2 Validierung des idealen Modells

Zur Veranschaulichung des Pulswechselrichtermodells im Bild 3.25 mit ohmsch-induktiver Last wird das Tastverhältnis konstant auf $d_1 = 0,5$ gehalten und mittels der Gegenspannung U_g der gewünschte Strom \bar{i}_1 eingestellt. Es folgt nun ein Validierungsexperiment mit $\bar{i}_1 = -30$ A. Der Versuch testet das Modell bei negativem Strom \bar{i}_1 (siehe Bild 3.27). Für das folgende Experiment wird die Simulation für 0,2 Sekunden durchgeführt, das sind 1000 Pulsperioden. Die letzte Pulsperiode wurde im Plot dargestellt. Der Plot zeigt die Ausgangsspannung am Spannungszwischenkreisumrichter u_1 , den Strom i_1 , den Spannungsabfall am Widerstand u_R , den Spannungsabfall an der Induktivität u_L und die Gegenspannung U_g . Der Strom i_1 ist in diesem Modell die Zustandsgröße und kann sich nicht sprunghaft ändern. Mit dem Plot lassen sich die Gleichung (3.3) und die Zusammenhänge im mathematischen Modell nachvollziehen. Die Spannung am Widerstand u_R ist proportional zum Strom i_1 . Da der Strom i_1 Zustandsgröße und damit stetig ist, d.h. keine Sprünge aufweist, besitzt auch u_R diese Eigenschaft. Die Gegenspannung ist konstant. Damit muss die Spannung an der Induktivität springen um die Sprünge der Eingangsspannung u_1 auszugleichen und die Gleichung (3.3) zu erfüllen. Das PWM-Signal u_1 und die Gegenspannung sind fix vorgegeben. Der Strom i_1 , die Spannung am Widerstand u_R und die Spannung an der Drossel u_L ergeben sich durch die mathematischen Zusammenhänge.

3.4.3 Diskussion der Validierung

Das Simulationsergebnis für den Validierungsversuch mit $\bar{i}_1 = -30$ A ist im Bild 3.27 dargestellt. Der Vergleich mit Bild 3.7 zeigt, dass das Simulationsmodell das gewünschte PWM-Signal für u_1 liefert. Die erforderliche Gegenspannung U_g für $\bar{i}_1 = -30$ A lässt sich mit der Gleichung (3.3) berechnen und ergibt für die Mittelwerte

$$\bar{u}_1 = \bar{u}_R + \bar{u}_L + U_g. \quad (3.7)$$

Bei konstantem Tastverhältnis d_1 ist der Mittelwert für $\bar{u}_L = 0 \text{ V}$ und der Spannungsabfall am Widerstand wird zu $\bar{u}_R = \bar{i}_1 \cdot R$. Damit wird aus (3.7)

$$\bar{u}_1 = \bar{i}_1 \cdot R + U_g. \quad (3.8)$$

Mit der Spannung am Ausgang des Pulswechselrichters $\bar{u}_1 = U_{zk} \cdot d_1 = 100 \text{ V} \cdot 0,5 = 50 \text{ V}$, dem ohmsche Widerstand $R = 0,3 \Omega$ und dem gewünschten Strom $\bar{i}_1 = -30 \text{ A}$ ergibt sich für die Gegenspannung

$$U_g = \bar{u}_1 - \bar{i}_1 \cdot R = 50 \text{ V} + 30 \text{ A} \cdot 0,3 \Omega = 59 \text{ V}. \quad (3.9)$$

Die Zeitkonstante τ ist im Verhältnis zur Pulsperiode T_a sehr groß. Damit ergibt sich der dreieckförmige (lineare) Funktionsverlauf für i_1 und u_R . Es ist somit nur ein kurzes Anfangsstück des exponentiellen Lade- bzw. Entladevorganges zu sehen.

$$\tau = \frac{L}{R} = \frac{0,01 \frac{\text{Vs}}{\text{A}}}{0,3 \frac{\text{V}}{\text{A}}} = \frac{1}{30} \text{ s} = 0,033 \text{ s} \quad (3.10)$$

$$\tau = 0,033 \text{ s} \gg 0,0002 \text{ s} = T_a \quad (3.11)$$

Der Spannungsabfall an der Induktivität hat im Validierungsversuch einen rechteckförmigen Funktionsverlauf. Dies ergibt sich wie vorhin, durch die große Zeitkonstante. Damit ist mit freiem Auge kein exponentieller Verlauf zwischen den Spannungssprüngen zu sehen.

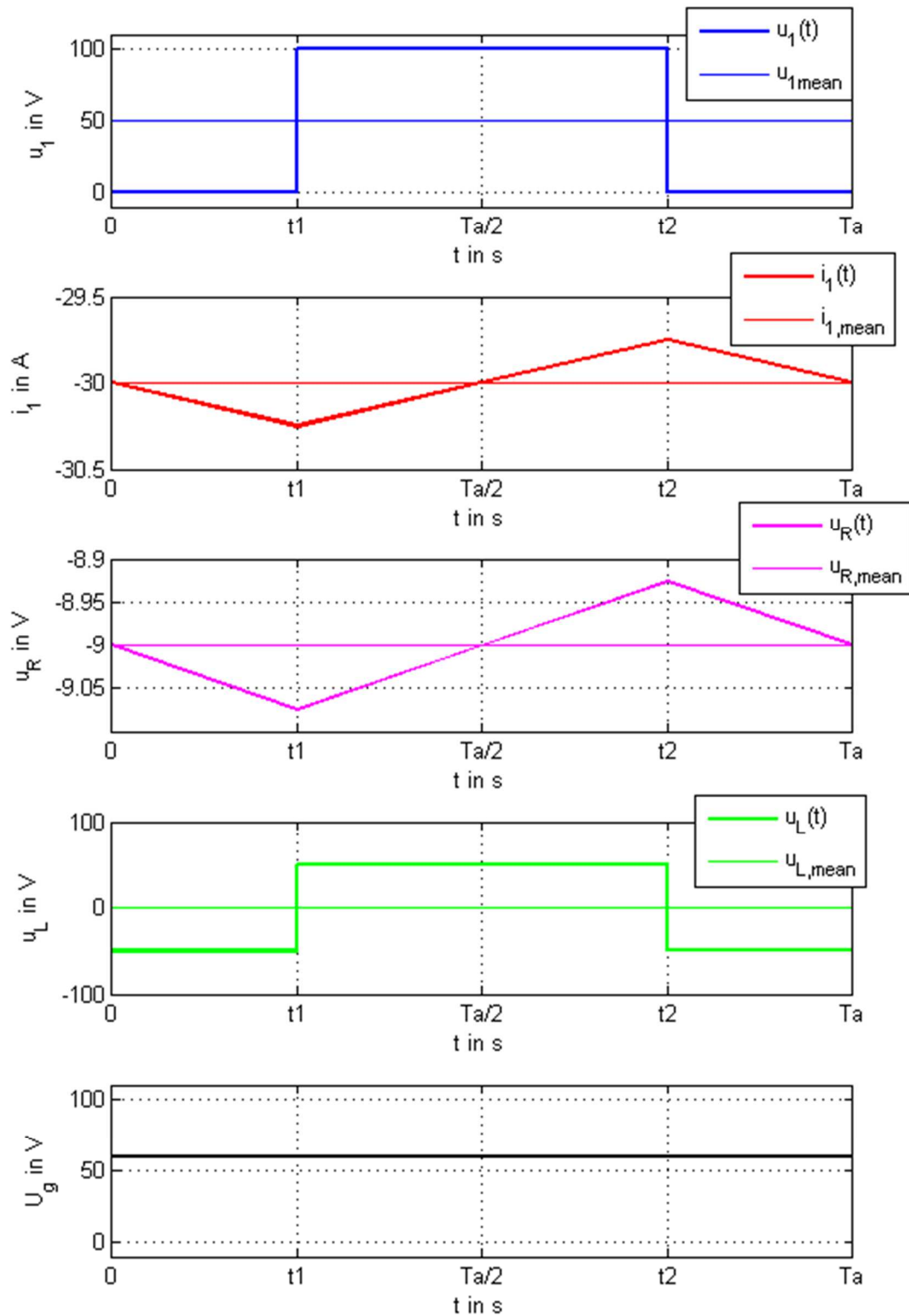


Bild 3.27: Validierungsexperiment mit negativem Strom $\bar{i}_1 = -30$ A

3.5 Vergleich mit Messung

Das ideale Modell des Spannungszwischenkreisumrichters ist wie zuvor erwähnt, nur eine Stütze für die weitere Modellverfeinerung. Das ideale Modell berücksichtigt keinen Spannungsabfall an der Umrichternichtlinearität, d.h. die Spannung $\bar{u}_{1,NL} = 0 \text{ V}$. Der Vergleich mit der Messung liefert, wie erwartet, die Abweichungen die im Bild 2.11 zu sehen sind.

Im nächsten Kapitel erfolgt nun der erste Schritt für die Berücksichtigung der Umrichternichtlinearität mittels der Verriegelungszeit.

4 Modell mit Verriegelungszeit

In diesem Kapitel folgt der erste Schritt für die Modellierung der Umrichternichtlinearität bei Spannungszwischenkreisumrichtern. Dabei wird im speziellen der Einfluss der Verriegelungszeit behandelt. Als erstes erfolgt die allgemeine Erklärung der Verriegelungszeit und deren Auswirkungen auf die Pulsweitenmodulation. Als nächstes folgt die konkrete Modellerweiterung für die Berücksichtigung der Verriegelungszeit. Anschließend werden die gewünschten Modelleigenschaften und der damit erhaltene Funktionsverlauf der Umrichternichtlinearität mit Testsimulationen verifiziert. Abschließend erfolgen der Vergleich mit dem gemessenen Verlauf der Umrichternichtlinearität und eine Diskussion der Modellqualität.

4.1 Verriegelungszeit

In der Realität ist es besonders wichtig zu beachten, dass die beiden Schalter in der Halbbrücke kein ideales Schaltverhalten besitzen wie es in Kapitel 3 vorgestellt wurde. Wird die Bedingung $S_{1o} = \bar{S}_{1u}$ verletzt, führt dies zu einem Kurzschluss des Zwischenkreises über die Halbbrücke. In der Ansteuerung der elektronischen Schalter (hier IGBT) sind die Schaltzeitpunkte der beiden Schalter unterschiedlich. Der eine elektronische Schalter benötigt eine gewisse Zeit zum Öffnen, erst dann kann der Andere geschlossen werden. Der Schaltvorgang ist keine Sprungfunktion, sondern hat eine endliche Anstiegszeit. Die Verriegelungszeit ist eine kurze Zeitspanne, die zwischen dem Start des Öffnungsvorgangs und dem Ende des Schließvorgangs vergeht. Für die Verriegelungszeit (engl. Dead time) ist auch der Begriff Totzeit gebräuchlich. Durch die Verriegelungszeit wird gewährleistet, dass kein Kurzschluss über die Halbbrücke auftritt. Die Verriegelungszeit t_v hat somit eine Auswirkung auf das Tastverhältnis d_1 und dadurch auch auf die Ausgangsspannung des Spannungszwischenkreisumrichters. In der folgenden Untersuchung wird ersichtlich, dass der Einfluss der Totzeit auf die Spannung am Umrichter Ausgang vom Vorzeichen des Stromes abhängig ist (siehe [3] und [2]).

Für die beiden Untersuchungen mit positivem und negativem Strom wird die Pulsperiode, wie aus Bild 4.1 und Bild 4.7 ersichtlich, in fünf Zeitintervalle unterteilt.

- a) $0 \leq t < t_1$
- b) $t_1 \leq t < t_{1v}$
- c) $t_{1v} \leq t < t_2$
- d) $t_2 \leq t < t_{2v}$
- e) $t_{2v} \leq t < T_a$

4.1.1 Strom positiv

Im Bild 4.1 bis Bild 4.6 sind die Vorgänge bei positivem Strom dargestellt. Der Spannungsverlauf am Umrichterausgang ist im Bild 4.1 abgebildet. Dabei ist zu sehen, wie die Spannungszeitfläche im Vergleich zum idealen Verhalten kleiner wird. Warum dies so ist, wird näher im Bild 4.2 bis Bild 4.6 erläutert. Dabei beschreibt jede Abbildung den Zustand der Halbbrücke im entsprechenden Zeitabschnitt.

- a) Schaltzustand während dem Zeitintervall $0 \leq t < t_1$ (siehe Bild 4.2)

Im ersten Zeitabschnitt ist der Schalter $S1_u$ geschlossen und das Potential am Ausgang ist eindeutig auf 0 V. Der Strom fließt bei positivem Vorzeichen über die untere Freilaufdiode ($D1_u$), da der IGBT nicht rückstromfähig ist.

- b) Schaltzustand während dem Zeitintervall $t_1 \leq t < t_{1v}$ (siehe Bild 4.3)

Während der Verriegelungszeit t_v im Intervall $t_1 \leq t < t_{1v}$ beginnt mit t_1 das Öffnen von $S1_u$. Der Strom fließt weiterhin über die Freilaufdiode und das Potential am Ausgang des Umrichters bleibt, im Gegensatz zum idealen Verhalten weiterhin auf 0 V.

- c) Schaltzustand während dem Zeitintervall $t_{1v} \leq t < t_2$ (siehe Bild 4.4)

Im nächsten Zeitintervall ist zum Zeitpunkt t_{1v} der Schalter $S1_o$ geschlossen und das Potential am Ausgang springt auf den Wert der Zwischenkreisspannung. Der Strom wechselt seinen Pfad von der unteren Freilaufdiode ($D1_u$) zum oberen IGBT ($S1_o$).

- d) Schaltzustand während dem Zeitintervall $t_2 \leq t < t_{2v}$ (siehe Bild 4.5)

Bei diesem Zeitabschnitt öffnet zum Zeitpunkt t_2 der Schalter $S1_o$. Der Strom wechselt zurück zur unteren Freilaufdiode und zieht damit das Potential am Ausgang auf 0 V.

- e) Schaltzustand während dem Zeitintervall $t_{2v} \leq t < T_a$ (siehe Bild 4.6)

Im letzten Zeitstück ist der Schalter $S1_u$ bei t_{2v} geschlossen und das Potential am Umrichterausgang bleibt auf 0 V definiert.

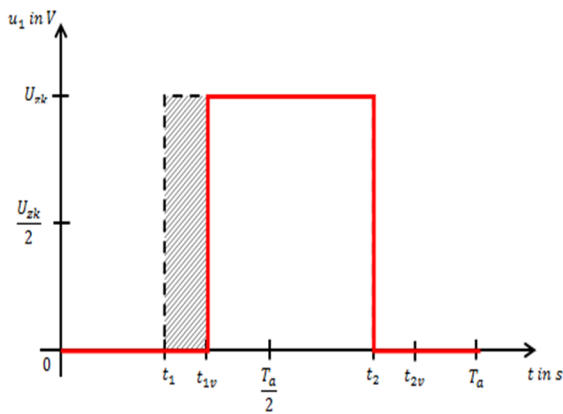


Bild 4.1: PWM mit t_v bei $i_l > 0$

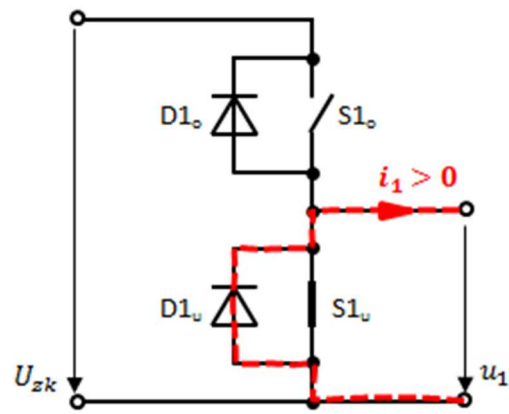


Bild 4.2: $0 \leq t < t_1$

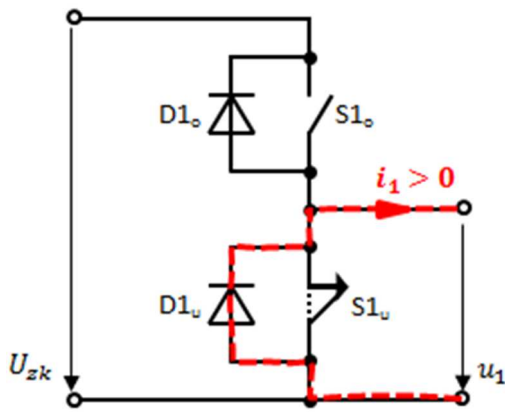


Bild 4.3: $t_1 \leq t < t_{1v}$

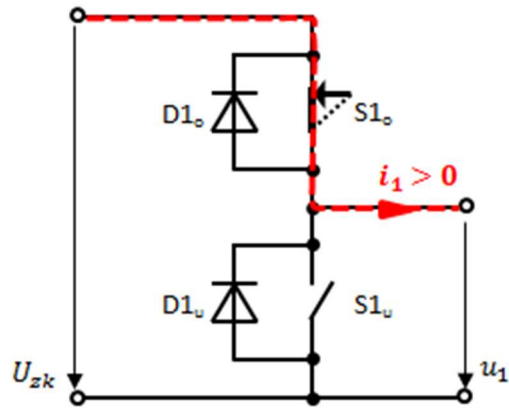


Bild 4.4: $t_{1v} \leq t < t_2$

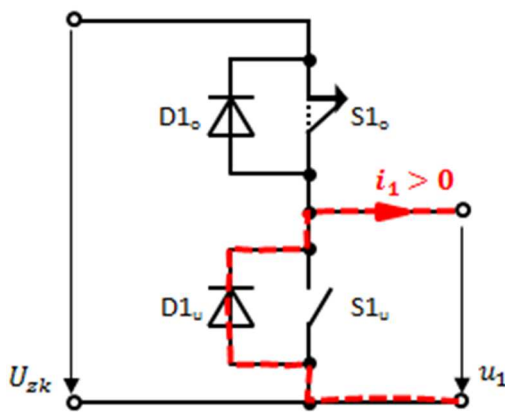


Bild 4.5: $t_2 \leq t < t_{2v}$

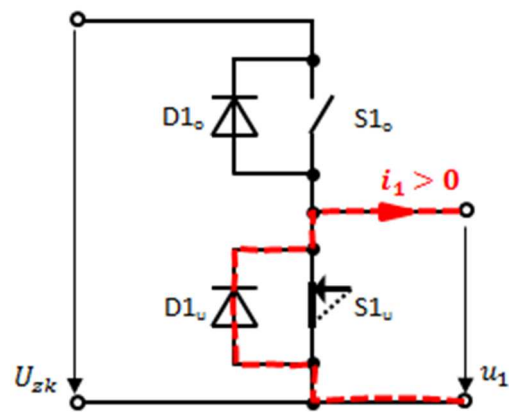


Bild 4.6: $t_{2v} \leq t < T_a$

Fazit aus den Betrachtungen zum positiven Strom

Der Vergleich von Bild 3.1 mit Bild 4.1 zeigt, dass die Verriegelungszeit die ideale Pulsbreite verkleinert und damit auch der Mittelwert am Pulsrichterausgang kleiner ist.

4.1.2 Strom negativ

Nun folgt das Verhalten bei negativem Strom. Im Bild 4.7 ist der Spannungsverlauf am Ausgang des Spannungszwischenkreisumrichters für diesen Fall dargestellt. Die Zeitachse wird zur Erklärung wieder wie vorher in fünf Teile unterschieden. Zu diesen fünf Zeitintervallen sind im Bild 4.8 bis Bild 4.12 die zugehörigen Schaltzustände an der Halbbrücke abgebildet.

- a) Schaltzustand während dem Zeitintervall $0 \leq t < t_1$ (siehe Bild 4.8)

Das Ausgangspotential des Umrichters liegt im ersten Zeitintervall auf 0 V, da $S1_u$ geschlossen ist. Der Strom fließt nun mit negativem Vorzeichen über den IGBT.

- b) Schaltzustand während dem Zeitintervall $t_1 \leq t < t_{1v}$ (siehe Bild 4.9)

Im zweiten Zeitintervall öffnet der Schalter $S1_u$ zum Zeitpunkt t_1 und es ist die Verriegelungszeit t_v wirksam. Dabei ändert der Strom den Weg und fließt nun über die obere Freilaufdiode und das Potential springt auf den Wert der Zwischenkreisspannung.

- c) Schaltzustand während dem Zeitintervall $t_{1v} \leq t < t_2$ (siehe Bild 4.10)

Nach dem Ablauf der Totzeit kommt der dritte Zeitabschnitt. Zum Zeitpunkt t_{1v} ist der obere Schalter $S1_o$ geschlossen. Das Potential am Ausgang bleibt auf U_{zk} und der Strom fließt weiterhin über die Freilaufdiode, da der IGBT nicht rückstromfähig ist.

- d) Schaltzustand während dem Zeitintervall $t_2 \leq t < t_{2v}$ (siehe Bild 4.11)

In dieser Zeitspanne öffnet der Schalter $S1_o$ bei t_2 und der Strom fließt weiterhin über die Freilaufdiode ($D1_o$). Die Totzeit ist nun wieder wirksam und das Ausgangspotential des Pulswechselrichters bleibt weiter auf dem Niveau der Zwischenkreisspannung.

- e) Schaltzustand während dem Zeitintervall $t_2 \leq t < T_a$ (siehe Bild 4.12)

Im letzten Zeitintervall der Pulsperiode wird zum Zeitpunkt t_{2v} der Schalter $S1_u$ geschlossen und das Potential am Ausgang wird auf 0 V fixiert. Der Strom fließt nun wieder über den unteren IGBT ($S1_u$).

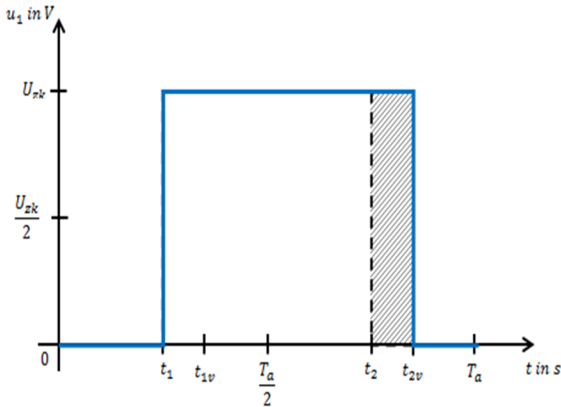


Bild 4.7: PWM mit t_v bei $i_1 < 0$

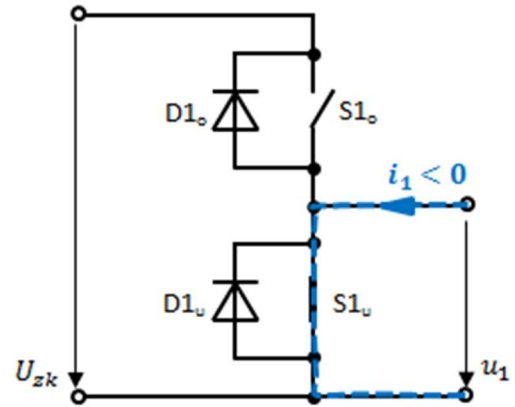


Bild 4.8: $0 \leq t < t_1$

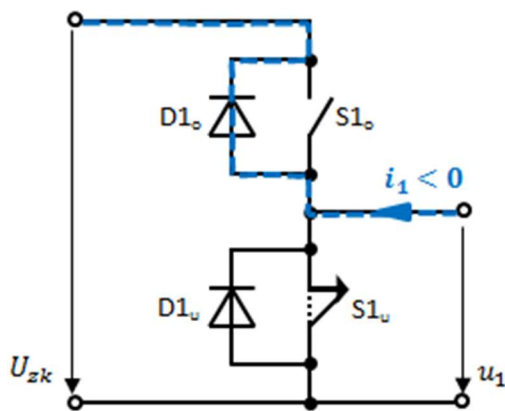


Bild 4.9: $t_1 \leq t < t_{1v}$

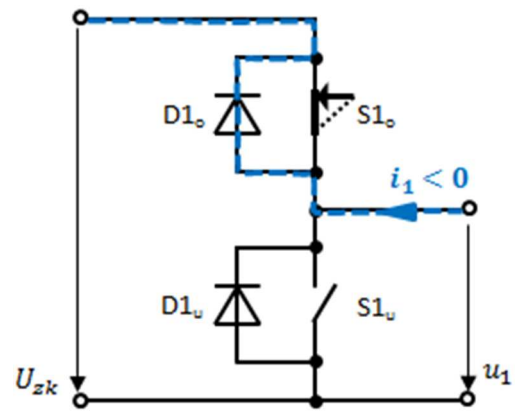


Bild 4.10: $t_{1v} \leq t < t_2$

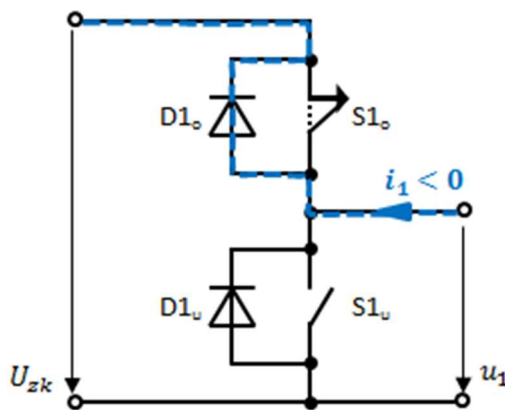


Bild 4.11: $t_2 \leq t < t_{2v}$

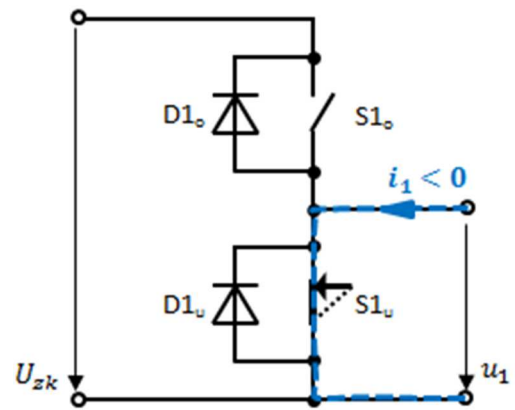


Bild 4.12: $t_{2v} \leq t < T_a$

Fazit aus den Betrachtungen zum negativen Strom

Der Vergleich zwischen idealem Modell im Bild 3.7 und dem Modell mit Berücksichtigung der Totzeit im Bild 4.7 führt zu einer Vergrößerung der Pulsbreite. Fließt ein negativer Strom über die Halbbrücke, vergrößert sich dabei der Spannungsmittelwert am Ausgang des Pulswechselrichters.

4.1.3 Strom null

Wie aus den beiden obigen Betrachtungen ersichtlich, sind nur strikt positive bzw. negative Ströme behandelt worden. Was passiert, wenn zum Zeitpunkt t_1 bzw. t_2 der Strom null ist? Bleibt die Spannung u_1 auf 0 V oder springt sie auf den Wert der Zwischenkreisspannung? Die Spannungsabfälle am Widerstand und an der Induktivität sind null. Daraus folgt aus (3.7), dass $\bar{u}_1 = U_g$. Bei Strom $\bar{i}_1 = 0$ ist die Gegenspannung $U_g = U_{zk}/2$, somit erhält man für $\bar{u}_1 = U_{zk}/2$. In diesem Fall wird zu den Zeitpunkten t_1 und t_2 die halbe Zwischenkreisspannung am Ausgang des PWR angelegt. Für die Dauer der Verriegelungszeit bleicht u_1 auf $U_{zk}/2$. Es stellt sich nun die Frage wie oft dieses Ereignis eintritt und welche Auswirkungen auf die Simulation auftreten? Mehr dazu in der Analyse der Testsimulation für die Umrichternichtlinearität.

4.1.4 Diskussion zur Abhängigkeit vom Stromvorzeichen

Durch die Berücksichtigung der Verriegelungszeit wird wie vorhin schon erwähnt die Pulsbreite in Abhängigkeit vom Stromvorzeichen beeinflusst. Dadurch ändert sich auch die Aufteilung der Stromführung zwischen IGBT und Freilaufdiode. Im Vergleich mit dem idealen Modell in Bild 3.13 wird bei positivem Strom die Zeit in der die Freilaufdiode $D1_u$ den Strom führt um die Totzeit größer und damit das Zeitintervall für $S1_o$ um t_v verkleinert (siehe Bild 4.13).

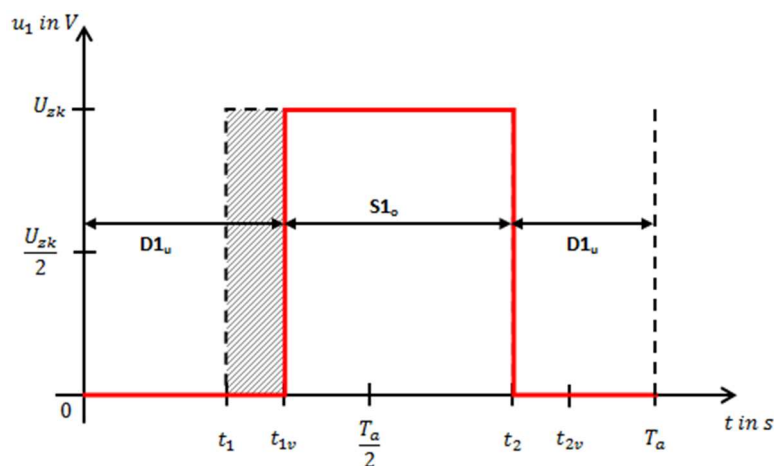


Bild 4.13: Aufteilung des Stromes $i_1 > 0$ auf IGBT und Freilaufdiode unter Berücksichtigung der Verriegelungszeit

Für einen negativen Strom gilt grundsätzlich dasselbe, nur wird hier die Zeit von der oberen Freilaufdiode $D1_o$ um t_v vergrößert bzw. der Zeitabschnitt, in dem der untere Schalter $S1_u$ den Strom führt, um die Verriegelungszeit verkürzt (siehe Bild 4.14).

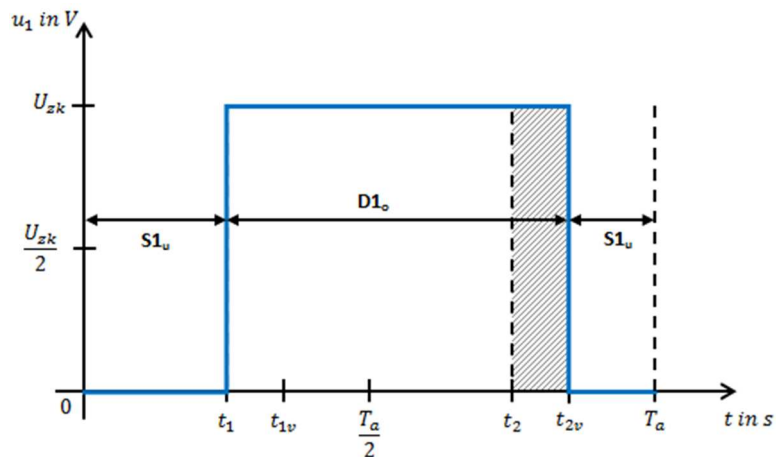


Bild 4.14: Aufteilung des Stromes $i_1 < 0$ auf IGBT und Freilaufdiode unter Berücksichtigung der Verriegelungszeit

4.2 Modellerweiterung für die Berücksichtigung der Verriegelungszeit

Im Bild 4.15 ist der Simulink-Koppelplan für die Modellerweiterung mit Berücksichtigung der Verriegelungszeit abgebildet. Die Blöcke „ohmsch-induktive Last mit Gegenspannung“ bzw. „Mittelwertmessung“ bleiben unverändert. Im Unterschied zum idealen Modell im Bild 3.25 besitzt der PWR-Block den Strom i_1 als zusätzlichen Eingang.

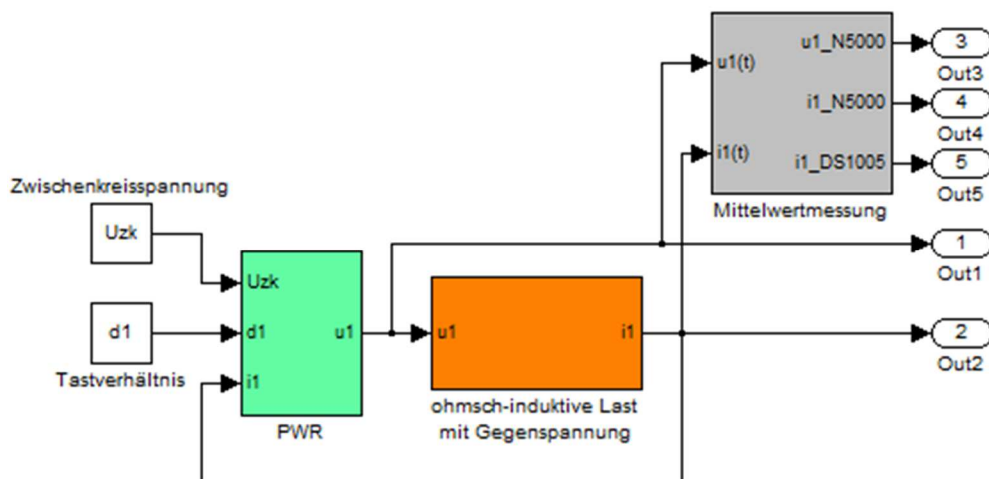


Bild 4.15: Simulink-Koppelplan für das Modell mit Berücksichtigung der Verriegelungszeit

4.2.1 Eingangs- bzw. Ausgangssignale der S-Function

Der Block für den Pulswechselrichter „PWR“ ist ähnlich zu dem im Bild 3.26 dargestellten Koppelplan des idealen Modells. Für die Berücksichtigung der Verriegelungszeit hat die S-Function den Momentanwertverlauf des Stromes als zusätzlichen dritten Eingang. Damit ist es möglich in Abhängigkeit vom Stromvorzeichen das Ausgangssignal der S-Function zu beeinflussen.

4.2.2 Simulationsablauf in der S-Function

Die MATLAB-Funktion „varhit4tvi.m“ im S-Function-Block wird nun um die Berücksichtigung der Verriegelungszeit erweitert. Für die Entscheidung zu den Zeitpunkten t_1 und t_2 steht der Momentanwert des Stromes zu Verfügung. Es ist auch ein zusätzlicher diskreter Zustand notwendig um die Zeitintervalle in denen die Verriegelungszeit aktiv ist zu speichern. Im Vergleich zum idealen Modell werden zwei weitere Zeitpunkte (t_{1v} bzw. t_{2v}) berücksichtigt. Bis zum Zeitpunkt t_1 verläuft alles gleich wie im idealen Modell. Zum Zeitpunkt t_1 erfolgt im Schritt „GetTimeOfNextVarHit“ die Vorgabe des nächsten Berechnungszeitpunktes an dem die S-Function aufgerufen wird zum Zeitpunkt $t_{1v} = t_1 + t_v$. In nächste Schritt „Outputs“ wird nun in Abhängigkeit vom Stromvorzeichen der Ausgang zum Zeitpunkt t_1 festgelegt. Aus 3.1.1 und 3.1.2 abgeleitet, bleibt der Wert für einen positiven Strom auf null und für einen negativen Strom wird er auf 1 gesetzt. Für die Dauer der Verriegelungszeit bleibt der Ausgang der S-Function auf dem entsprechenden Wert. Für den nächsten Zeitpunkt t_{1v} wird der Ausgang der S-Function unabhängig vom Strom auf den Wert 1 gelegt. Zum Zeitpunkt t_2 wird als nächster Berechnungszeitpunkt t_{2v} festgelegt. Der Ausgangswert der S-Function ist nun wieder vom Stromvorzeichen abhängig und wird im Fall $i_1 > 0$ auf null bzw. für $i_1 < 0$ auf 1 gesetzt. Der Ausgang der S-Function wird zum Zeitpunkt t_{2v} unabhängig vom Stromvorzeichen null. Für Details der Änderungen in der S-Function wird auf den Quellcode im Anhang A.2 verwiesen.

Die Verriegelungszeit wird über die Dialogbox „Function Block Parameters: S-Function“ als zusätzlicher Parameter t_v definiert.

In Tabelle 2 ist eine Testsimulation über drei Pulsperioden aufgelistet. Der Mittelwert des Stromes beträgt dabei $\bar{i}_1 = 1$ A. Im Vergleich mit Tabelle 1 aus dem vorhergehenden Kapitel mit dem idealen Modell sind der zusätzliche Zustand x_2 und der dritte Eingang u_3 hinzugekommen. In der Spalte *sys* sind für den Schritt Update zwei Werte eingetragen, dies sind die beiden Zustandswerte für den nächsten Zeitpunkt. Pro Pulsperiode sind zwei Zeitpunkte mehr, diese sind t_{1v} und t_{2v} .

Die in fett geschriebenen Einträge der Zeitspalte t und in der Spalte mit sys (Output) ergeben den im Bild 4.1 dargestellten Funktionsverlauf für die Spannung. Dabei ist zu beachten, dass der Output in der Variable sys noch mit der Zwischenkreisspannung U_{zk} zu multiplizieren ist.

4.2.3 Funktionstest des erweiterten Modells

Um die korrekte Funktionsweise der PWM mit Berücksichtigung der Verriegelungszeit zu überprüfen, wurde der S-Function-Block mit den Stromwerten $i_1 > 0$, $i_1 < 0$ und $i_1 = 0$ getestet. Zur besseren Veranschaulichung der Auswirkungen auf die Ausgangsspannung durch die Verriegelungszeit, wird diese mit $t_v = 10 \mu\text{s}$ willkürlich gewählt. Das MATLAB/Simulink-Modell aus Bild 4.15 wurde für den Funktionstest verändert. Die Testsimulation wurde ohne Last durchgeführt und der Strom i_1 mit konstanten Werten vorgegeben.

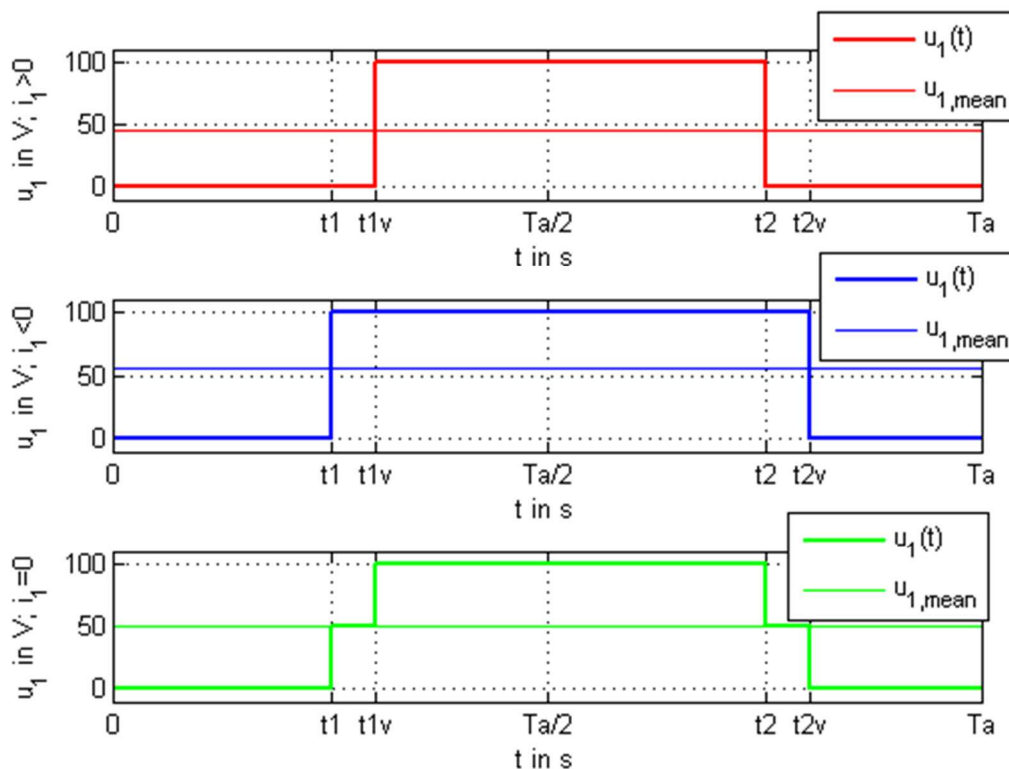


Bild 4.16: Vergleich der Testergebnisse für $i_1 > 0$, $i_1 < 0$ und $i_1 = 0$

Bild 4.16 bestätigt das gewünschte Modellverhalten. Für $i_1 > 0$ wird die Pulsbreite gegenüber dem idealen um t_v schmaler und der Spannungsmittelwert \bar{u}_1 sinkt in Abhängigkeit von t_v . Hingegen ist für $i_1 < 0$ die Pulsbreite um t_v größer und der Spannungsmittelwert \bar{u}_1 steigt, abhängig von t_v . Wenn $i_1 = 0$ bleibt der

Spannungsmittelwert \bar{u}_1 auf dem Wert des idealen Modells ($\bar{u}_1 = d_1 \cdot U_{zk}$) und ein Spannungsabfall an der Umrichternichtlinearität ist nicht vorhanden.

4.2.4 Testsimulation für die Umrichternichtlinearität

Für die Erzeugung der Umrichternichtlinearität wird das Tastverhältnis konstant auf $d_1 = 0.5$ gehalten und die Gegenspannung U_g schrittweise variiert. Durch die Veränderung von U_g können die gewünschten Strommittelwerte $-20 \text{ A} < \bar{i}_1 < 20 \text{ A}$ eingestellt, und damit der Spannungsabfall an der Umrichternichtlinearität in Abhängigkeit vom Strommittelwert in einem Diagramm dargestellt werden. Für jeden gewünschten Gegenspannungswert wird ein eigener Simulationsdurchlauf von 0,2s durchgeführt. Durch die konstante Tastrate d_1 erhält man für die Simulationsdauer einen stationären Wert für die Mittelwerte. Die Mittelwerte aus der letzten Pulsperiode werden abgespeichert und für die graphische Darstellung verwendet. Aus den vorigen Überlegungen folgt für die Umrichternichtlinearität der Funktionsverlauf einer Signum-Funktion.

Um den Einfluss der Verriegelungszeit zu veranschaulichen, wird der ideale Fall mit fünf verschiedenen großen Totzeiten verglichen. Wie im Bild 4.17 zu sehen, ist auf der x-Achse der jeweilige Strommittelwert aufgetragen, auf der y-Achse der Spannungsabfall $\bar{u}_{1,NL}$ verursacht durch die Verriegelungszeit t_v . Die Berechnung von $\bar{u}_{1,NL}$ erfolgt mit

$$\bar{u}_{1,NL}(k) = \frac{1}{T_a} \int_{k \cdot T_a}^{(k+1) \cdot T_a} (u_{1,soll} - u_{1,ist}) dt. \quad (4.1)$$

Dabei ist in der Simulation $\bar{u}_{1,soll} = U_{zk} \cdot d_1$ und $\bar{u}_{1,ist}$ ist das PWM-Signal mit Berücksichtigung der Verriegelungszeit.

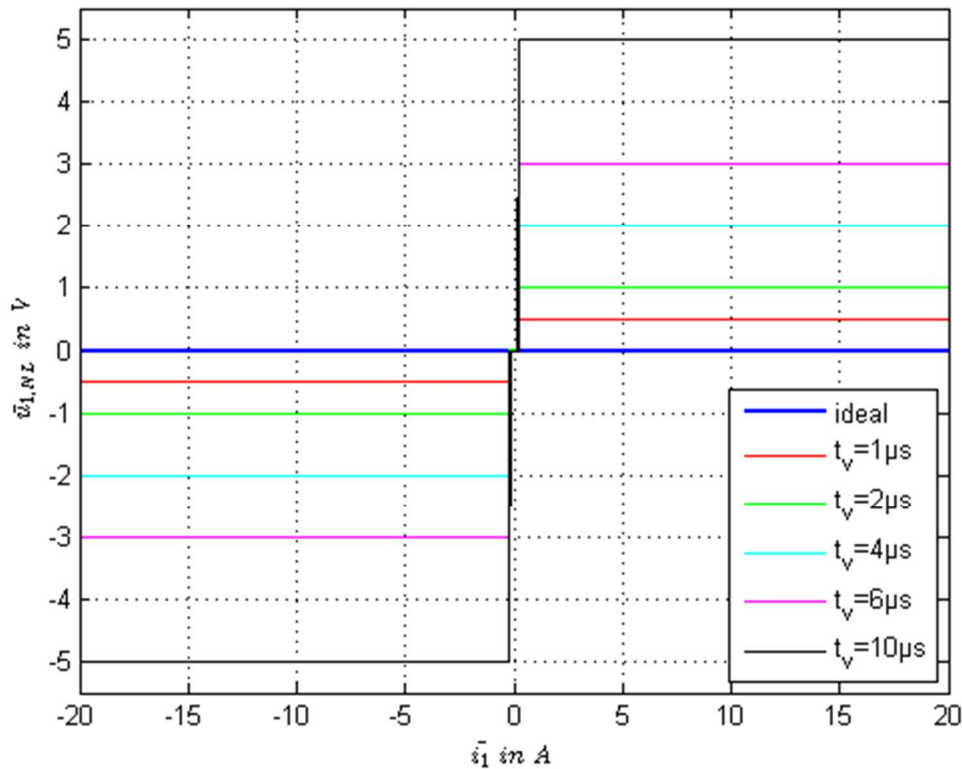


Bild 4.17: Testversuch zur Modellierung der Umrichternichtlinearität mit unterschiedlicher Verriegelungszeit

Je nach Wahl der Verriegelungszeit variiert die Größe des Spannungsverlusts. Der Spannungsabfall an der Umrichternichtlinearität kann in diesem Fall einfach mit

$$\bar{u}_{1,NL} = \frac{t_v}{T_a} \cdot U_{zk} \quad (4.2)$$

berechnet werden.

Die vermutete Signum-Funktion ist damit für das Modell nachgewiesen, dennoch ist ein genauerer Blick auf den sprunghaften Übergang nötig, der im Bild 4.18 dargestellt ist. Daraus wird ersichtlich, dass der Funktionsverlauf der Nichtlinearität von der Signum-Funktion abweicht.

Die Analyse dieser Graphik ergibt, dass die Kennlinien der Umrichternichtlinearität prinzipiell in drei Bereiche eingeteilt werden können. Für einen positiven Strom erfolgt mit dem Start im Koordinatenursprung ein schmaler Abschnitt der Kennlinie, in dem keine Verriegelungszeit wirksam ist und daher der Spannungsabfall aufgrund der Umrichternichtlinearität Null ist. Dies ist der erste Bereich. Der zweite Bereich ist der sprunghafte Übergang zum wirksamen Abschnitt der Totzeit. Im dritten Bereich ist der durch die Umrichternichtlinearität bedingte Spannungsabfall, für alle größer werdenden Strommittelwerte, konstant. Diese Unterteilung kann in selber Weise mit dem negativen

Strom getroffen werden. Der letztgenannte Bereich zeigt das erwartete Verhalten des Modells. Dies wurde auch im vorhergehenden Test bestätigt. Eine spezielle Untersuchung erfolgt für die beiden erstgenannten Bereiche. Besonderes Augenmerk wird hierbei auf den sprunghaften Übergang gelegt. Es folgen für beide interessierenden Bereiche Testsimulationen um die Momentanwerte u_1 und i_1 zu diskutieren.

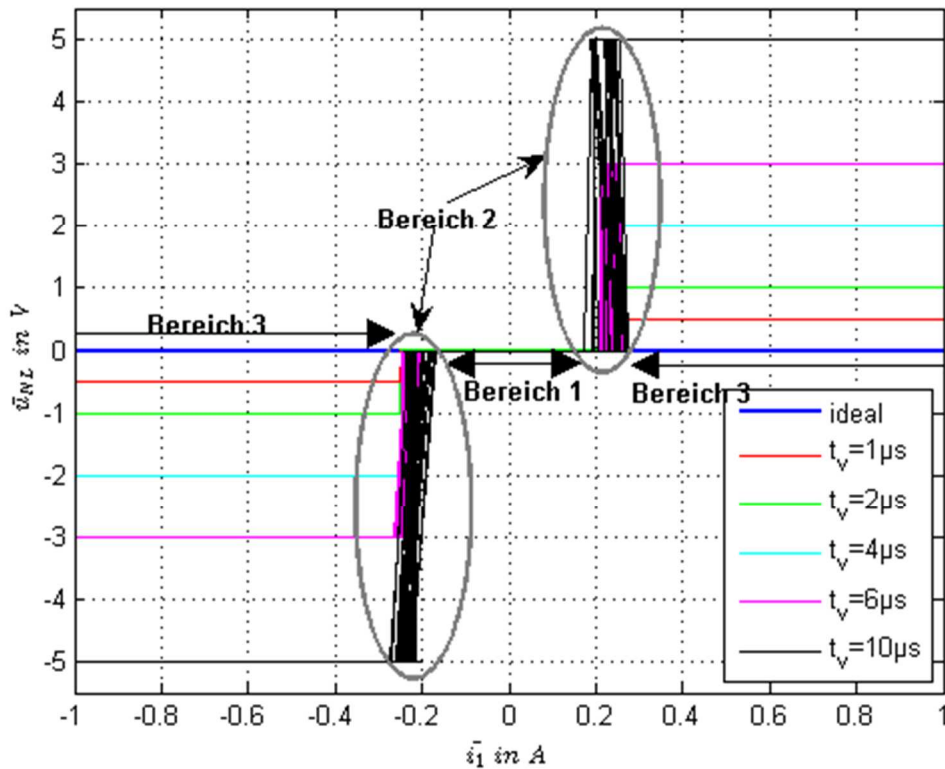


Bild 4.18: Ausschnitt der Umrichternichtlinearität für $-1 \text{ A} \leq \bar{i}_1 \leq 1 \text{ A}$

4.2.5 Analyse der Simulationsprobleme

Bereich 1

Für den Testversuch wurde der Strommittelwert mit $\bar{i}_1 = 0.1 \text{ A}$ gewählt. Die Simulationsergebnisse sind im Bild 4.19 abgebildet. Zwar ist der Mittelwert größer als null, der Funktionsverlauf des Stromes hat jedoch auch negative Werte. Damit ergeben sich Vorzeichenwechsel im Stromsignal und es kommt zu einem speziellen Spannungsverlauf am Ausgang des Spannungszwischenkreisumrichters, der gleich dem idealen ist. Die Verhältnisse an der Halbbrücke in diesem Abschnitt sind im Bild 4.21 bis Bild 4.25 abgebildet. Es erfolgt wieder wie vorhin die Unterteilung in fünf Zeitintervalle.

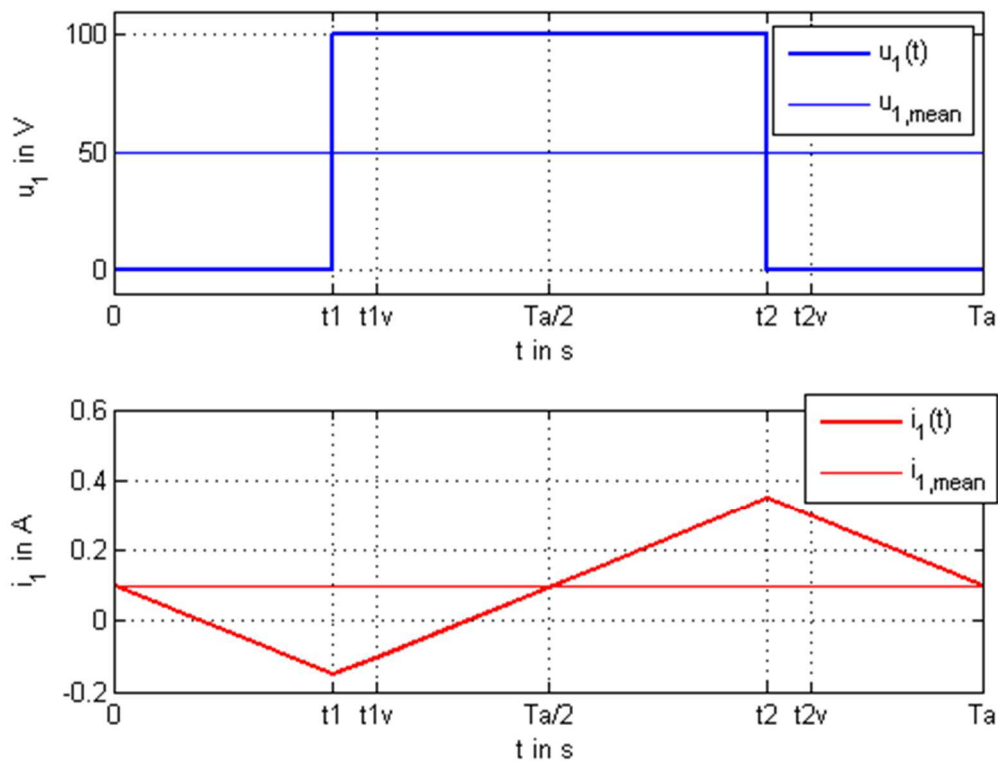


Bild 4.19: Testsimulation für Bereich 1 mit $\bar{i}_1 = 0,1 \text{ A}$

- a) Schaltzustand während dem Zeitintervall $0 \leq t < t_1$ (siehe Bild 4.21)

Im ersten Zeitintervall ist der Strom zu Beginn positiv und der Strompfad verläuft über die Freilaufdiode. Anschließend erfolgt eine Stromumkehr und er fließt über den IGBT. Dabei bleibt das Potential am Ausgang auf 0 V, da dies durch $S1_u$ festgelegt ist.

- b) Schaltzustand während dem Zeitintervall $t_1 \leq t < t_{1v}$ (siehe Bild 4.22)

Während dem nächsten Zeitabschnitt ist die Verriegelungszeit wirksam. Zum Zeitpunkt t_1 öffnet $S1_u$ und der Strom in negativer Richtung wechselt den Pfad von $S1_u$ auf die obere Freilaufdiode. Nun ist das Potential nicht mehr über die Halbbrücke definiert, sondern wird durch die Last vorgegeben, da beide leistungselektronischen Schalter geöffnet sind. Damit der Strom über die obere Freilaufdiode fließen kann, muss das Potential am Ausgang auf U_{zk} springen. Dies ist durch die Induktivität in der Last möglich. Der Strom hat in dieser Zeitphase immer negatives Vorzeichen.

- c) Schaltzustand während dem Zeitintervall $t_{1v} \leq t < t_2$ (siehe Bild 4.23)

Im darauffolgenden Zeitabschnitt schließt sich zum Zeitpunkt t_{1v} der obere Schalter $S1_o$, dabei fließt der Strom weiterhin über die obere Freilaufdiode und das Potential am

Ausgang bleibt auf U_{zk} und wird wieder über die Halbbrücke vorgegeben. Es folgt nun ein Vorzeichenwechsel im Strom. Dieser fließt in positiver Richtung über $S1_o$.

d) Schaltzustand während dem Zeitintervall $t_2 \leq t < t_{2v}$ (Bild 4.24)

Das nächste Zeitintervall entspricht wieder der Verriegelungszeit. Zum Zeitpunkt t_2 öffnet der obere Schalter und der positive Strom verändert seinen Pfad und fließt nun über die untere Freilaufdiode. Das Potential am Umrichterausgang wird wieder über die Last bestimmt und springt zurück auf 0 V. Die Stromrichtung bleibt während dieser Phase dieselbe.

e) Schaltzustand während dem Zeitintervall $t_{2v} \leq t \leq T_a$ (Bild 4.25)

Im letzten Zeitabschnitt schließt zum Zeitpunkt t_{2v} der untere Schalter. Das Potential am Ausgang bleibt weiterhin auf 0 V, wird nun aber wieder durch den Spannungszwischenkreisumrichter definiert.

Aus dieser Diskussion wird klar, dass die vorhin getroffene Annahme, durch einen positiven Strommittelwert \bar{i}_1 sind auch die Momentanwerte i_1 positiv, nicht zutrifft. Durch das Ein- und Ausschalten einer ohmsch-induktiven Last kommt es zu einem Stromrippel. Dadurch kann der Strommittelwert zwar positiv sein, aber in den entscheidenden Zeitpunkten ist der Momentanwert i_1 negativ. Die in diesem Abschnitt diskutierten Modelleigenschaften gelten nur für Vorzeichenwechsel außerhalb der Totzeit, da in diesen Zeitbereichen der Strom zwischen IGBT und Freilaufdiode im jeweiligen Zweig der Halbbrücke wechselt.

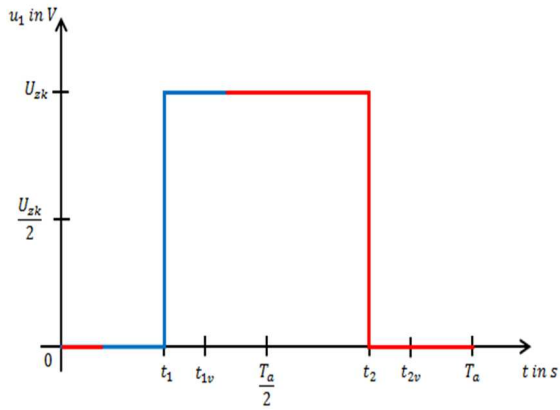


Bild 4.20: PWM (t_v) bei $i_1 > 0$ und $i_1 < 0$

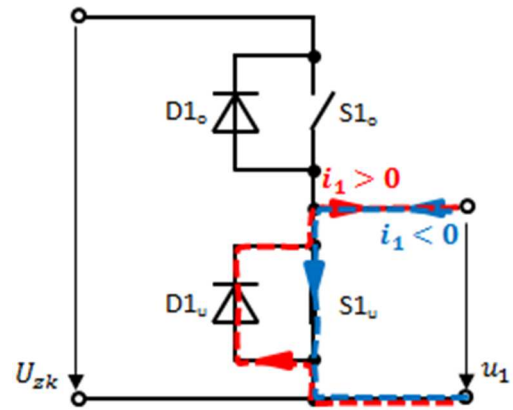


Bild 4.21: $0 \leq t < t_1$

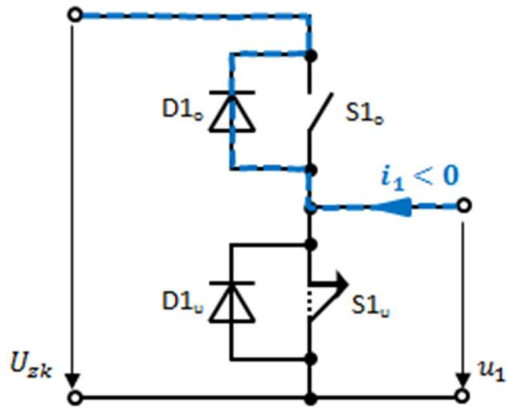


Bild 4.22: $t_1 \leq t < t_{1v}$

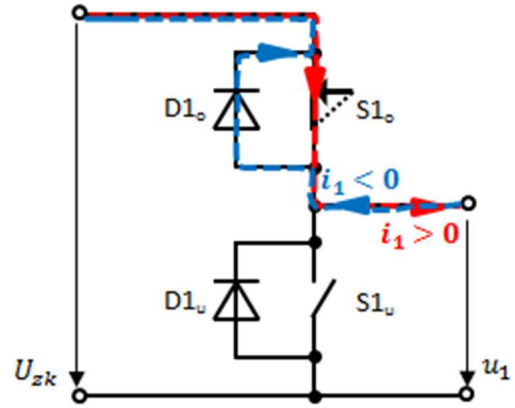


Bild 4.23: $t_{1v} \leq t < t_2$

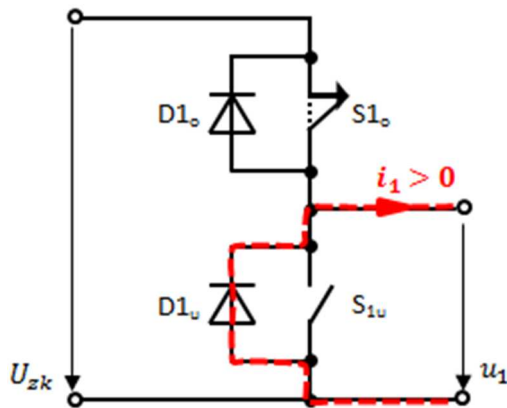


Bild 4.24: $t_2 \leq t < t_{2v}$

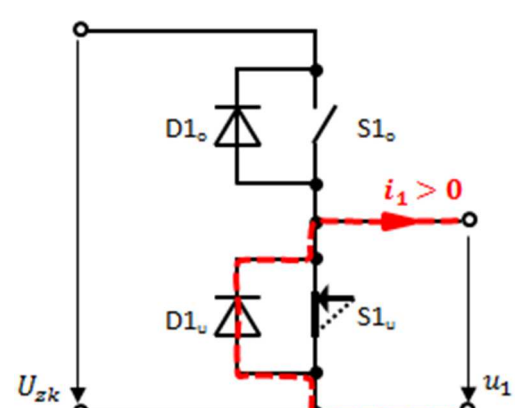


Bild 4.25: $t_{2v} \leq t < T_a$

Bereich 2

Für den Bereich 2 ergeben sich unerwartete Simulationsergebnisse. Bei den beiden anderen Bereichen sind die Mittelwerte im eingeschwungenen Zustand konstant. In diesem speziellen Abschnitt ist dies nicht der Fall, es ergibt sich kein stationärer Zustand, wie im Bild 4.26 ersichtlich wird.

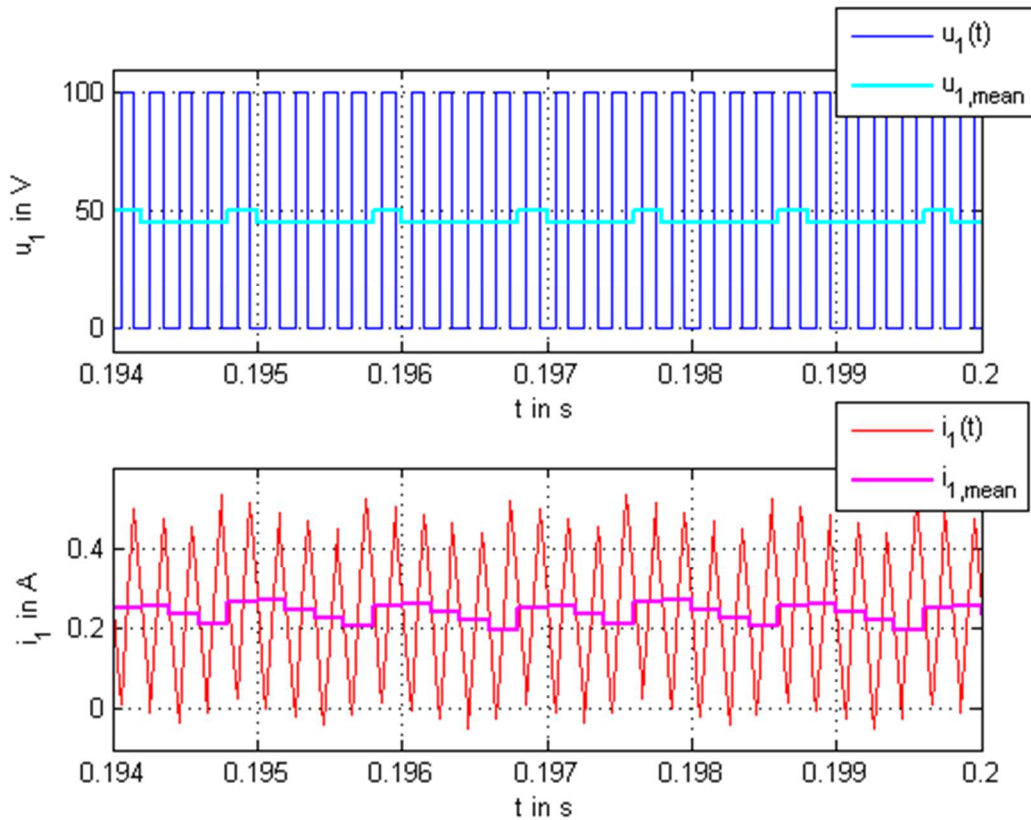


Bild 4.26: Letzten 30 Pulsperioden der Testsimulation für Bereich 2

Da Spannungs- und Strommittelwert nicht konstant sind und dadurch verschiedene Werte in der letzten Pulsperiode möglich sind, lässt sich der spezielle Funktionsverlauf der Umrichternichtlinearität im Bild 4.18 begründen.

Diese Simulationsprobleme treten auf, weil das Stromvorzeichen nur zu den Zeitpunkten t_1 und t_2 ausgewertet wird und während der Verriegelungszeit nicht auf Vorzeichenwechsel im Strom geachtet wird. Diese Problematik wird im Bild 4.27 dargestellt, wobei hier die Mittelwertverläufe nicht um eine Pulsperiode nach rechts verschoben sind. Zur besseren Veranschaulichung sind die in diesem Fall relevanten Zeitpunkte t_1 und t_{1v} eingezeichnet. In der ersten Pulsperiode ist zum Zeitpunkt t_1 der Strom negativ. Mit dem Schaltzustand im Bild 4.9 springt das Potential am Umrichterausgang auf U_{zk} . Dadurch ergibt sich ein Spannungsmittelwert von $\bar{u}_1 = 50 \text{ V}$ über die erste Pulsperiode. Durch diese Anhebung der Ausgangsspannung um 5 V zur vorhergehenden Pulsperiode wird auch der Strom \bar{i}_1 erhöht. In der zweiten Pulsperiode ist zum Zeitpunkt t_1 der Momentanwert des Stromes i_1 positiv. Bild 4.3 beschreibt die Schaltverhältnisse am Pulswechselrichter. In diesem Fall

bleibt die Ausgangsspannung auf 0 V. Die Totzeit ist wieder aktiv und es folgt ein Spannungsabfall von 5 V an der Umrichternichtlinearität. Dies führt zu einer Verkleinerung der Spannung \bar{u}_1 . Der Strommittelwert steigt in der zweiten Pulsperiode etwas an und wird in den nachfolgenden Pulsperioden kleiner. Zum entscheidenden Zeitpunkt t_1 ist i_1 in den folgenden Pulsperioden immer positiv. Der Spannungsmittelwert bleibt auf 45 V. Erst in der letzten dargestellten Pulsperiode ist zum Zeitpunkt t_1 der Strom wieder negativ und die Spannung am Spannungszwischenkreisumrichter steigt an. Damit beginnt der gerade beschriebene Vorgang wieder von vorne.

Das essentielle Problem dabei ist, dass es in der Simulation während der Verriegelungszeit zum Vorzeichenwechsel im Stromverlauf kommt, dies in der Realität jedoch nicht vorkommt. Wenn der Strom in der Totzeit Null wird, bleibt er dort kleben bis zum Zeitpunkt t_{1v} . Dies lässt sich damit begründen, dass während der Verriegelungszeit beide Schalter geöffnet sind. Somit bleiben für den Stromfluss die beiden Freilaufdioden übrig. Hier führt die untere Freilaufdiode ($D1_u$) den positiven Strom i_1 . Dieser wird während der Totzeit null. Für einen Vorzeichenwechsel und das Weiterfließen über die obere Freilaufdiode ($D1_o$) müsste u_1 auf U_{zk} springen bzw. die Gegenspannung U_g größer als die Zwischenkreisspannung sein, da in diesem Fall die Ausgangsspannung u_1 am Pulswechselrichter von der Last vorgegeben wird (während der Totzeit sind $S1_u$ und $S1_o$ geöffnet). Die Spannung an der Induktivität beträgt $u_L = 0$ V bzw. ist die Gegenspannung $U_g < U_{zk}$. Der Spannungsabfall am Widerstand ist ebenfalls null. Damit wird die Gleichung $u_1 = u_R + u_L + U_g$ zu $u_1 = U_g$. Dieser Vorgang wird als Stromlücken (engl. zero clamping effect) bezeichnet. Im nächsten Abschnitt wird nun ein Modell vorgestellt in dem das Stromlücken berücksichtigt wird.

Bisher wurden die Verhältnisse für positive Ströme erläutert. Für negative Strommittelwerte ist das Problem ähnlich, nur treten die Vorzeichenwechsel während der Verriegelungszeit nach dem Schaltzeitpunkt t_2 auf. In diesem Fall bleibt der Strom, wenn er in diesem Zeitabschnitt null wird, bis zum Zeitpunkt t_{2v} auf null.

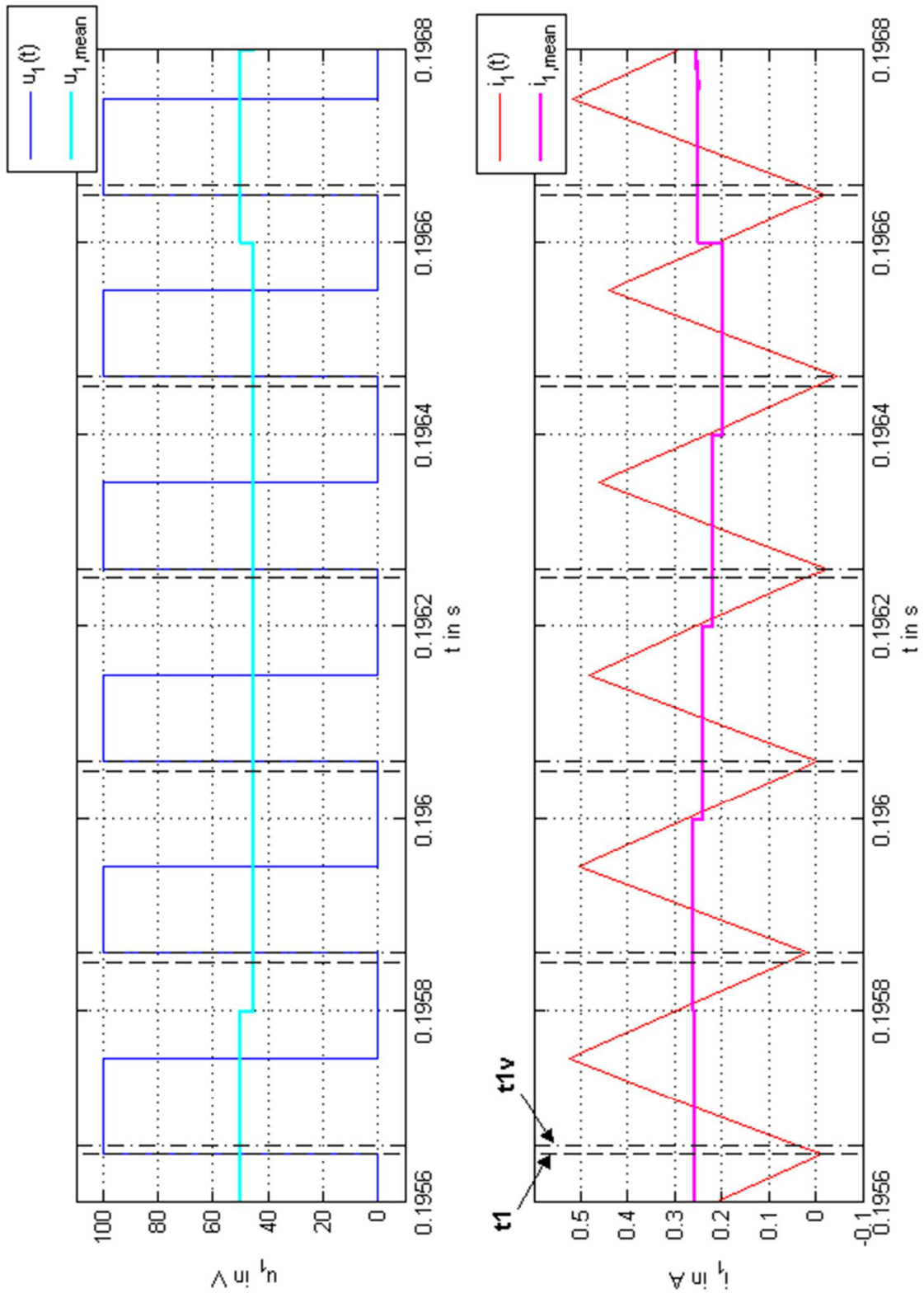


Bild 4.27: Darstellung einer Simulation zur Erklärung des Modellverhaltens im Bereich 2

4.3 Modellmodifizierung zur Berücksichtigung des Stromlückens

Das MATLAB/Simulink-Modell ändert sich im Blick auf Bild 4.15 nur im Block „PWR“. Der Koppelplan für den Pulswechselrichter ist im Bild 4.28 abgebildet.

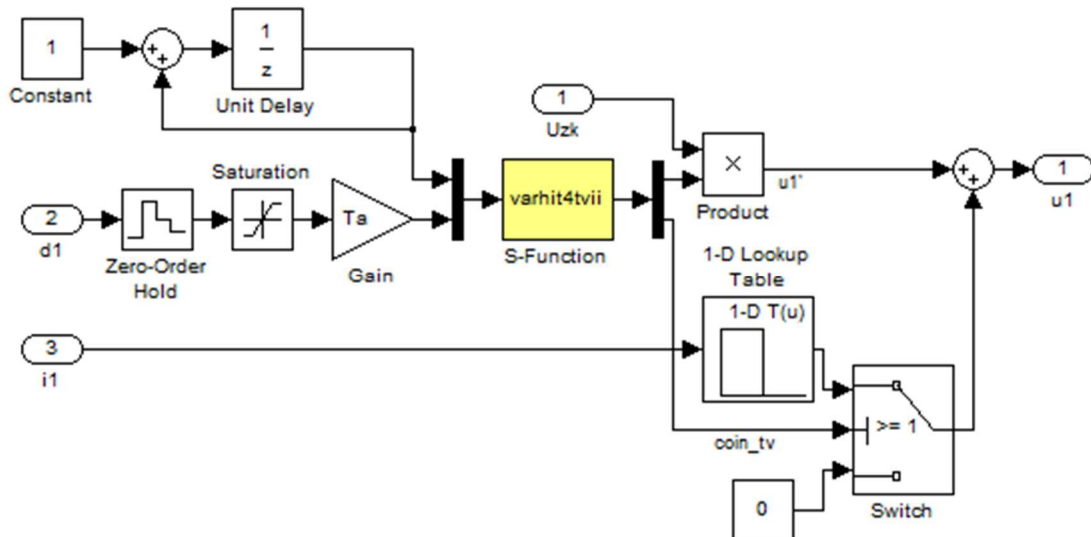


Bild 4.28: Modifiziertes Modell des Pulswechselrichters im Block „PWR“

4.3.1 Änderungen in der S-Function

Eingangs- und Ausgangssignale der S-Function

Der zeitliche Verlauf des Stromes ist in der Modellmodifizierung nicht mehr Eingangsgröße für die S-Function. Dieses Modell verfügt über einen zusätzlichen Ausgang und dieser gibt an, wann die Verriegelungszeit aktiv ist. Dieses Signal ist während der Pulsperiode in den Zeitabschnitten der Verriegelungszeit 1. Sonst hat es den Wert null. Dieses zusätzliche Ausgangssignal wird für das Schalten des Switch-Blocks benötigt.

Da die Vorzeichenüberprüfung des Stromes nicht mehr in der S-Function erfolgt, wird von dieser immer der gleiche Puls ausgegeben. Zum Zeitpunkt t_{1v} springt das Signal auf eins und zum Zeitpunkt t_2 geht es zurück auf null. Daraus folgt, dass während der Verriegelungszeit t_v der erste Ausgang der S-Function immer auf null ist.

Simulationsablauf der S-Function

Der Simulationsablauf ist ähnlich zum vorhergehenden Modellierungsschritt mit Berücksichtigung der Verriegelungszeit in Tabelle 2. Die Spalte mit u_3 (Stromeingang) fällt weg bzw. in der Spalte für sys kommt in den Zeilen für Output ein zusätzlicher Wert für das zweite Ausgangssignal hinzu. Der Wert ist zum Zeitpunkt t_1 (0,00005s, 0,00025 und 0,00045) bzw. für t_2 (0,000015s, 0,00035 und 0,00055) 1. Für alle anderen Zeitpunkte

ist der zweite Ausgang auf null.

Die hierfür verwendete MATLAB-Funktion ist „varhit4tvii.m“. Deren Quellcode befindet sich im Anhang A.3.

4.3.2 Ausgangsspannung u_1

Es sei nochmals festgehalten, dass der erste Ausgang der S-Funktion unabhängig vom Stromvorzeichen immer den vorher erklärten Puls produziert. Dieser wird, wie im Bild 4.28 dargestellt, mit der Zwischenkreisspannung multipliziert. Dies ergibt den Spannungspuls u_1' .

Es stellt sich nun die Frage welchen Spannungswert u_1 in der Verriegelungszeit annimmt? Da die Spannung u_1' während der Verriegelungszeit immer null ist, kann mit dem „Switch“ je nach Stromvorzeichen eine gewünschte Spannung zu u_1' addiert werden. Das zusätzliche Ausgangssignal wird zur Betätigung des „Switch“ verwendet. Wenn die Verriegelungszeit aktiv ist, wird der obere Eingang auf den Ausgang durchgeschaltet und zur Spannung u_1' wird eine Spannung in Abhängigkeit des Stromes i_1 (Lookup-Table) addiert. Mit dieser Modellstruktur wird erreicht, dass bei einem Stromnulldurchgang der Strom i_1 auf null bzw. sehr klein bleibt und erst nach dem Ende der Totzeit einen von null verschiedenen Wert annehmen kann. Die Kennlinie für die Ausgangsspannung am Pulswechselrichter während der Verriegelungszeit ist in der Lookup-Tabelle abgelegt (siehe Bild 4.29).

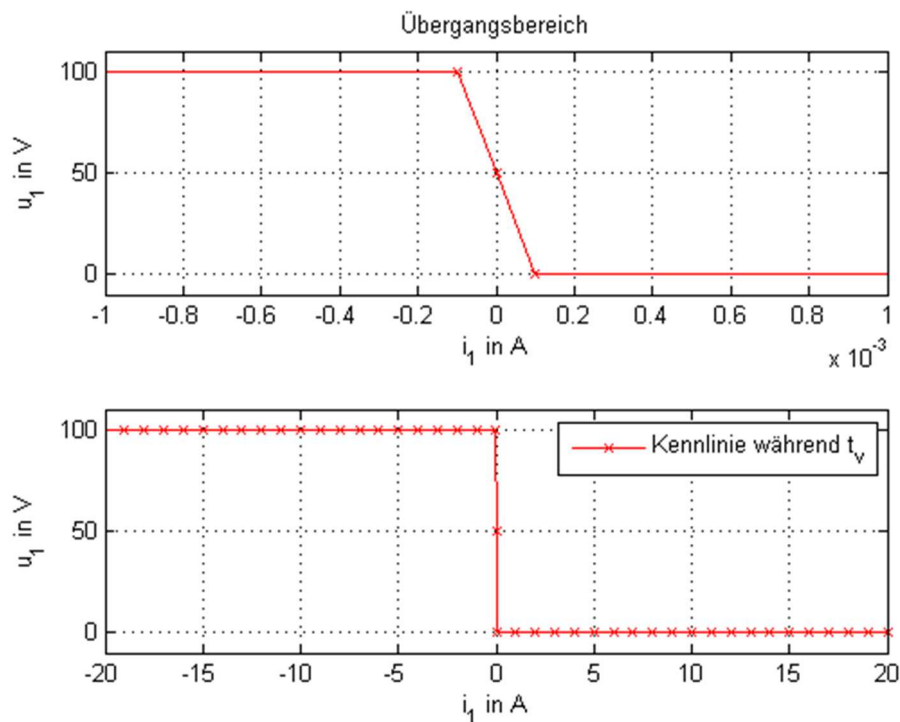


Bild 4.29: Kennlinie für die Lookup-Tabelle mit $R_{rv} = 500 \text{ k}\Omega$

Der Sprung von der Zwischenkreisspannung $U_{zk} = 100 \text{ V}$ auf 0 V wird durch eine Widerstandsgerade ersetzt. In diesem Fall beträgt der Widerstand $R_{tv} = 500 \text{ k}\Omega$. Dadurch ergibt sich in Abhängigkeit von der U_{zk} im Strombereich $-1e-4 \leq i_1 \leq 1e-4$ die lineare Widerstandsgerade.

Variation von R_{tv}

Um den Einfluss des Widerstandes R_{tv} auf das Simulationsverhalten zu untersuchen wurde mit unterschiedlichen Widerständen die Kennlinie der Umrichternichtlinearität simuliert. Dabei wurden die Zwischenkreisspannung mit $U_{zk} = 100 \text{ V}$ bzw. die Verriegelungszeit mit $t_v = 2 \mu\text{s}$ festgelegt. Die Simulationsergebnisse der Variation von R_{tv} sind im Bild 4.30 dargestellt. Daraus ist ersichtlich, dass für die Widerstände $R_{tv} = \{50 \text{ k}\Omega, 500 \text{ k}\Omega, 5 \text{ M}\Omega\}$ die Simulationsergebnisse sehr gut übereinstimmen. Damit lässt sich nachvollziehen, warum der Wert mit $R_{tv} = 500 \text{ k}\Omega$ für die weiteren Untersuchungen in dieser Arbeit gewählt wurde.

Es handelt sich hierbei um ein System, in dem der Eigenwert stark verändert wird, da durch die Kennlinie im System der Widerstand R beeinflusst wird. Für Ströme $|i_1| > 1e-4$ ist der gesamte Widerstand im System $R_{ges} = 0,3 \Omega$. Damit ergibt sich eine Zeitkonstante von $\tau_R \approx 0,03\text{s}$. Für sehr kleine Ströme $|i_1| < 1e-4$ steigt der gesamte Widerstand auf $R_{ges} = 500 \text{ k}\Omega + 0,3 \Omega \approx 500 \text{ k}\Omega = R_{tv}$ an. Daraus ergibt sich nun die Zeitkonstante $\tau_{Rtv} \approx 0,2 \mu\text{s}$. Abhängig vom Strom besitzt das mathematische Modell zwei unterschiedliche Zeitkonstanten die sehr unterschiedlich sind. Für kleine Ströme ist die Zeitkonstante um den Faktor 10000 kleiner. Der Solver „ode45“ berücksichtigt dies in der Simulation. Dadurch wird die Schrittweite kleiner, um die Funktionsverläufe mit der geforderten Genauigkeit nachbilden zu können.

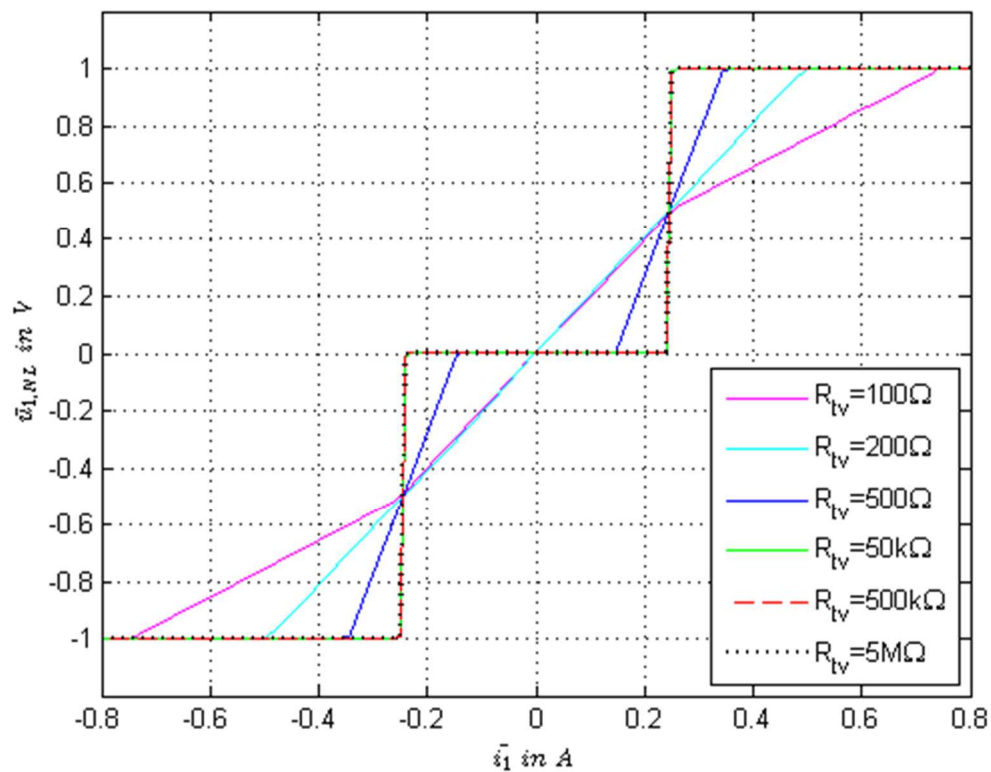


Bild 4.30: Variation von $R_{tv} = \{100 \Omega, 200 \Omega, 500 \Omega, 50 \text{ k}\Omega, 500 \text{ k}\Omega, 5 \text{ M}\Omega\}$

4.3.3 Testsimulation für die Umrichternichtlinearität

Durch das modifizierte MATLAB/Simulink-Modell, mit Berücksichtigung der Verriegelungszeit, in dem auch das Stromlücken modelliert wird, erhält man für die Simulation der Umrichternichtlinearität den im Bild 4.31 dargestellten Verlauf. Ein Vergleich mit Bild 4.18 zeigt, dass dieses Modell nun einwandfrei funktioniert und sinnvolle Ergebnisse liefert.

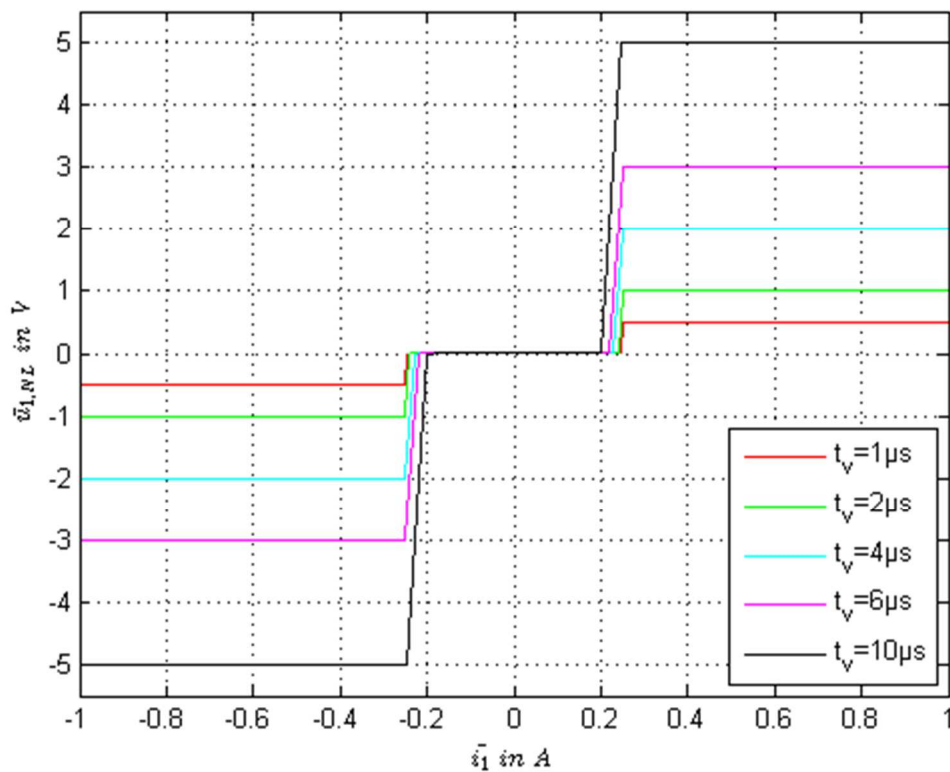


Bild 4.31: Ausschnitt der Umrichternichtlinearität mit modifiziertem Modell

4.4 Validierung des Modells mit Verriegelungszeit

Die Simulationen mit Berücksichtigung der Verriegelungszeit erscheinen im Bild 4.32 als grobe Näherung der Umrichternichtlinearität. Der nichtlineare Funktionsverlauf der Messung ist in der simulierten Umrichternichtlinearität nicht vorhanden. Die Kurve mit $t_v = 4 \mu\text{s}$ ist im Bereich von $\pm 10 \text{ A}$ bis $\pm 20 \text{ A}$ akzeptabel. Die Totzeit t_v beträgt aber ca. $2 \mu\text{s}$. Der Funktionsverlauf für $t_v = 2 \mu\text{s}$ ist nur in einem sehr schmalen Strombereich ($0,4 \text{ A} < \bar{i}_1 < 1 \text{ A}$) eine passable Approximation für die Messung.

Dieses Modell ist sehr ungenau im Strombereich $-5 \text{ A} < \bar{i}_1 < 5 \text{ A}$. Die Kurve im angesprochenen Bereich der Umrichternichtlinearität kann mit diesem mathematischen Modell noch nicht zufriedenstellend nachgebildet werden. Es bedarf weiterer Untersuchungen, um Modellverbesserungen durchzuführen. Im nächsten Kapitel widmen wir uns den bis jetzt vernachlässigten Spannungsabfällen an den leistungselektronischen Bauelementen und deren Einfluss auf die Umrichternichtlinearität.

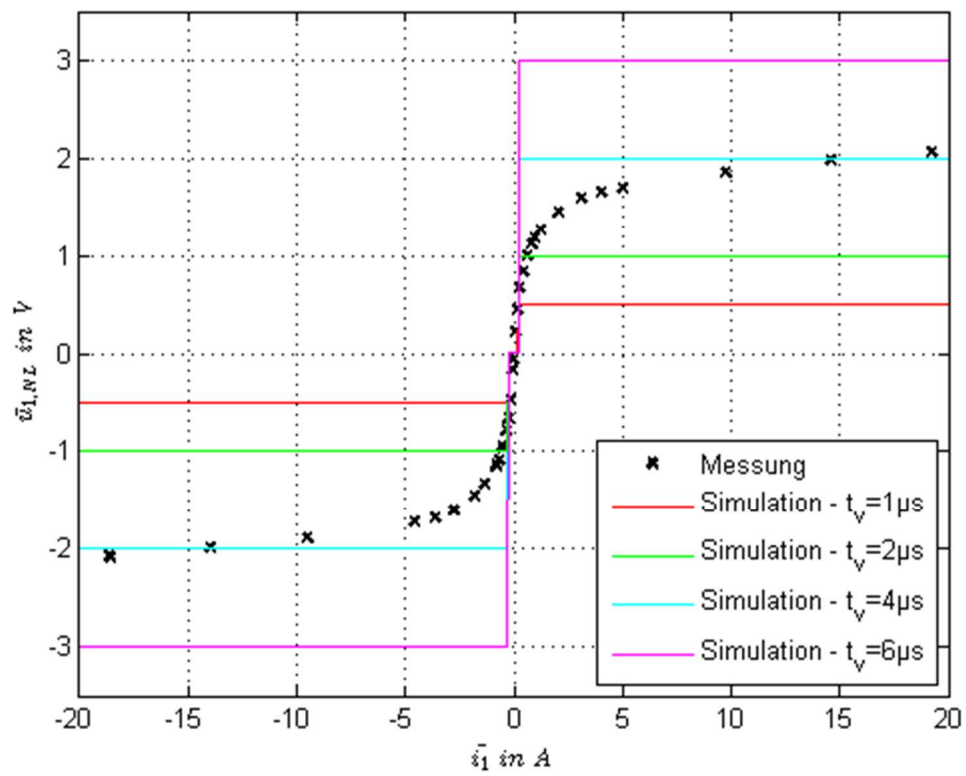


Bild 4.32: Vergleich zwischen Messung und Simulation der Umrichternichtlinearität

5 Modell mit Spannungsabfällen an IGBT und Freilaufdiode

Dieses Kapitel befasst sich mit den Auswirkungen der Spannungsabfälle an der Freilaufdiode und dem IGBT. Zu Beginn werden die Spannungsabfälle im Ausgangssignal des Spannungszwischenkreisumrichters für positive und negative Ströme untersucht. Danach folgt die Ermittlung der konkreten Kennlinien für IGBT und der Freilaufdiode. Den Abschluss bilden die Implementierung der Spannungsabfälle im MATLAB/Simulink-Modell sowie die Diskussion der generierten Simulationsergebnisse.

5.1 Spannungsabfälle an den elektronischen Bauelementen

Bisher wurden die Spannungsabfälle am IGBT und an der Freilaufdiode vernachlässigt. Da die bisher beschriebenen Modelle noch Ungenauigkeiten aufweisen, werden diese Spannungsabfälle im MATLAB/Simulink-Modell nun berücksichtigt. Wie schon vorher diskutiert, kann der Strom über vier mögliche Pfade fließen. Dabei kommt es je nach Stromhöhe zu unterschiedlich großen Spannungsabfällen an den elektronischen Bauelementen.

Wie oben bei den Betrachtungen zur Verriegelungszeit, sind die Spannungsabfälle an Freilaufdiode und IGBT ebenfalls vom Stromvorzeichen abhängig. Es wird angenommen, dass die beiden Freilaufdioden (D_{1u} und D_{1o}) in der Halbbrücke ident sind und mit einer Kennlinie beschrieben werden. Selbiges trifft auch auf die IGBT's zu. Für die Erklärung der Verhältnisse an der Halbbrücke in diesem Fall, wird die Pulsperiode wieder in die fünf bekannten Zeitintervalle aufgeteilt.

- a) $0 \leq t < t_1$
- b) $t_1 \leq t < t_{1v}$
- c) $t_{1v} \leq t < t_2$
- d) $t_2 \leq t < t_{2v}$
- e) $t_{2v} \leq t < T_a$

In den beiden folgenden Abschnitten werden die Auswirkungen der Spannungsabfälle an den elektronischen Bauelementen je nach Stromvorzeichen untersucht.

5.1.1 Strom positiv

Im Bild 5.1 bis Bild 5.6 sind die Vorgänge mit Berücksichtigung der Spannungsabfälle in einer Pulsperiode für $i_1 > 0$ dargestellt. Der Funktionsverlauf der Spannung u_1 am Ausgang des Spannungszwischenkreisumrichters ist im Bild 5.1 abgebildet. Die Schaltzustände an der Halbbrücke sind im Bild 5.2 bis Bild 5.6 zu sehen. Jede dieser Abbildungen beschreibt einen Ausschnitt aus der Pulsperiode.

a) Schaltzustand während dem Zeitintervall $0 \leq t < t_1$ (siehe Bild 5.2)

Der Schalter S_{1u} ist geschlossen und durch die Bedingung $S_{1u} = \overline{S_{1o}}$ ist S_{1o} offen. Ein Strom in positiver Richtung fließt über die untere Freilaufdiode (D_{1u}) und in diesem Fall fällt an der Freilaufdiode eine Spannung u_D in Flussrichtung ab. Dieser Spannungsabfall ist auch abhängig von der Größe des Stromes. Damit gilt $u_1 = -u_D(i_1)$.

b) Schaltzustand während dem Zeitintervall $t_1 \leq t < t_{1v}$ (siehe Bild 5.3)

Zu Beginn dieses Zeitintervalls öffnet S_{1u} und die Verriegelungszeit ist wirksam. Der Strom fließt weiterhin über die untere Freilaufdiode (D_{1u}) und wie vorhin ist $u_1 = -u_D(i_1)$.

c) Schaltzustand während dem Zeitintervall $t_{1v} \leq t < t_2$ (siehe Bild 5.4)

Zum Zeitpunkt t_{1v} ist der obere Schalter (S_{1o}) geschlossen, der Strom in positiver Richtung wechselt den Pfad und fließt nun über den oberen Schalter. Auch am IGBT tritt ein Spannungsabfall u_S in Stromrichtung auf. Damit wird die Spannung am Umrichterausgang zu $u_1 = U_{zk} - u_S(i_1)$. Im Vergleich zu den beiden obigen Modellierungen ist die Spannung u_1 in diesem Zeitabschnitt kleiner als die Zwischenkreisspannung.

d) Schaltzustand während dem Zeitintervall $t_2 \leq t < t_{2v}$ (siehe Bild 5.5)

Am Anfang dieses Zeitintervalls zum Zeitpunkt t_{2v} öffnet der obere Schalter (S_{1o}) und der Strom wechselt den Pfad auf die untere Freilaufdiode (D_{1u}). Damit ist die Spannung $u_1 = -u_D(i_1)$.

e) Schaltzustand während dem Zeitintervall $t_{2v} \leq t < T_a$ (siehe Bild 5.6)

Der untere Schalter (S_{1u}) ist zum Zeitpunkt t_{2v} geschlossen und der Strom fließt wie zuvor über die untere Freilaufdiode. Die Spannung am Ausgang des Pulswechselrichters ist wieder $u_1 = -u_D(i_1)$.

Fazit aus den Betrachtungen zum positiven Strom

Im Bezug zur vorhergehenden Analyse der Verriegelungszeit im Bild 4.1 wird der PWM-Puls erneut reduziert. Die Pulsbreite ist dieselbe wie in den Betrachtungen zur Totzeit, aber durch die Berücksichtigung der Spannungsabfälle an IGBT und der Freilaufdiode wird die Höhe des Pulses verkleinert.

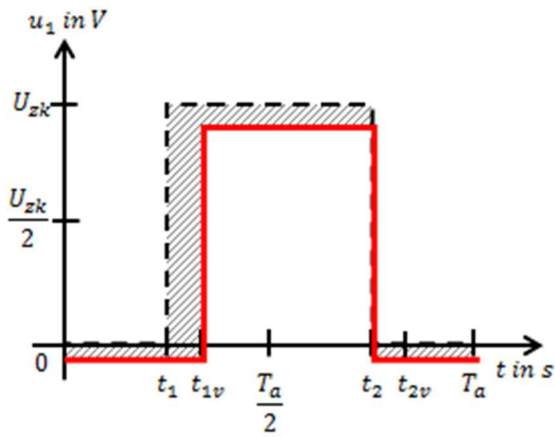


Bild 5.1: PWM mit Spannungsabfällen $i_1 > 0$

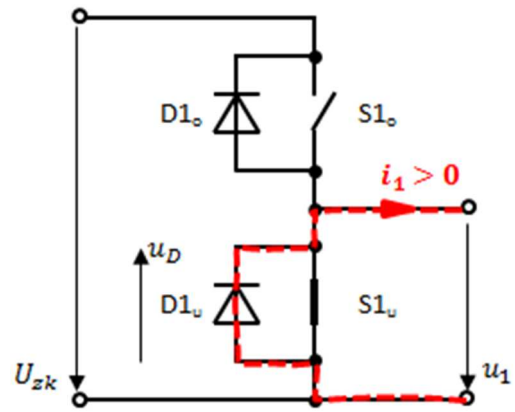


Bild 5.2: $0 \leq t < t_1$

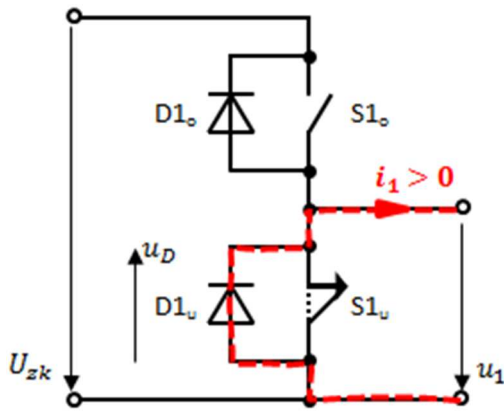


Bild 5.3: $t_1 \leq t < t_{1v}$

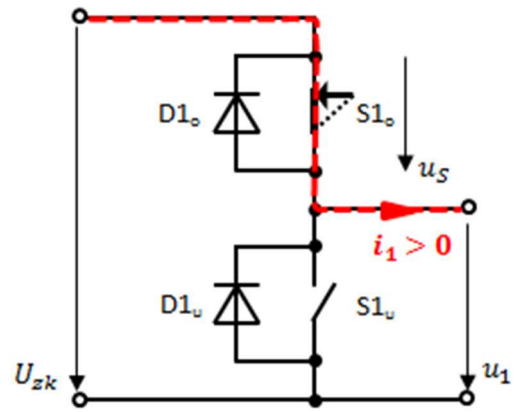


Bild 5.4: $t_{1v} \leq t < t_2$

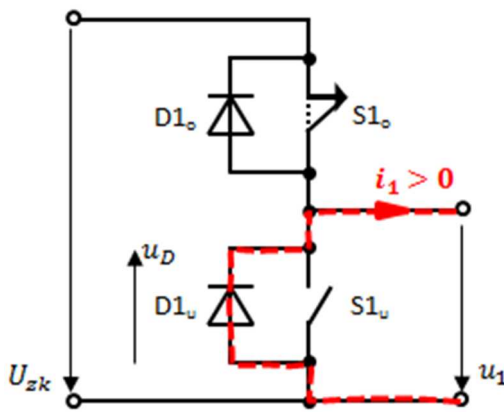


Bild 5.5: $t_2 \leq t < t_{2v}$

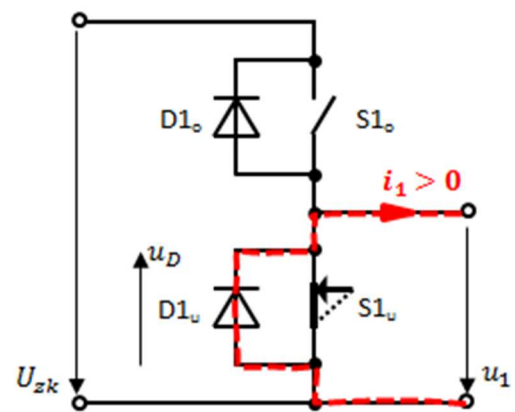


Bild 5.6: $t_{2v} \leq t < T_a$

5.1.2 Strom negativ

Der Spannungspuls für den negativen Strom ist im Bild 5.7 abgebildet. Die Schaltzustände für die fünf Zeitintervalle sind im Bild 5.8 bis Bild 5.12 dargestellt. Nun folgt die Erklärung, wie der PWM-Puls bei negativem Strom i_1 zustande kommt.

a) Schaltzustand während dem Zeitintervall $0 \leq t < t_1$ (siehe Bild 5.8)

In diesem Zeitintervall ist der untere Schalter ($S1_u$) geschlossen und der negative Strom i_1 fließt über $S1_u$. Auch beim IGBT-Modul fällt ja nach Stromhöhe eine Spannung u_S ab und damit wird $u_1 = u_S(i_1)$.

b) Schaltzustand während dem Zeitintervall $t_1 \leq t < t_{1v}$ (siehe Bild 5.9)

Zum Zeitpunkt t_1 öffnet der untere Schalter ($S1_u$) und der negative Strom i_1 wechselt seinen Pfad auf die obere Freilaufdiode ($D1_o$). Die Spannung am Ausgang des Spannungszwischenkreisumrichters beträgt nun $u_1 = U_{zk} + u_D(i_1)$.

c) Schaltzustand während dem Zeitintervall $t_{1v} \leq t < t_2$ (siehe Bild 5.10)

Am Anfang des Zeitabschnitts t_{1v} ist der obere Schalter ($S1_o$) geschlossen und der Strom fließt weiterhin über die obere Freilaufdiode ($D1_o$). Die Spannung beträgt $u_1 = U_{zk} + u_D(i_1)$.

d) Schaltzustand während dem Zeitintervall $t_2 \leq t < t_{2v}$ (siehe Bild 5.11)

Zu Beginn dieses Intervalls t_2 öffnet der obere Schalter und die Last ist für die Dauer der Verriegelungszeit vom Spannungszwischenkreisumrichter getrennt. Der Pfad des Stromes bleibt gleich und er fließt weiterhin über die obere Freilaufdiode ($D1_o$). Die Spannung am Umrichterausgang bleibt unverändert $u_1 = U_{zk} + u_D(i_1)$.

e) Schaltzustand während dem Zeitintervall $t_{2v} \leq t < T_a$ (siehe Bild 5.12)

Zum Zeitpunkt t_{2v} ist der untere Schalter ($S1_u$) geschlossen und der negative Strom i_1 wechselt den Pfad von der oberen Freilaufdiode ($D1_o$) zum unteren IGBT ($S1_u$). Damit wird die Spannung wieder zu $u_1 = u_S(i_1)$.

Fazit aus den Betrachtungen zum negativen Strom

Der Vergleich zwischen dem PWM-Puls aus dem Modell mit der Verriegelungszeit im Bild 4.7 und dem Modell mit zusätzlicher Berücksichtigung der Spannungsabfälle in Bild 5.7 zeigt, dass die Pulsbreite unverändert ist, aber die Höhe des Pulses zugenommen hat.

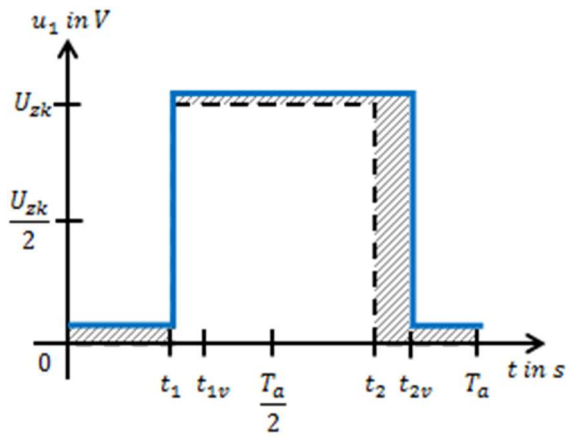


Bild 5.7: PWM mit Spannungsabfällen $i_1 < 0$

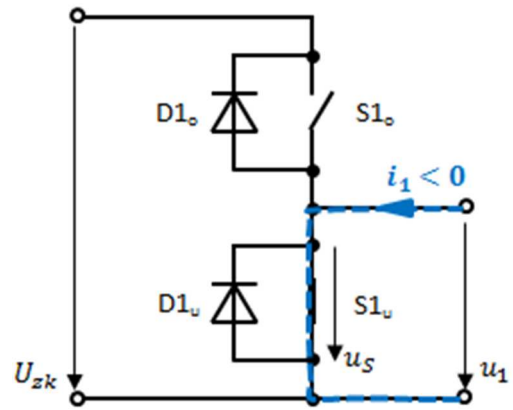


Bild 5.8: $0 \leq t < t_1$

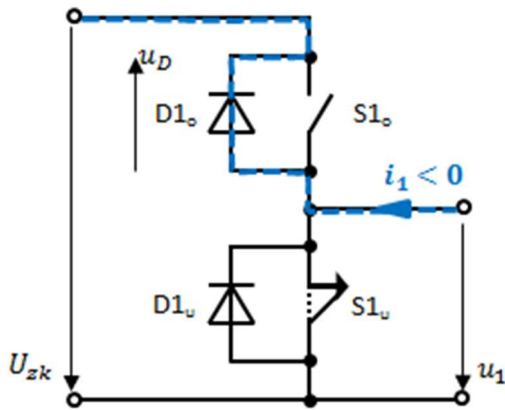


Bild 5.9: $t_1 \leq t < t_{1v}$

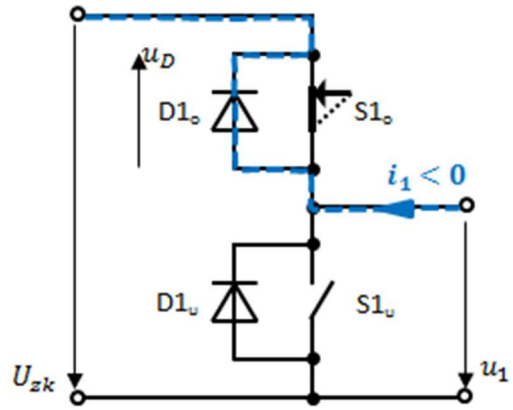


Bild 5.10: $t_{1v} \leq t < t_2$

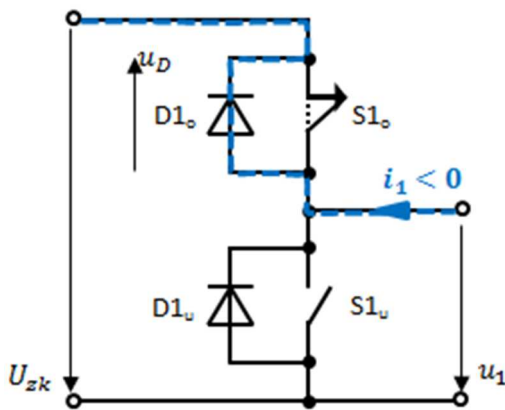


Bild 5.11: $t_2 \leq t < t_{2v}$

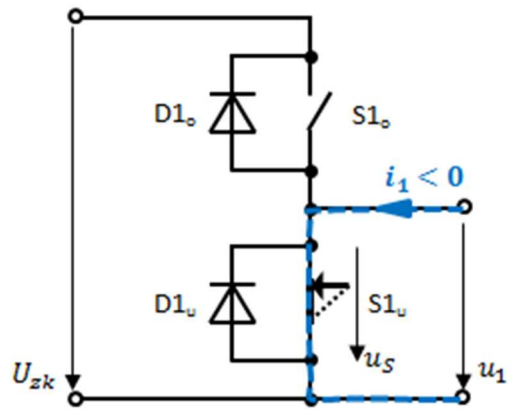


Bild 5.12: $t_{2v} \leq t < T_a$

5.1.3 Strom null

Die beiden vorhergehenden Diskussionen haben Fälle behandelt, in denen der Strommittelwert \bar{i}_1 und auch alle Momentanwerte i_1 positiv oder negativ waren. In folgender Abhandlung geht es um den speziellen Fall, in dem der Strommittelwert zwar positiv bzw. negativ ist, dies aber für die zugehörigen Zeitverläufe des Stromes nicht zutrifft. Im Bild 5.13 ist als Beispiel für diesen Fall der PWM-Puls mit Strommittelwert $\bar{i}_1 = 0$ dargestellt.

Im folgenden Abschnitt werden die Nulldurchgänge des Stromes diskutiert, die in den Zeitintervallen a), c), und e) auftreten.

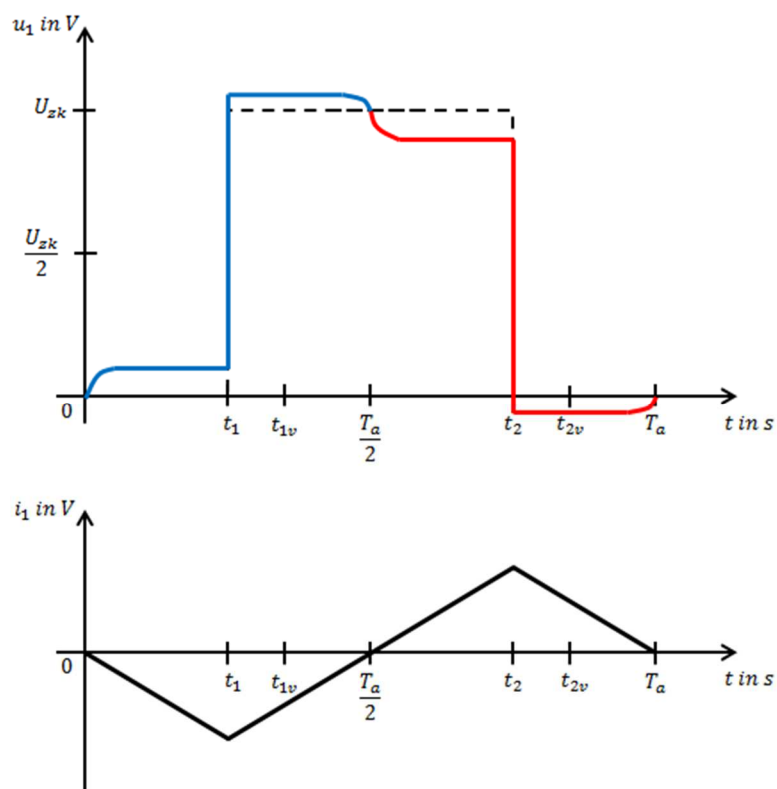


Bild 5.13: PWM-Puls bei $\bar{i}_1 = 0$ unter Berücksichtigung der Spannungsabfälle

Im ersten Zeitintervall $0 \leq t < t_1$ ist der Strom i_1 am Anfang null und nimmt dann negative Werte an. Der Strompfad führt dabei über $S1_u$ und $u_1 = u_S(i_1)$ (siehe Bild 5.8). Das nächste Zeitintervall $t_1 \leq t < t_{1v}$ entspricht Verriegelungszeit t_v . Zum Zeitpunkt t_1 öffnet der stromführende IGBT $S1_u$ und der Strom wechselt seinen Pfad auf die obere Freilaufdiode $D1_o$. Die Ausgangsspannung ist $u_1 = U_{zk} + u_D(i_1)$ (siehe Bild 5.9). Im dritten Zeitabschnitt $t_{1v} \leq t < t_2$ ist der Strom i_1 im Anfangsbereich negativ und u_1 ergibt sich wie vorhin (siehe Bild 5.10). Zum Zeitpunkt $T_a/2$ hat der Strom einen Nulldurchgang und der Strom i_1 wird positiv. Dabei wechselt i_1 seinen Weg von $D1_o$ zu $S1_o$ und $u_1 = U_{zk} - u_S(i_1)$ (siehe Bild 5.4). Das nächste Zeitintervall $t_2 \leq t < t_{2v}$ ist die Totzeit. Zum Zeitpunkt t_2 öffnet der IGBT $S1_o$ und der Strom i_1 wechselt seinen Flusspfad auf die untere

Freilaufdiode $D1_u$. Die Ausgangsspannung des Pulswechselrichters ist $u_1 = -u_D(i_1)$ (siehe Bild 5.5). Für den letzten Zeitabschnitt $t_{2v} \leq t < T_a$ ist $u_1 = -u_D(i_1)$ und das Schließen des unteren IGBT hat keinen Einfluss auf die Ausgangsspannung des Umrichters (siehe Bild 5.6).

Die Zeitintervalle b) und d) entsprechen den Verriegelungszeiten. Dabei tritt wieder das Stromlücken auf. Im Zeitintervall der Verriegelungszeit ist die Spannung u_1 vom Zwischenkreis abgetrennt. Die Ausgangsspannung des Spannungszwischenkreisumrichters wird in diesem Fall von der Last bestimmt. Ein Stromnulldurchgang ist in diesem Fall nicht möglich. Wenn der Strom während der Verriegelungszeit null wird, bleibt er bis zu deren Ende auf diesem Wert. Dieser Effekt wurde schon im vorhergehenden Kapitel ausführlich behandelt. Die dort gewonnenen Erkenntnisse beschreiben in gleicher Weise die in diesem Abschnitt auftretenden Phänomene.

5.2 Kennlinien von IGBT und Freilaufdiode

Die elektronischen Bauelemente besitzen nichtlineare Kennlinien. Die Kennlinie wird in einen Durchlassbereich und einen Sperrbereich unterteilt. Die klassische Diodenkennlinie ist eine logarithmische Funktion. Der IGBT hat aufgrund seiner Halbleiterstruktur im Anfangs- und Endbereich einen anderen Kennlinienverlauf.

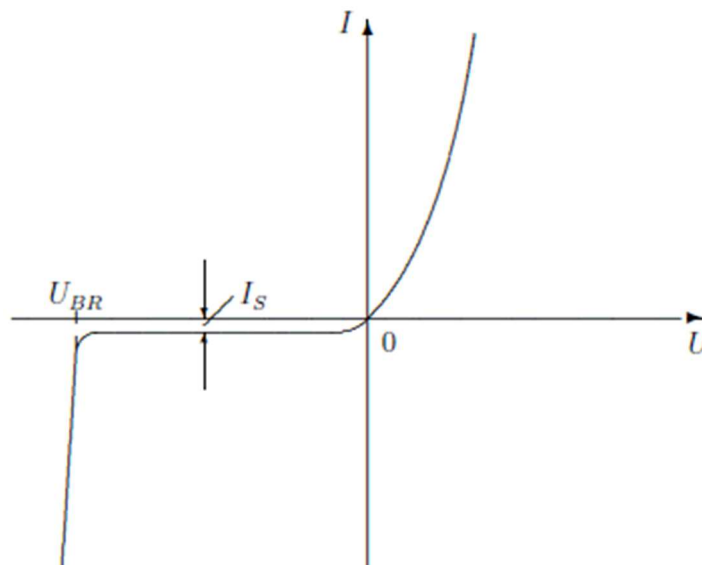


Bild 5.14: Dioden-Kennlinie aus [11]

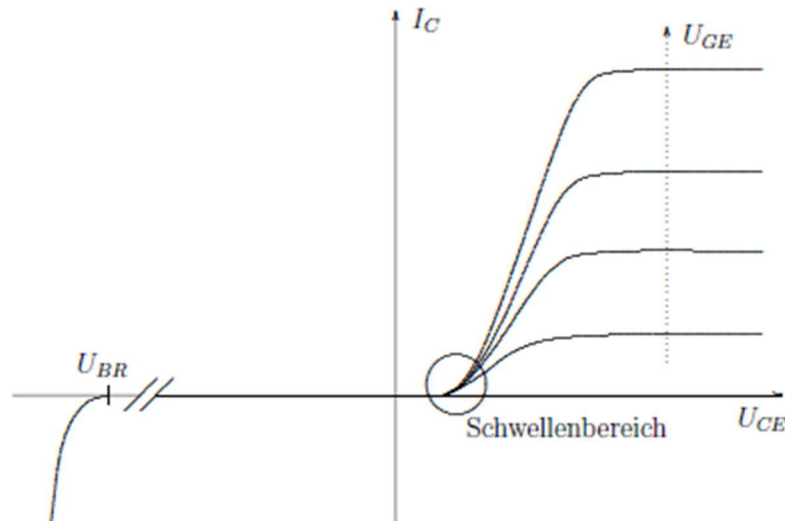


Bild 5.15: Ausgangs-Kennlinienfeld eines IGBT aus [11]

Für den verwendeten Spannungszwischenkreisumrichter gibt es für die beiden elektronischen Bauteile Datenblätter die zur Verfügung stehen. Im interessierenden Strombereich ± 20 A sind die Kennlinien nicht dokumentiert. Andere Datenblätter mit vergleichbaren Bauelementen zeigen im gewünschten Strombereich auch keine Aussagekräftigen Kennlinienverläufe. Die Kennlinien für Freilaufdiode und IGBT werden aus den zeitlichen Messungen mit dem Leistungsmessgerät N5000 bestimmt. Dazu kommen wir noch später (siehe 5.2.2 bzw. 5.2.3). Zuvor soll nun geklärt werden, ob die Berücksichtigung des Sperrverhaltens notwendig ist.

5.2.1 Diskussion über die Berücksichtigung des Sperrverhaltens von IGBT und Freilaufdiode

Dabei wird die vorhergehende Diskussion der Schaltzustände für positiven und negativen Strom auf drei grundlegende Schaltzustände reduziert. Der erste Schaltzustand beschreibt die Halbbrücke mit geschlossenem Schalter S_{1u} . In diesem Fall lassen sich Bild 5.2 und Bild 5.6 mit $i_1 > 0$ bzw. Bild 5.8 und Bild 5.12 mit $i_1 < 0$ im Bild 5.16 zusammenfassen.

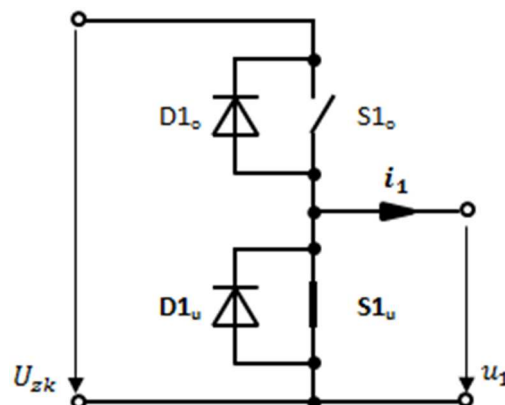


Bild 5.16: Schaltzustand 1

Hier ist der untere Zweig der Halbbrücke mit $D1_u$ und $S1_u$ wirksam. Aus der Analyse in 5.1.1 und 5.1.2 lässt sich beweisen, dass je nach Stromvorzeichen die Freilaufdiode oder der IGBT den Strom führt. Im Bild 5.17 sind die beiden Kennlinien der leistungselektronischen Bauelemente abgebildet.

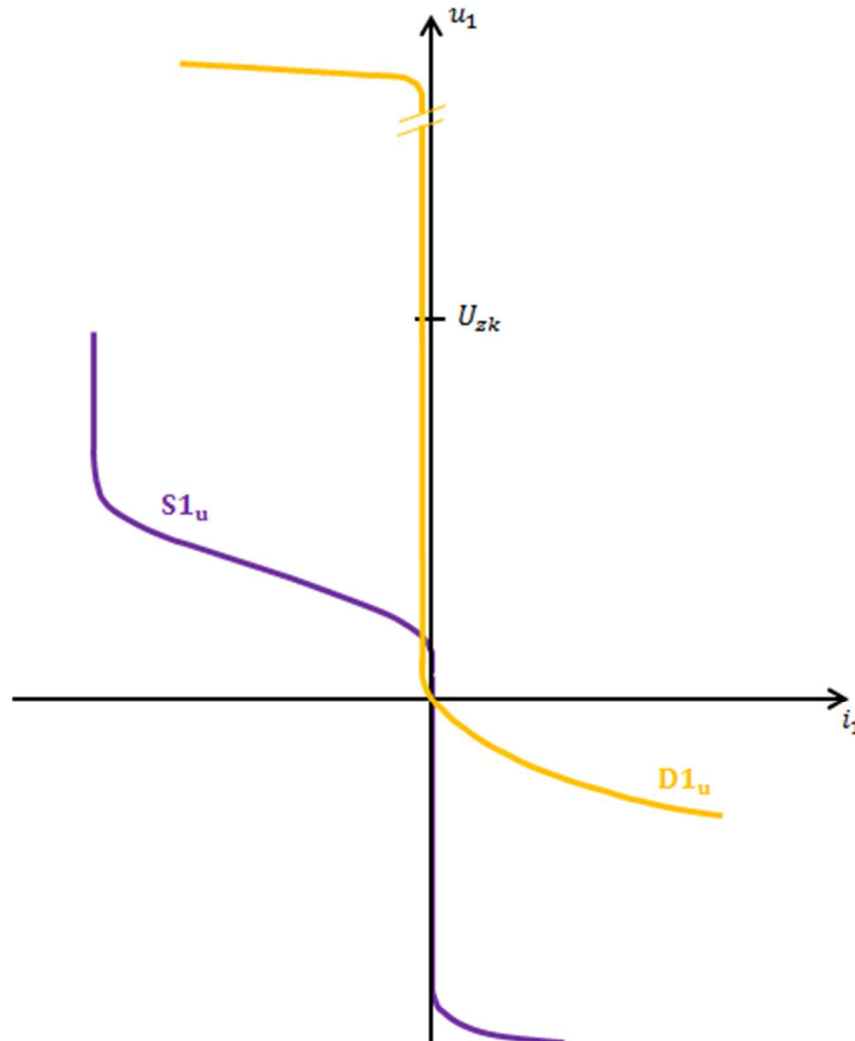


Bild 5.17: Kennlinien für $D1_u$ und $S1_u$ im Schaltzustand 1

Für $i_1 > 0$ fließt der Strom über $D1_u$. An der Freilaufdiode fällt dabei die Spannung $u_D(i_1)$ ab. In diesem Fall fließt auch ein sehr kleiner Sperrstrom über den IGBT $S1_u$. Dieser kann aber für die weiteren Betrachtungen vernachlässigt werden, da er im Verhältnis zu i_1 sehr klein ist. Der Fall $i_1 < 0$ verhält sich umgekehrt. Der elektronische Schalter $S1_u$ führt den Strom und über die Freilaufdiode fließt ein sehr geringer Sperrstrom. Auch hier kann der Sperrstrom von $D1_u$ vernachlässigt werden. Aus diesen Überlegungen folgt, dass im Schaltzustand 1 für positive Ströme i_1 die Freilaufdiode $D1_u$ und für negative Ströme der IGBT $S1_u$ die Kennlinie beschreibt und die Sperrströme nicht berücksichtigt werden.

Der zweite Schaltzustand gilt während der Verriegelungszeit. Hierbei sind beide Schalter geöffnet. Dieser Zustand trifft auf Bild 5.3 und Bild 5.5 für den positiven Stromfluss bzw.

Bild 5.9 und Bild 5.11 bei negativer Stromrichtung zu. Im Bild 5.18 ist der zweite grundlegende Schaltzustand dargestellt.

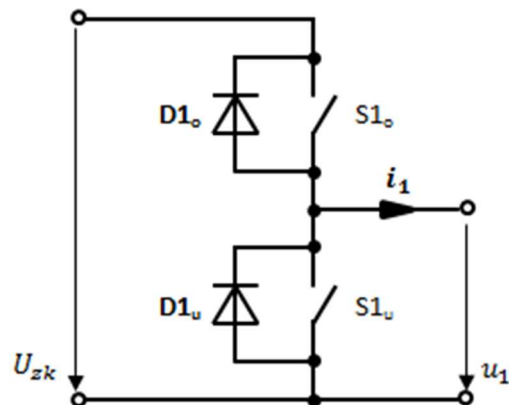


Bild 5.18: Schaltzustand 2

In diesem Fall sind die beiden Freilaufdioden die relevanten Bauteile. Die beiden IGBT's sind geöffnet und werden nicht berücksichtigt. Die Kennlinien für $D1_u$ und $D1_o$ sind im Bild 5.19 abgebildet.

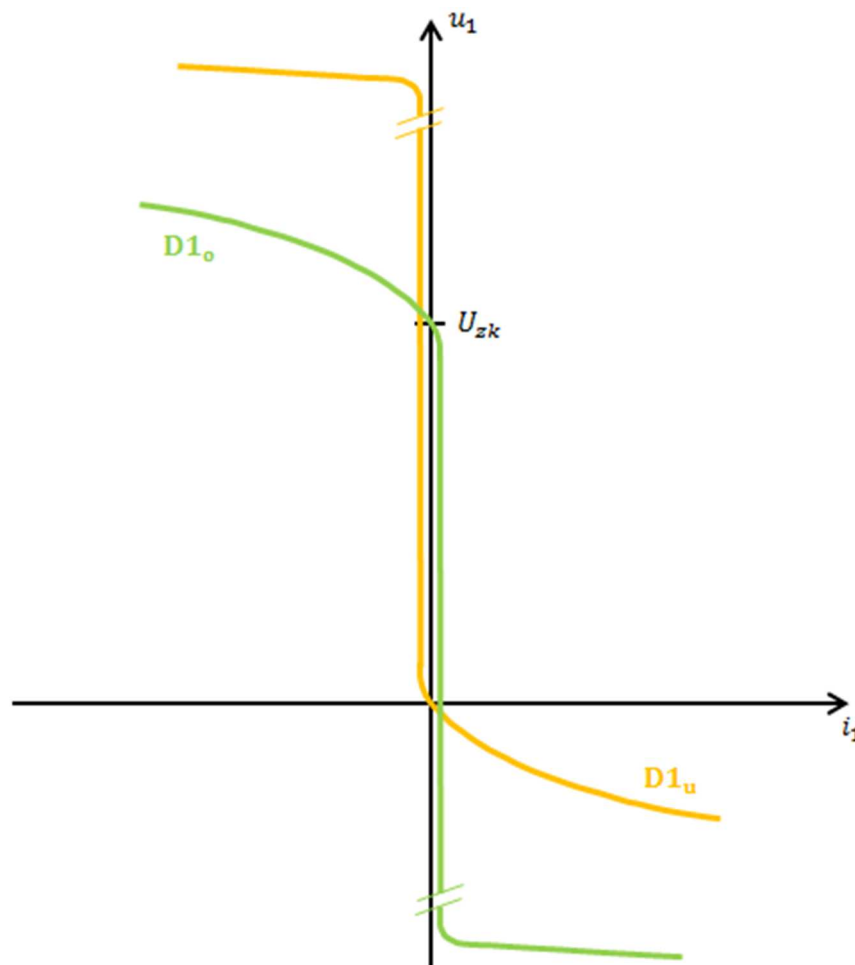


Bild 5.19: Kennlinien für $D1_u$ und $D1_o$ im Schaltzustand 2

Ein Strom $i_1 > 0$ fließt über die unter Freilaufdiode $D1_u$ und es kommt wie vorher beschrieben, zu einem Spannungsabfall. In diesem Fall muss die Spannung $u_1 < 0$ sein um

den Stromfluss zu ermöglichen. Die obere Freilaufdiode $D1_o$ sperrt und dadurch fließt ein sehr kleiner Sperrstrom. Für $i_1 < 0$ fließt der Strom über die obere Freilaufdiode $D1_o$. Um in diesem Zustand einen negativen Strom i_1 zu treiben, muss $u_1 > U_{zk}$ sein. Die untere Freilaufdiode befindet sich dabei im Sperrbereich. Auch hier werden die Sperrströme der beiden Freilaufdioden vernachlässigt, da sie um ein Vielfaches kleiner sind als die Durchlassströme. In diesem Schaltzustand ist für $i_1 = 0$ nicht eine Spannung $u_1 = 0$ zugeordnet, sondern alle möglichen Spannungen $0 \leq u_1 \leq U_{zk}$. Diese Unstetigkeit im Kurvenverlauf wird wie im vorhergehenden Kapitel mit einer Widerstandsgeraden approximiert. Die konkrete Realisierung dazu folgt weiter unten.

In diesem speziellen Fall während der Totzeit hängt das Verhalten des Spannungszwischenkreisumrichters von der angeschlossenen Last ab. Die Last bestimmt das Potential im Verbindungspunkt zwischen den beiden Freilaufdioden. Es wird auch nochmals darauf hingewiesen, dass ein Vorzeichenwechsel in diesem Schaltzustand nicht passieren kann. Wenn ein positiver Strom über die untere Freilaufdiode $D1_u$ fließt und der Strom bis $i_1 = 0$ absinkt, dann bleibt in diesem Fall der Strom auf null. Erst ein Übergang in einen der beiden anderen Schalterzustände bewirkt von null verschiedene Werte.

Im dritten und letzten Schaltzustand werden die obigen Betrachtungen im Bild 5.4 bzw. Bild 5.10 zusammengefasst und in Bild 5.20 nochmals graphisch dargestellt.

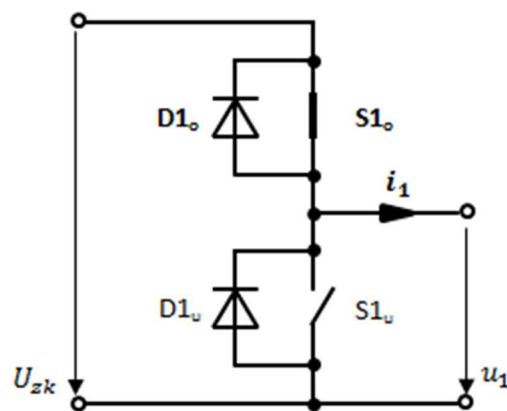


Bild 5.20: Schaltzustand 3

Der obere Zweig in der Halbbrücke ist hier von besonderem Interesse. Da $S1_o$ geschlossen ist fließt der Strom i_1 je nach Vorzeichen über den IGBT $S1_o$ oder die Freilaufdiode $D1_o$. Die Kennlinien für diesen Fall sind im Bild 5.21 abgebildet.

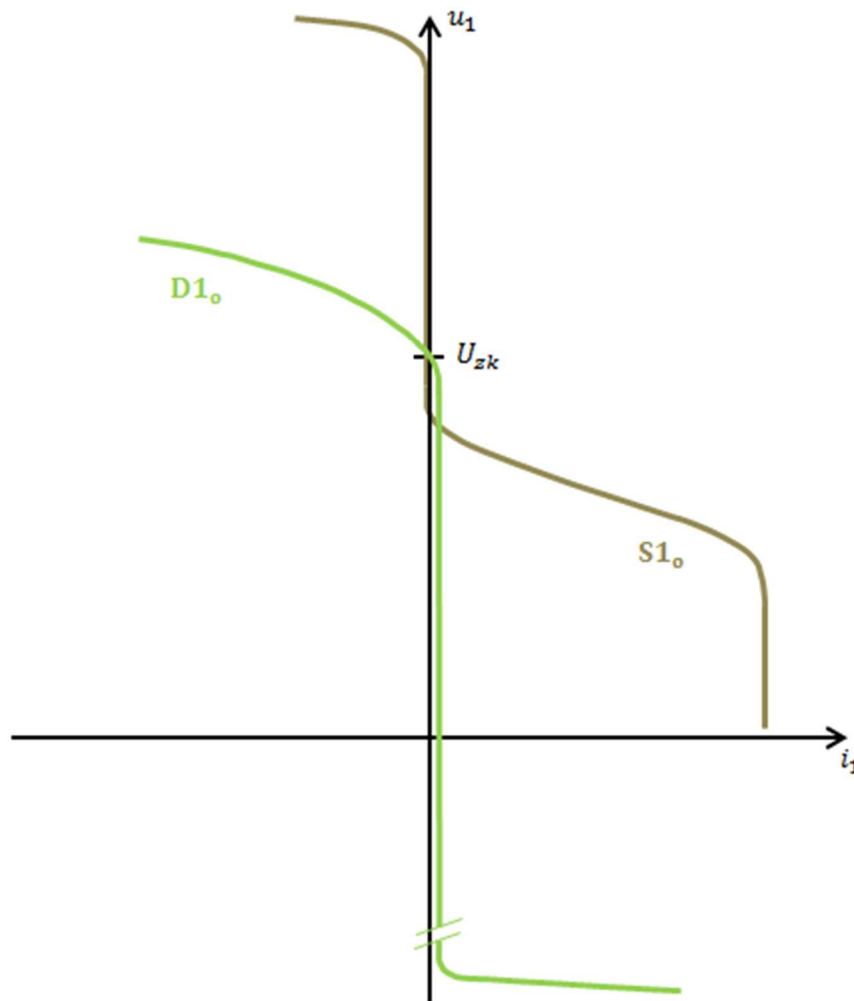


Bild 5.21: Kennlinien für $D1_o$ und $S1_o$ im Schaltzustand 3

Ein Strom $i_1 > 0$ fließt über den oberen IGBT $S1_o$ und es fließt ein sehr kleiner positiver Sperrstrom über die obere Freilaufdiode. Für $i_1 < 0$ geht der Strompfad über die obere Freilaufdiode. Ein sehr winziger Anteil fließt als Sperrstrom über den elektronischen Schalter $S1_o$. Auch hier werden die Sperrströme vernachlässigt, da sie im Verhältnis zu den Durchlassströmen relativ klein sind. Im Unterschied zum Schaltzustand 1, ist für $i_1 = 0$ die Spannung $u_1 = U_{zk}$. Aus diesen Betrachtungen folgt, dass für den positiven Strom i_1 die Kennlinie von $S1_o$ gilt und für die umgekehrte Stromrichtung der Durchlassverlauf von $D1_o$. Das Sperrverhalten von $S1_o$ und $D1_o$ wird für die weiteren Diskussionen vernachlässigt.

Durch die Vernachlässigung des Sperrverhaltens vereinfachen sich die zwei Kennlinien für die jeweils drei Schaltzustände zu einem einzigen Kennlinienverlauf. Für die Schaltzustände 1 und 3 erfolgt dies durch die Verbindung der Durchlasskennlinien von Freilaufdiode und IGBT. Für den Schaltzustand 2 müssen die beiden Durchlasskennlinien der Freilaufdioden mit einer Widerstandsgeraden verbunden werden. Diese Vorgangsweise wird noch durch die spätere Modellerweiterung veranschaulicht. Nun werden wir wie

vorhin schon angesprochen, die nichtlinearen Kennlinien für den IGBT und die Freilaufdiode bestimmen.

5.2.2 Ermittlung der Kennlinie für die Freilaufdiode

Für die Bestimmung der Kennlinie für die Freilaufdiode werden die Momentanwertverläufe, gemessen mit dem Leistungsmessgerät N5000, verwendet. Aus den vorhergehenden Betrachtungen 5.1.1 und 5.1.2, lässt sich ableiten in welchen Abschnitten von u_1 und i_1 die Freilaufdioden den Strom führen. Im Bild 5.22 und Bild 5.23 sind die Abschnitte in der die Freilaufdioden die Stromführung übernehmen für positive und negative Ströme abgebildet.

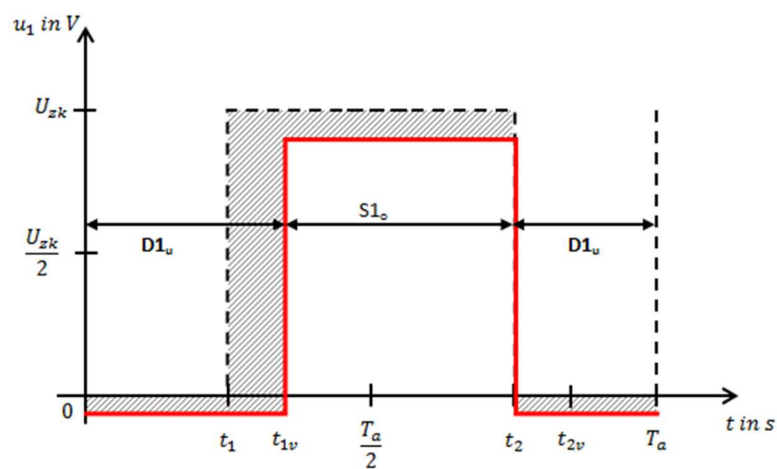


Bild 5.22: Aufteilung der Stromführung zwischen IGBT und Freilaufdiode bei $i_1 > 0$

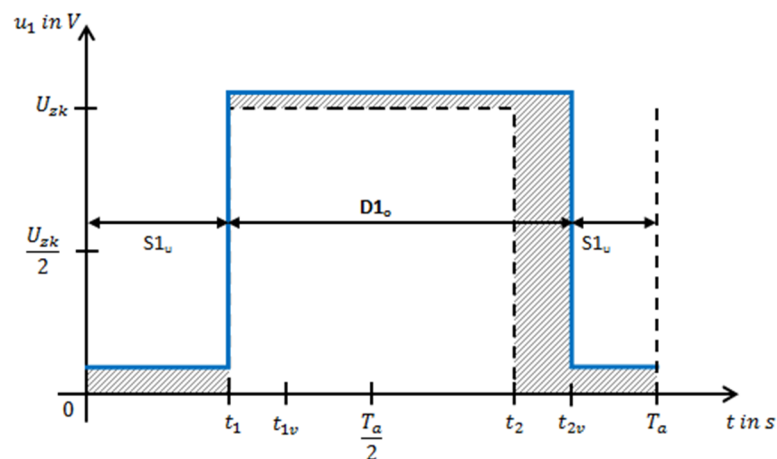


Bild 5.23: Aufteilung der Stromführung zwischen IGBT und Freilaufdiode bei $i_1 < 0$

Aus Bild 5.22 und Bild 5.23 wird ersichtlich, dass für die Bestimmung der Kennlinie für die Freilaufdiode Messungen mit $i_1 > 0$ verwendet werden. Die Auswertung ist in diesem Fall einfacher, da u_D direkt bzw. ohne Umrechnungen über die Zwischenkreisspannung verwendet werden kann.

Der spezielle Fall mit Strommittelwert $\bar{i}_1 = 0$, zeigt Bild 5.24 die Aufteilung der Stromführung auf die leistungselektronischen Bauteile. In dieser schematischen Skizze ist beim Vorzeichenwechsel im Strom der nichtlineare Übergang ersichtlich. Der Ausschnitt aus dem Zeitintervall $t_2 < t < T_a$ wird für den Anfangsbereich der Kennlinie verwendet.

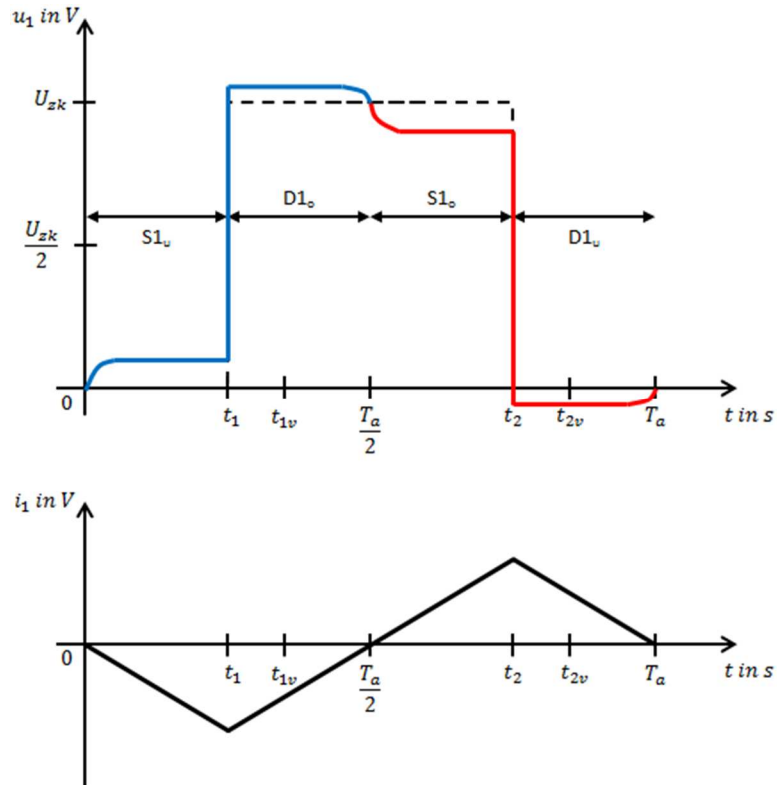


Bild 5.24: Aufteilung der Stromführung zwischen IGBT und Freilaufdiode bei $\bar{i}_1 = 0$

Mittels der Gegenspannung U_g kann der Strom im Intervall $-30 A \leq \bar{i}_1 \leq 30 A$ beliebig eingestellt werden. Zu jedem gewünschten Strom erhält man dann auch den zugehörigen PWM-Verlauf u_1 . Wie oben schon gezeigt ist in bestimmten Abschnitten der Ausgangsspannung des Spannungszwischenkreisumrichters $u_1 = -u_D$, der Spannungsabfall an der Freilaufdiode.

Somit lassen sich für beliebige Stromwerte die dazugehörigen Spannungsabfälle ermitteln. Dies geschieht dadurch, dass die Funktionsverläufe für i_1 und u_1 , im Bereich in dem die Freilaufdiode den Strom führt, herausgeschnitten werden.

Für den Anfangsverlauf der Kennlinie werden alle Punkte aus dem Messausschnitt im Bild 5.25 in ein Koordinatensystem, mit i_1 als Abszisse und u_1 als Ordinate, eingefügt (siehe Bild 5.27).

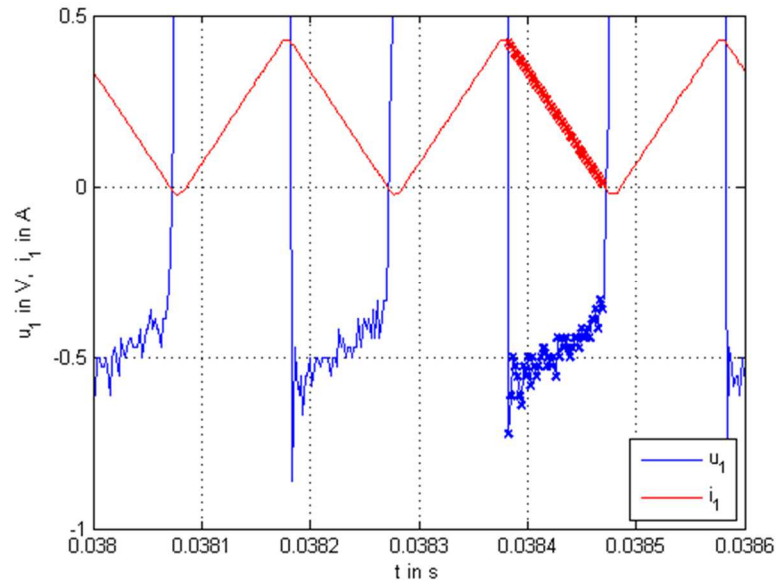


Bild 5.25: Messausschnitt für Kennlinienanfang der Freilaufdiode

Dieser Ausschnitt aus dem Messsignal im Bild 5.26 ist ein Beispiel für einen bestimmten Strommittelwert für den weiteren Kennlinienverlauf der Freilaufdiode. Durch das Auftragen dieser Messwerte im Bild 5.27 mit $u_1(i_1)$ ergibt sich eine „Punktwolke“. Nun wird der arithmetische Mittelwert für i_1 und u_1 berechnet. Dies ergibt nun einen Punkt, im Bild 5.27 dargestellt durch dunkelblaue Ringe, der für die weiterführende Berechnung herangezogen wird. Mit dieser Vorgehensweise lassen sich mehrere Punkte im interessierenden Strombereich ermitteln. Dadurch ergibt sich eine passable Approximation der Kennlinie für die Freilaufdiode.

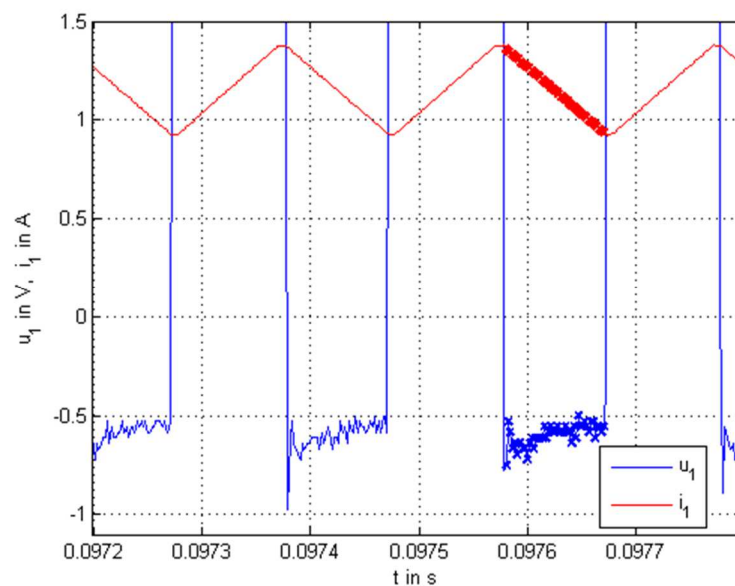


Bild 5.26: Ausschnitt aus der Messung für die Bestimmung der Kennlinie

Mit Hilfe des Curve-Fitting-Tool von MATLAB kann eine passende mathematische Funktion in die berechneten Punkte angenähert werden. Im nachfolgenden Bild 5.27 sind

die ausgeschnittenen Signale, die Mittelungspunkte und die mit dem Curve-Fitting-Tool erhaltene mathematische Funktion dargestellt.

Die beste Approximation ergibt sich durch Anwendung der Potenzfunktion (engl. Powerfunction) mit folgender Struktur

$$u_D(i_1) = a \cdot x^b + c \quad (5.1)$$

und den Parametern $a = 0,2314$, $b = 0,3656$ und $c = 0,3597$.

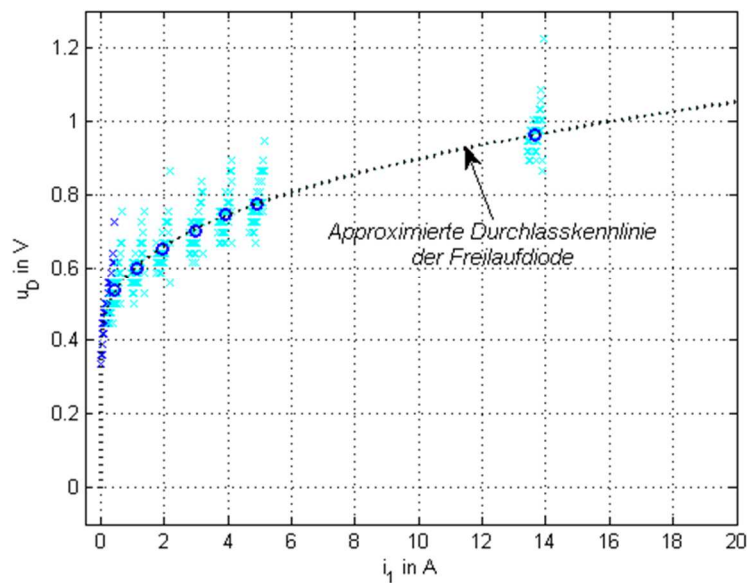


Bild 5.27: Ermittelte Kennlinie der Freilaufdiode und approximierte Potenzfunktion

5.2.3 Ermittlung der Kennlinie für den IGBT

Für den IGBT wird auf ähnliche Weise der Kennlinienverlauf bestimmt. Für den Anfangsverlauf der Kennlinie nützen wir die Zeitspanne in der Pulsperiode, in der der IGBT den Strom führt. Aus Bild 5.24 lassen sich zwei Möglichkeiten ableiten. Im Intervall der Pulsperiode von t_1 bis t_2 kann der Messverlauf vom Stromnulldurchgang bis t_2 verwendet werden. Die andere Option ist im Zeitabschnitt zwischen t_2 und t_1 der nächsten Pulsperiode. Diese startet ebenso beim Stromnulldurchgang und verläuft bis zu t_1 . Hier wurde die erste Möglichkeit gewählt und im Bild 5.28 dargestellt.

Der restliche Bereich der IGBT-Kennlinie kann mit Messpunkten entweder für $i_1 > 0$ im Zeitintervall $t_{1v} < t < t_2$ oder für $i_1 < 0$ im Zeitintervall $t_{2v} < t < t_1$ bestimmt werden und folgt aus Bild 5.22 bzw. Bild 5.23. Für die Ermittlung der IGBT-Kennlinie wird die erste Option eingesetzt, da für $U_{zk} = 100$ V nur Messungen mit positivem Strom durchgeführt wurden. Ein Beispiel dafür ist im Bild 5.29 abgebildet.

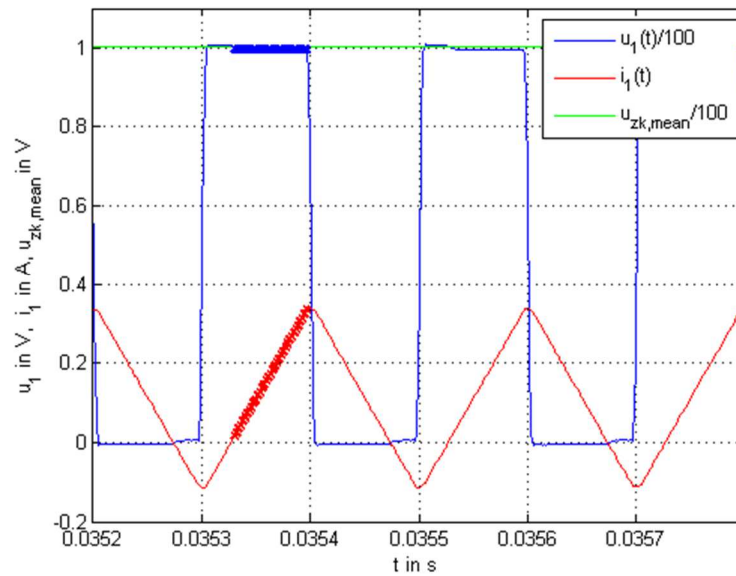


Bild 5.28: Messausschnitt für Bestimmung des Startbereichs der IGBT-Kennlinie

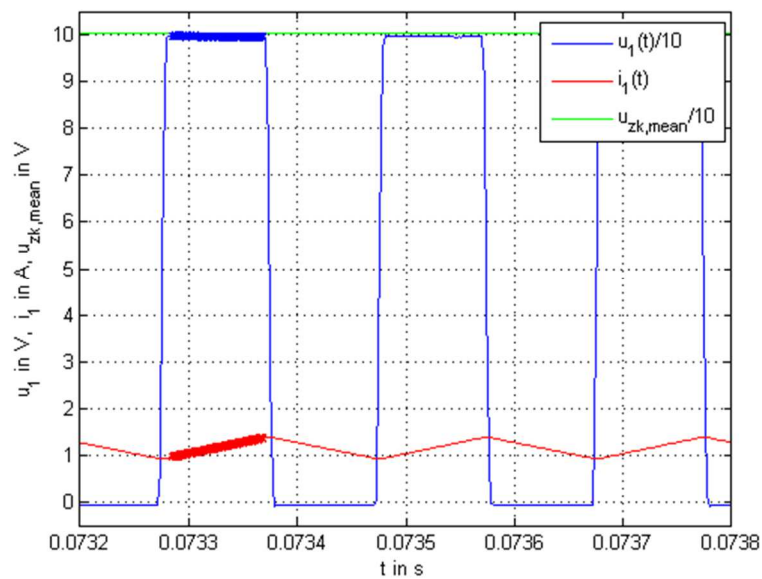


Bild 5.29: Messausschnitt für die Ermittlung des Kennlinienverlaufs des IGBT

Im Bild 5.30 ist der aus den Messungen ermittelte Kennlinienverlauf für den IGBT abgebildet. Auch in diesem Fall erhält man mittels der Potenzfunktion des Curve-Fitting-Tool eine gute Näherung zur IGBT-Kennlinie mit (5.1) und den Parametern $a = 0,2022$, $b = 0,4054$ und $c = 0,4268$.

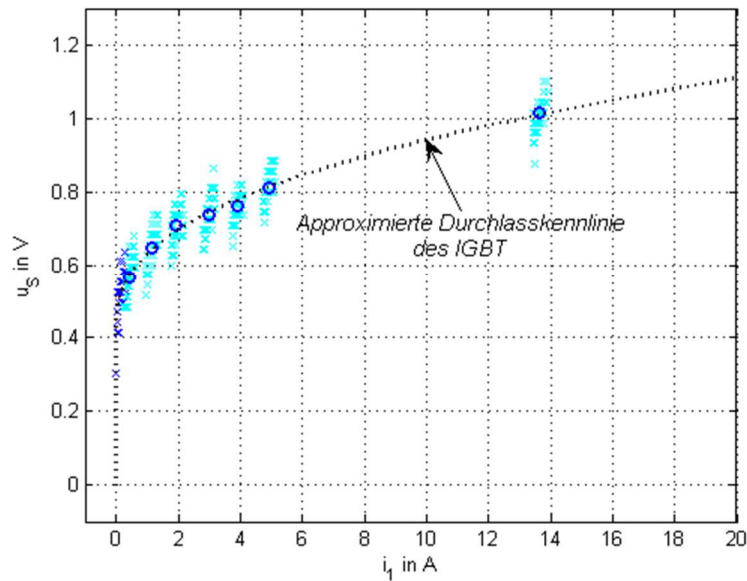


Bild 5.30: Ermittelte Kennlinie des IGBT

5.2.4 Zusammensetzung der beiden Kennlinien

Die Diskussion der drei allgemeinen Schalterzustände und die Vernachlässigung der Sperrbereiche werden nun für das nachfolgende Simulationsmodell zusammengefasst.

Schaltzustand 1

Dieser Zustand der Halbbrücke ist im Bild 5.16 abgebildet und die dazugehörigen Kennlinien für den IGBT und die Freilaufdiode im Bild 5.17. Unter den vorhin genannten Bedingungen ergibt sich für die Modellierung folgende zusammengefasste Kennlinie die im Bild 5.31 dargestellt ist.

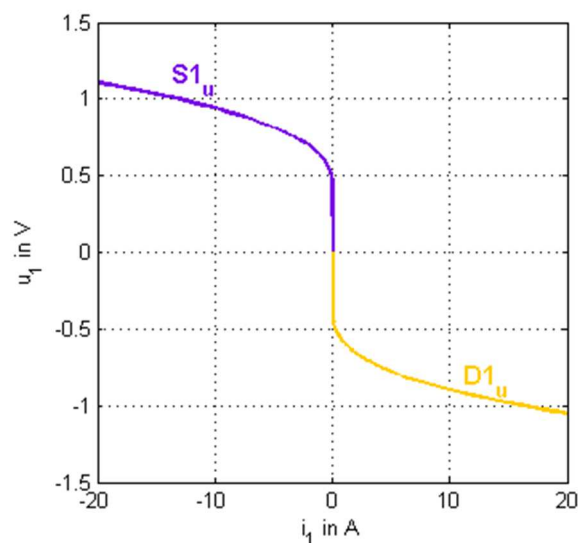


Bild 5.31: Kennlinie des Schaltzustands 1 für das Simulationsmodell

Nun ist noch interessant wie der Übergang der Kennlinie im Schaltzustand 1 genau verläuft. Abgebildet ist dies im Bild 5.32. Dabei ist ersichtlich, dass der Anfangsbereich der beiden Durchlasskennlinien eine Widerstandsgerade darstellt. Durch die Gleichung (5.1) zur Approximation der beiden Durchlasskennlinien lässt sich für einen gewünschten Strom die Spannung u_D bzw. u_S ermitteln. Der Strom wurde hierbei mit $i_1 = 0,01$ A gewählt. Damit erhält man für $u_D(i_1) = 0,4027$ V bzw. für $u_S(i_1) = 0,4581$ V. Die beiden Durchlasskennlinien von Freilaufdiode und IGBT werden nun folgendermaßen zusammengesetzt. Der erste Punkt der Kennlinie bildet der Koordinatenursprung mit $u_D(0) = 0$ V bzw. $u_S(0) = 0$ V. Der zweite Punkt der Kennlinie bilden die vorhin berechneten Punkte $u_D(0,01) = 0,4027$ V bzw. $u_S(0,01) = 0,4581$ V. Damit ergeben sich zwischen den ersten beiden Punkten der Durchlasskennlinien Widerstandsgeraden mit $R_{D,on} \approx 40,3 \Omega$ und $R_{S,on} \approx 45,8 \Omega$. Alle weiteren Punkte der Kennlinien werden mit der Approximationsgleichung (5.1) ermittelt.

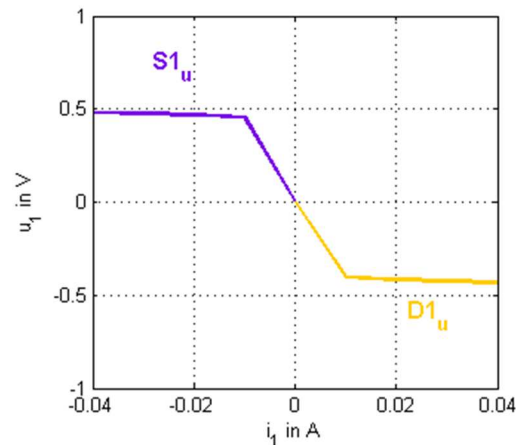


Bild 5.32: Ausschnitt der Kennlinie des Schaltzustands 1

Schaltzustand 2

Der zweite Schaltzustand repräsentiert die Verriegelungszeit. Im Bild 5.19 ist das zugehörige Ersatzschaltbild der Halbbrücke abgebildet. In diesem Fall sind die beiden Freilaufdioden relevant. Dies ist im Bild 5.19 ersichtlich. Für die Simulation wird die vereinfachte zusammengesetzte Kennlinie im Bild 5.33 verwendet.

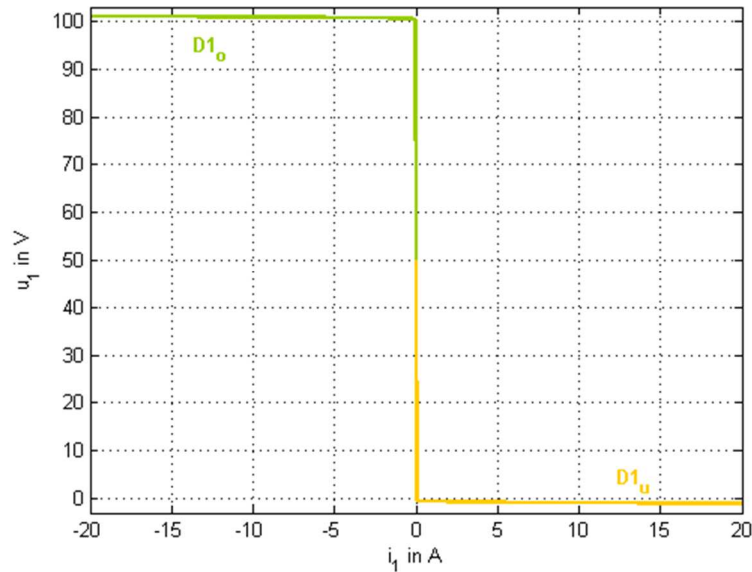


Bild 5.33: Kennlinie des Schaltzustands 2 für das Simulationsmodell

Die Verbindung der beiden Diodenkennlinien wird hier in ähnlicher Weise durchgeführt, wie beim Modell mit Berücksichtigung der Verriegelungszeit. Im Übergangsbereich in dem der Strom $i_1 = 0$ wird, kommt es zu einer linearen Widerstandsapproximation (siehe Bild 5.34).

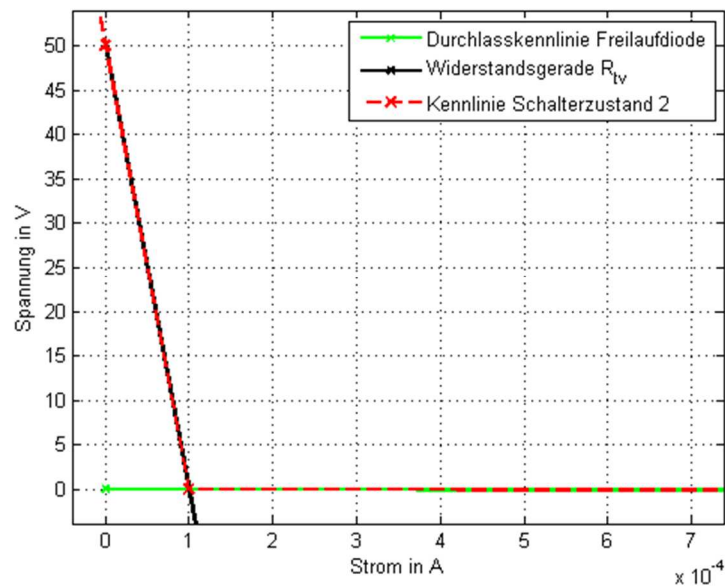


Bild 5.34: Zusammensetzung der Widerstandsgeraden mit der Durchlasskennlinie der Freilaufdiode

Dabei werden die beiden Schnittpunkte der Widerstandsgeraden ($R_{TV} = 500 \text{ k}\Omega$) mit den Durchlasskennlinien von oberer und unterer Freilaufdiode ermittelt. Für die untere Freilaufdiode ($D1_u$) ist dies im Bild 5.34 zu sehen. Für sehr kleine Ströme verläuft die Kennlinie entlang der Widerstandsgeraden bis zum Schnittpunkt mit der Durchlasskennlinie der Freilaufdiode. Danach verläuft die Kennlinie entlang der nichtlinearen Durchlasskennlinie der Freilaufdiode.

In der Simulation werden durch Vorgabe des Widerstandes R_{tv} bzw. der Durchlasskennlinie für die Freilaufdiode die beiden Schnittpunkte berechnet und die Kennlinie für den Schaltzustand 2, wie zuvor beschrieben, zusammengesetzt.

Schaltzustand 3

Diesen Abschnitt in der Pulsperiode beschreiben Bild 5.20 und Bild 5.21. Für das Simulationsmodell wird folgende vereinfachte Kennlinie im Bild 5.35 verwendet. Der Übergangsbereich bei $i_1 = 0$ A ist ähnlich zu dem im Schaltzustand 1.

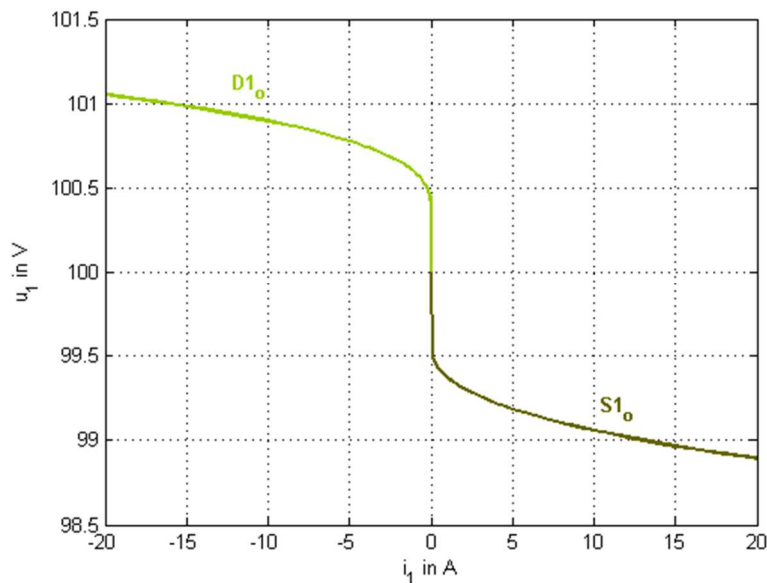


Bild 5.35: Kennlinie des Schaltzustands 3 für das Simulationsmodell

5.3 Modellerweiterung für die Berücksichtigung der Spannungsabfälle an den elektronischen Bauelementen

Die grundlegende Modellstruktur wie im Bild 4.15 bleibt erhalten. Erneuerungen gibt es im Block „PWR“ der im Bild 5.36 dargestellt ist. Für das Modell mit Berücksichtigung der Spannungsabfälle an IGBT und Freilaufdiode wurde der Block „Spannungsabfälle IGBT, Freilaufdiode“ für die drei Schalterzustände hinzugefügt.

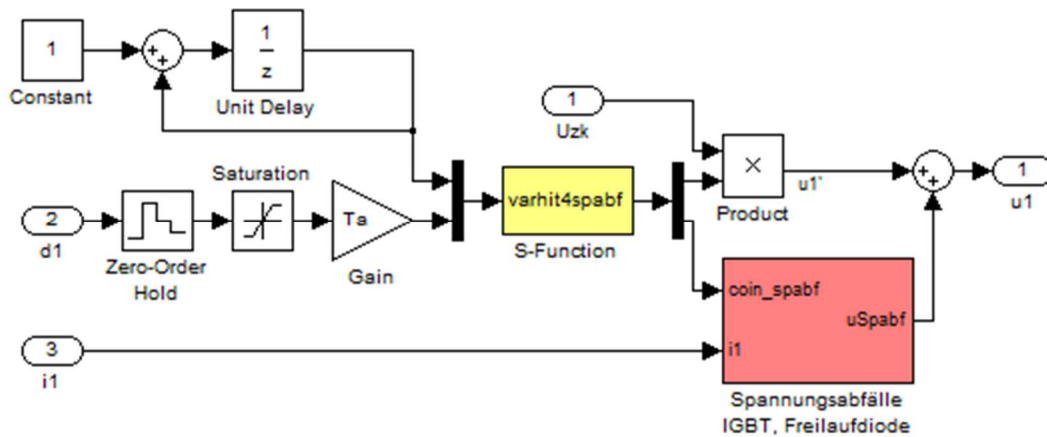


Bild 5.36: Koppelplan des Spannungszwischenkreisumrichters unter Berücksichtigung der Spannungsabfälle

5.3.1 Eingangs- bzw. Ausgangssignale der S-Function

Die Eingangssignale für die S-Function bleiben unverändert zum vorhergehenden Modell. Bei den Ausgangssignalen der S-Function bleibt das erste gleich wie im Modell mit der Berücksichtigung der Verriegelungszeit, das zweite wird jedoch verändert. Im vorhergehenden Modell wurde mit dem zweiten Ausgang der S-Function zwischen zwei Zuständen unterschieden. Während der Verriegelungszeit wurde der Wert 1 ausgegeben, für alle anderen Zeitintervalle null. In diesem Modell wird dieses Signal verändert und hat ganzzahlige Werte zwischen 1 und 3. Dabei entsprechen die drei Werte den drei Schalterzuständen. Im Bild 5.37 ist dieses Signal für eine Pulsperiode abgebildet. Dieses Signal dient dazu, zwischen den in der Simulation gerade wirksamen Schalterzuständen zu Switchen.

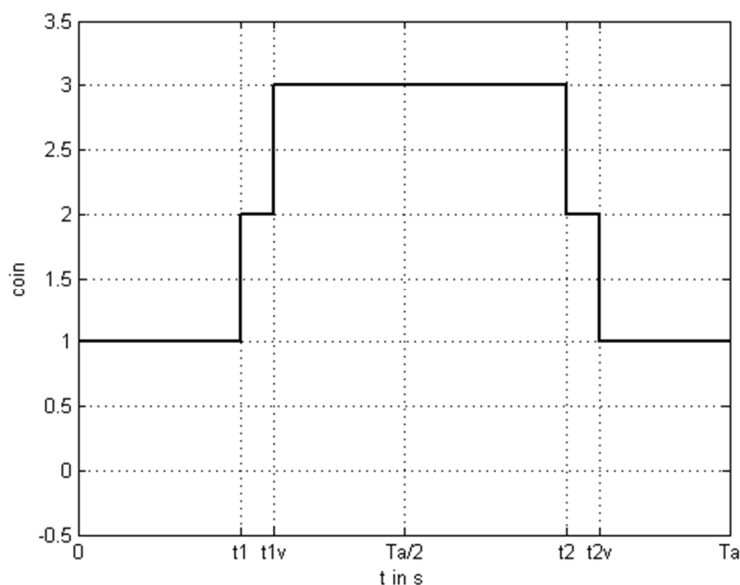


Bild 5.37: Zweiter Ausgang der S-Function mit den drei ganzzahligen Schalterzuständen

5.3.2 Simulationsablauf in der S-Function

Bei der MATLAB-Funktion „varhit4spabf.m“ wurde im Vergleich zum vorhergehenden Modell der zweite Ausgang der S-Function, wie vorhin beschrieben, implementiert. Der Simulationsablauf über drei Pulsperioden gestaltet sich ähnlich zu dem in Tabelle 2. Die Spalte mit u_3 fällt weg, da der Strom i_1 für diese Modellrealisierung nicht benötigt wird. In der Spalte *sys* kommt in der Zeile Output ein zweiter Ausgang hinzu, der für eine Pulsperiode die Werte aus Bild 5.37 annimmt. Für weitere Details wird an dieser Stelle auf den Quellcode im Anhang A.4 verwiesen.

5.3.3 Ausgangsspannung u_1

Im Bild 5.36 ist der Block „PWR“ aus dem MATLAB/Simulink-Modell abgebildet. Zum Vergleich mit dem vorhergehenden Modell im Bild 4.28, wurde die „Lookup-Table“, der „Switch“ und „Constant-Block“ durch den Subblock „Spannungsabfälle IGBT, Freilaufdiode“ ersetzt. Dieser Block enthält den im Bild 5.38 abgebildeten Koppelplan.

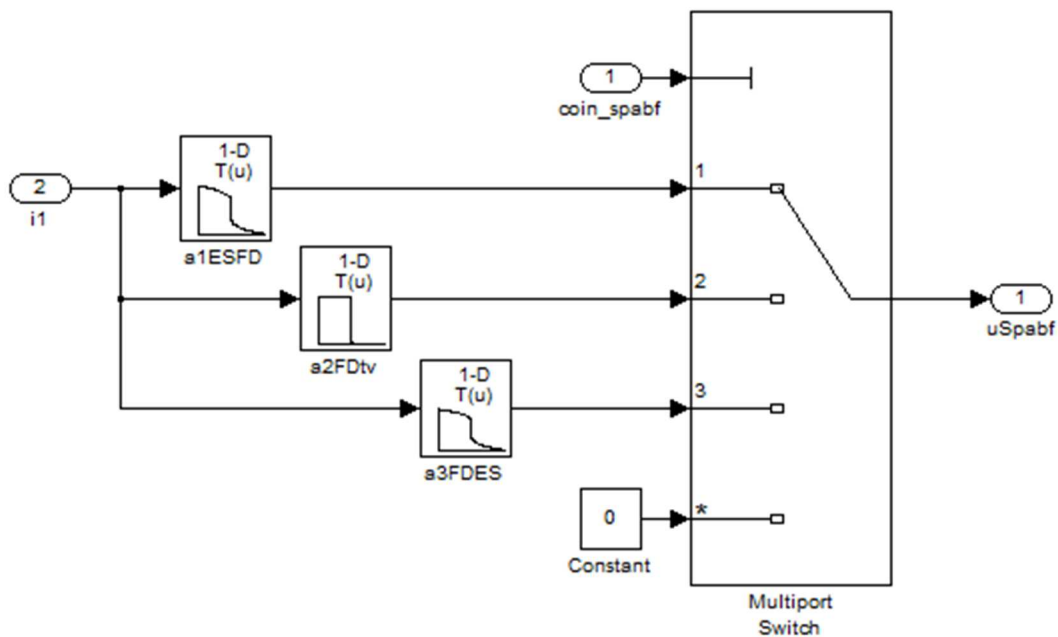


Bild 5.38: Block „Spannungsabfälle IGBT, Freilaufdiode“

Der Eingang „coin_spabf“ (*control input*) ist das zweite Ausgangssignal aus der S-Function und hat drei mögliche Werte. Damit wird der „Multiport-Switch“ in die gewünschte Stellung gebracht. Normalerweise sollte dieses Eingangssignal nur die Werte der Menge $\{1, 2, 3\}$ besitzen. Ist dies nicht der Fall, tritt dieser Fehler dadurch zu Tage, dass auf die vierte Switch-Position (mit * gekennzeichnet) geschaltet wird und damit der Ausgang „uSpabf“ für ein gewisses Zeitintervall auf null bleibt.

Der zweite Eingang ist der Strom i_1 . Aus diesem wird mit den Lookup-Tabellen der jeweilige Spannungsabfall erzeugt. In der ersten Lookup-Tabelle „a1ESFD“ ist die im Bild

5.31 dargestellte Kennlinie für den ersten Schaltzustand eingetragen. Die zweite Lookup-Tabelle „a2FDtv“ enthält die Kennlinie aus Bild 5.33 und die dritte „a3FDES“ beinhaltet den Kennlinienverlauf aus Bild 5.35. Beim dritten Schaltzustand ist zu beachten, dass der Wert der Zwischenkreisspannung (hier $U_{zk} = 100 \text{ V}$) im Bild 5.35 für den Kurvenverlauf in der Lookup-Tabelle abzuziehen ist. Da die S-Function in diesem Zeitintervall $t_{1v} \leq t < t_2$ immer den Wert 1 ausgibt und damit u_1' der Zwischenkreisspannung entspricht, wird nur mehr der Spannungsabfall an den leistungselektronischen Bauelementen benötigt.

Die Ausgangsspannung des Spannungszwischenkreisumrichters $u_1 = u_1' + u_{\text{Spabf}}$. Das Signal u_1' ist dasselbe wie im vorhergehenden Modell und entspricht dem Spannungspuls der zum Zeitpunkt t_{1v} auf den Wert der Zwischenkreisspannung springt bzw. zum Zeitpunkt t_2 zurück auf 0 V. Es ist nochmals festzuhalten, dass dieser Spannungspuls unabhängig vom Stromvorzeichen ist und somit für alle Pulsperioden die gleiche Pulsbreite besitzt. Durch das Signal „uSpabf“ wird in Abhängigkeit vom Strom der Spannungsabfall an IGBT und Freilaufdiode berücksichtigt und zum Spannungspuls u_1' addiert. Damit ergibt sich die Ausgangsspannung u_1 des Pulswechselrichters.

5.4 Validierung des Modells mit Spannungsabfällen an IGBT und Freilaufdiode

Der Vergleich zwischen Messung und Simulation mit Berücksichtigung der Spannungsabfälle an der Umrichternichtlinearität ist im Bild 5.39 dargestellt. Interessant ist hierbei auch die Gegenüberstellung mit dem vorhergehenden Modell im Bild 4.32. Dabei wird klar ersichtlich, dass der Simulationsverlauf mit Verriegelungszeit $t_v = 2 \mu\text{s}$ im Amperebereich $5 \text{ A} < \bar{i}_1 < 20 \text{ A}$ bzw. $-20 \text{ A} > \bar{i}_1 > -5 \text{ A}$ sehr gut mit der Messung übereinstimmt. Daraus folgt das Fazit, dass mit diesem Modell die Nichtlinearität dieses Spannungszwischenkreisumrichters im Strombereich größer $\pm|5 \text{ A}|$ eine hinreichend genaue Approximation ist.

Ganz zufriedenstellend ist dieses Modell aber noch nicht, da im Stromintervall $-5 \text{ A} < \bar{i}_1 < 5 \text{ A}$ der Kurvenverlauf bzw. die Kurvenform von der Simulation im Vergleich mit der Messung Ungenauigkeiten aufweist. Diese Abweichung und deren Ursache werden nun im folgenden Kapitel untersucht.

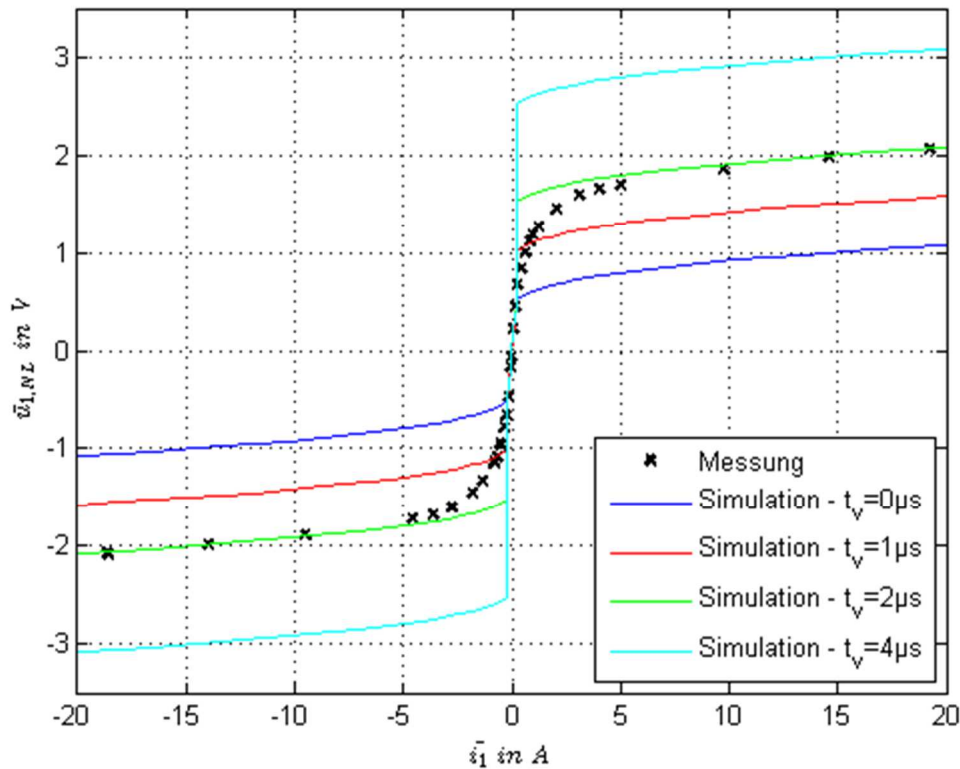


Bild 5.39: Vergleich der Umrichternichtlinearität zwischen Messung und Simulation

Diskussion der Simulierten Zeitverläufe von u_1 beim Stromlücken

Für den zeitlichen Funktionsverlauf der Ausgangsspannung u_1 ergeben sich hierbei die im Bild 5.40 und Bild 5.41 dargestellten Simulationsergebnisse. Dabei ist während der ersten Totzeit $t_1 \leq t \leq t_{1v}$ für das Modell mit Berücksichtigung der Spannungsabfälle ($u_{1,spabf}$) ein sonderbarer Funktionsverlauf zu sehen. Beim vorhergehenden Modell mit Verriegelungszeit ($u_{1,tv}$) tritt dies nicht auf. Die beiden Simulationen wurden mit folgenden Parameterwerten durchgeführt. $U_{zk} = 100 \text{ V}$, $d_1 = 0,5$, $t_v = 10 \mu\text{s}$, $R_{tv} = 500 \text{ k}\Omega$ und $U_g = 48 \text{ V}$. Als Solver wurde ode45 verwendet. Dabei hat die maximale Schrittweite den Wert $T_a/2$ bzw. die relative Toleranz $1e-6$.

Es wird vermutet, dass es durch die im vorhergehenden Kapitel diskutierten Auswirkungen von R_{tv} auf den Eigenwert, zu diesem komischen Funktionsverlauf kommt. Durch die Verwendung von anderen Solvern (z.B. ode15s bzw. ode23s) konnte dieser ungewünschte Funktionsverlauf vermieden werden. Auch die Verkleinerung der relativen Toleranz brachte zufriedenstellende Simulationsergebnisse. Da diese Probleme aber keinen merkbaren Einfluss auf die Dauer der Simulation bzw. auf die Genauigkeit Simulationsergebnisse für die Umrichternichtlinearität haben, werden sie in diesem Modellierungsschritt toleriert.

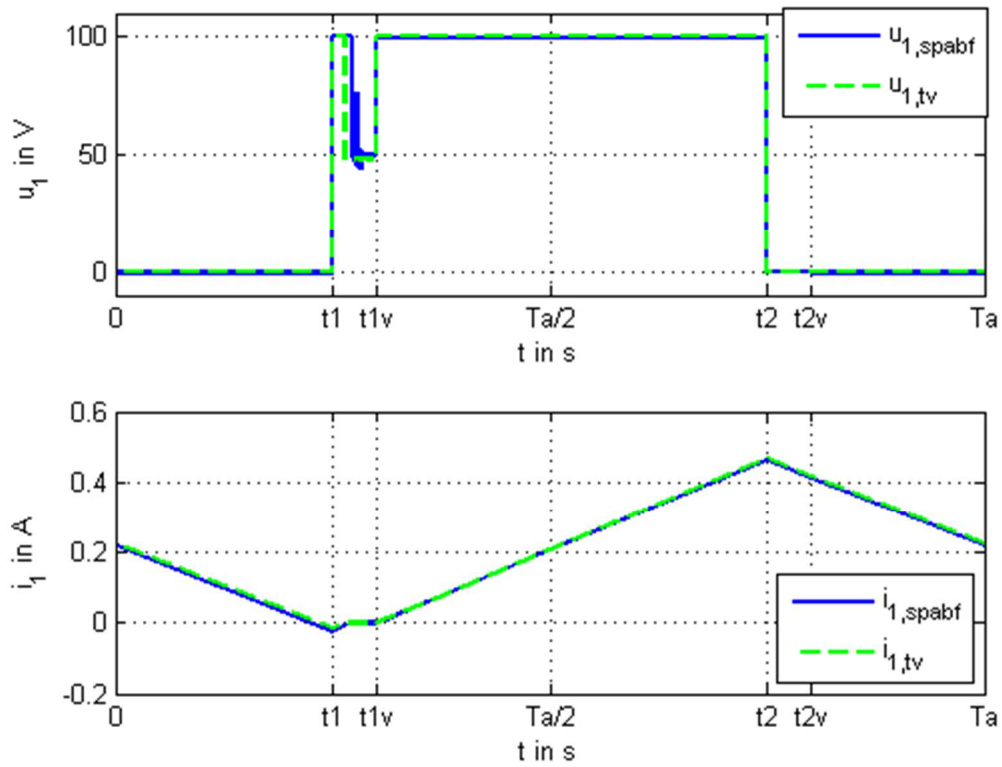


Bild 5.40: Simulierte Momentanwertverläufe von Strom und Spannung

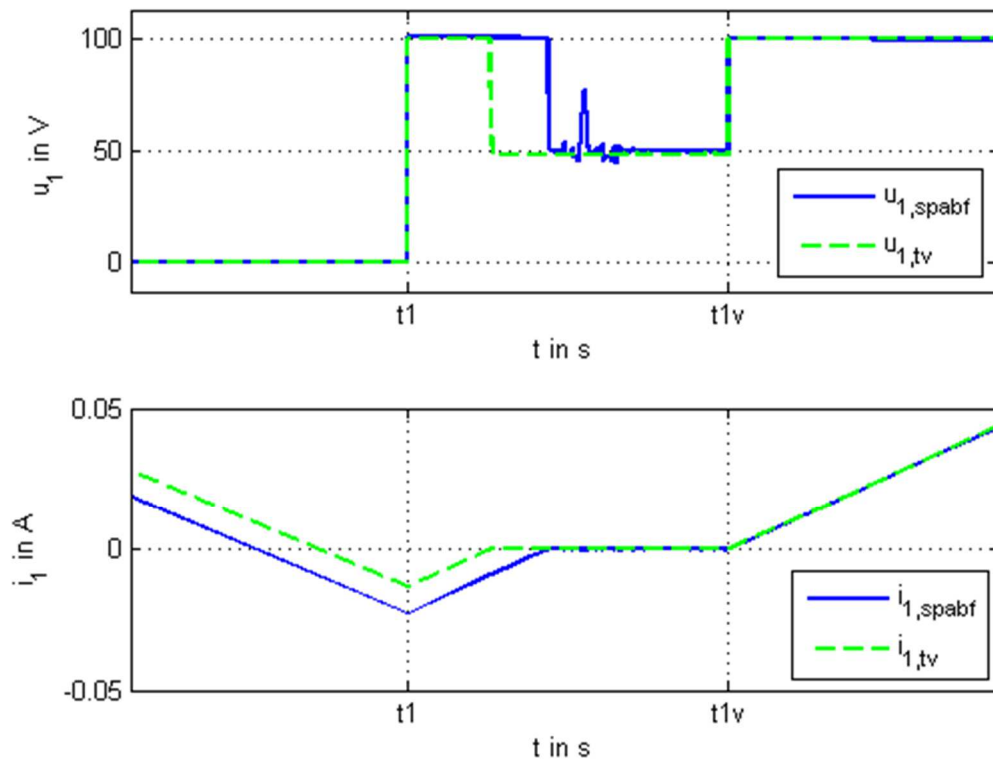


Bild 5.41: Ausschnitt der Verriegelungszeit in dem der Strom i_1 null wird (Stromlücken)

6 Modell unter Berücksichtigung der parasitärer Kapazität

In diesem Kapitel wird die letzte Modellerweiterung behandelt. Hier wird der Einfluss der parasitären Kapazität auf das Verhalten des Spannungszwischenkreisumrichters untersucht. Als erstes erfolgen für positiven bzw. negativen Strom i_1 die Auswirkungen des Einflusses von C_p auf das Verhalten der Halbbrücke. Danach folgen die Implementierung in das vorhergehende Modell und der Nachweis, dass das modifizierte Modell die Auswirkungen der parasitären Kapazität im Pulsmuster berücksichtigt. Den Abschluss bildet die Diskussion über den gemessenen und simulierten Kurvenverlauf der Umrichternichtlinearität.

6.1 Die parasitäre Kapazität

In [1] wird der Effekt der parasitären Kapazität als zusätzlicher wichtiger Faktor der Umrichternichtlinearität erwähnt. Am Ausgang des Spannungszwischenkreisumrichters wird die parasitäre Kapazität im Bild 6.1 durch C_p repräsentiert.

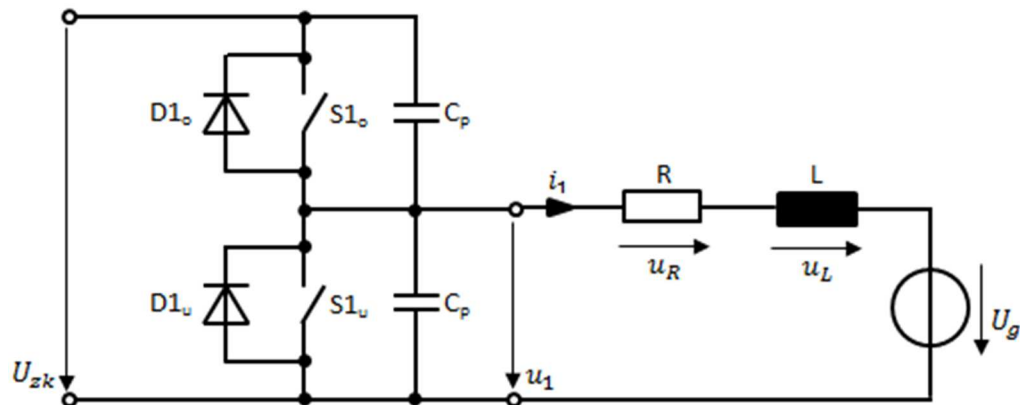


Bild 6.1: Halbbrücke mit Berücksichtigung der parasitären Kapazität

Es wird nun untersucht, welchen Einfluss die parasitäre Kapazität in Abhängigkeit vom Stromvorzeichen auf den Funktionsverlauf der Ausgangsspannung u_1 besitzt. Dabei wird die Pulsperiode T_a wie in den vorhergehenden Kapiteln in fünf Zeitintervalle unterteilt.

- $0 \leq t < t_1$
- $t_1 \leq t < t_{1v}$
- $t_{1v} \leq t < t_2$
- $t_2 \leq t < t_{2v}$
- $t_{2v} \leq t < T_a$

Für die folgende Diskussion werden die beiden parasitären Kapazitäten im Bild 6.1 mit $C_{p,o}$ und $C_{p,u}$ unterschieden. Vereinfachend wird aber in dieser Arbeit angenommen, dass $C_p = C_{p,u} = C_{p,o}$ gilt. Die unterschiedliche Bezeichnung dient zur eindeutigen Erklärung.

6.1.1 Strom positiv

Mit Bild 6.2 bis Bild 6.7 erfolgt die Erklärung für die Auswirkungen der parasitären Kapazität C_p auf das PWM-Signal bei positivem Strom i_1 . Im Bild 6.2 ist das Ausgangssignal u_1 des Spannungszwischenkreisumrichters abgebildet. Die dazugehörigen Schalterzustände der Halbbrücke für die jeweiligen Zeitintervalle sind im Bild 6.3 bis Bild 6.7 dargestellt.

a) Schaltzustand während dem Zeitintervall $0 \leq t < t_1$ (siehe Bild 6.3)

Der Schalter $S1_u$ ist geschlossen bzw. $S1_o$ ist offen. Ein Strom in positiver Richtung fließt über die untere Freilaufdiode ($D1_o$). In diesem Fall fällt an der Freilaufdiode eine Spannung $u_D(i_1)$ in Flussrichtung ab. Für das PWM-Signal am Umrichterausgang gilt $u_1 = -u_D(i_1)$. Diese Spannung liegt auch an der unteren parasitären Kapazität ($C_{p,u}$). Somit ist diese auf $-u_D(i_1)$ aufgeladen. Die obere parasitäre Kapazität ($C_{p,o}$) ist auf $u_{Cp,o} = U_{zk} + u_D(i_1)$ geladen. In diesem Zeitintervall hat C_p keine Auswirkung auf das Verhalten der Halbbrücke.

b) Schaltzustand während dem Zeitintervall $t_1 \leq t < t_{1v}$ (siehe Bild 6.4)

Zu Beginn dieses Zeitintervalls öffnet $S1_u$ und die Verriegelungszeit ist wirksam. Dieser Schaltvorgang hat keinen Einfluss auf den Weg des Stromes. Dieser fließt weiterhin über die untere Freilaufdiode ($D1_o$) und wie vorhin ist $u_1 = -u_D(i_1)$. Die Ladungszustände an den beiden parasitären Kapazitäten bleiben unverändert zum vorhergehenden Fall. Auch in diesem Zeitabschnitt zeigt C_p keine Wirkung.

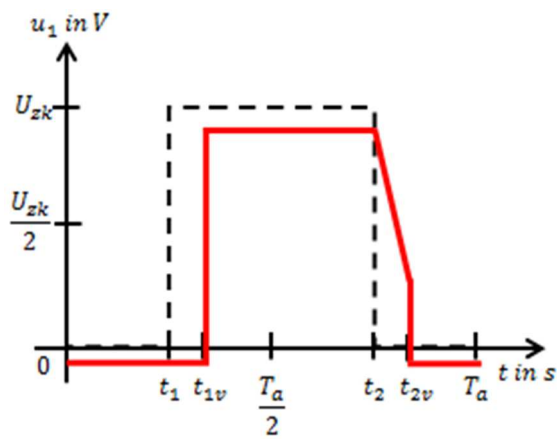


Bild 6.2: PWM mit parasitärer Kapazität $i_1 > 0$

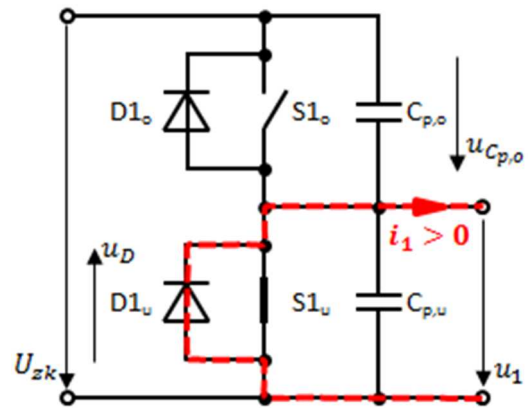


Bild 6.3: $0 \leq t < t_1$

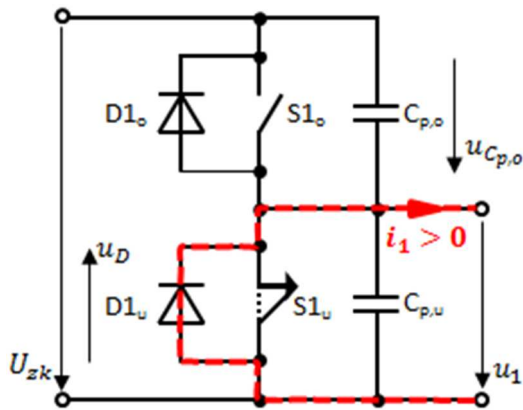


Bild 6.4: $t_1 \leq t < t_{1v}$

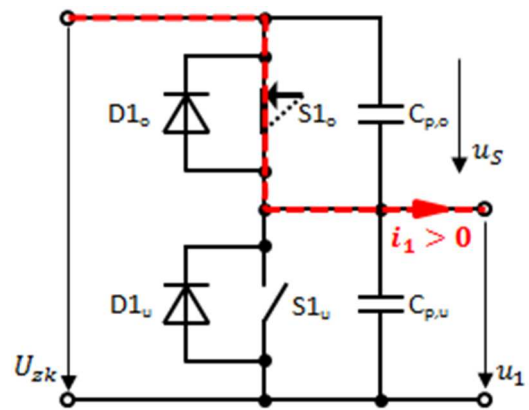


Bild 6.5: $t_{1v} \leq t < t_2$

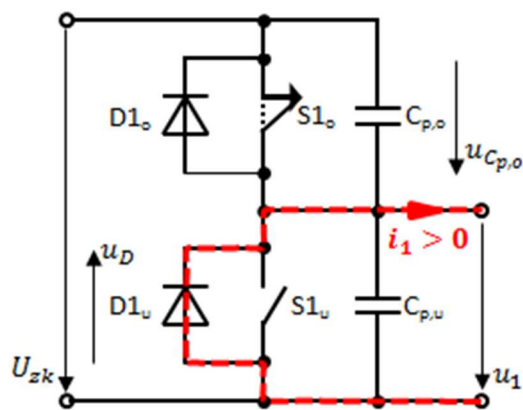


Bild 6.6: $t_2 \leq t < t_{2v}$

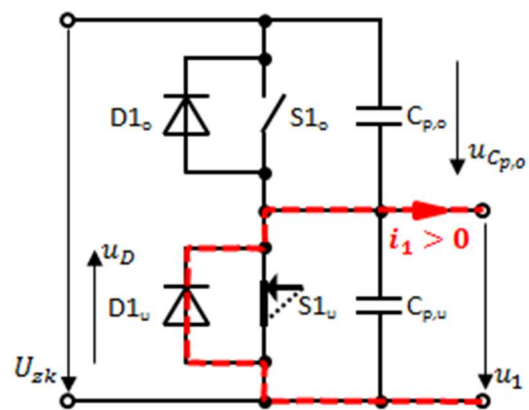
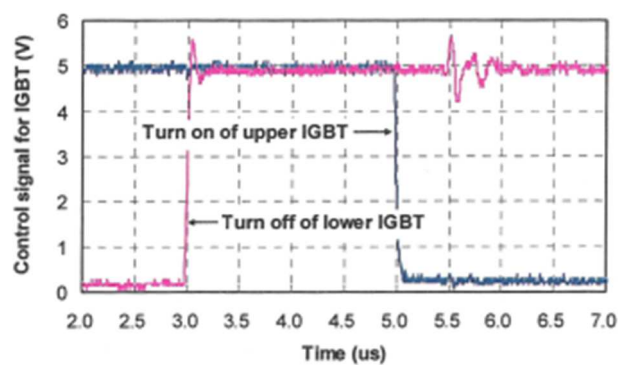


Bild 6.7: $t_{2v} \leq t < T_a$

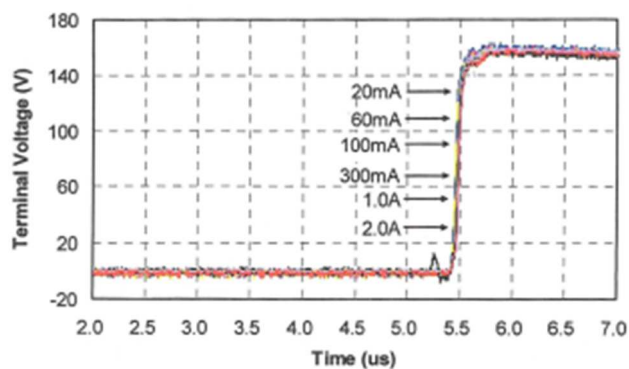
c) Schaltzustand während dem Zeitintervall $t_{1v} \leq t < t_2$ (siehe Bild 6.5)

Zum Zeitpunkt t_{1v} schließt der obere Schalter ($S1_o$), der Strom in positiver Richtung wechselt den Pfad und fließt nun über den oberen Schalter. Dabei fällt eine Spannung $u_S(i_1)$ in Stromrichtung ab. Diese Spannung liegt auch an der oberen parasitären Kapazität $C_{p,o}$, die parallel dazu liegt. Im vorigen Fall war $u_{C_{p,o}} = U_{zk} + u_D(i_1)$, somit entlädt sich $C_{p,o}$ nun auf $u_{C_{p,o}} = u_S(i_1)$. Damit wird die Spannung am Umrichter Ausgang, sowie an der unteren parasitären Kapazität zu $u_1 = U_{zk} - u_S(i_1)$. Dem zur Folge wird $C_{p,u}$ von $-u_D(i_1)$ auf $U_{zk} - u_S$ geladen.

Aus [1] ist bekannt wie dieser Vorgang abläuft. Wechselt der Strom von der Freilaufdiode ($D1_u$) zum IGBT ($S1_o$), passiert dies nahezu unmittelbar mit dem Schaltvorgang zum Zeitpunkt t_{1v} . Die Begründung dafür liegt in der Tatsache, dass der IGBT ein aktiver Schalter ist und einen kleinen Einschaltwiderstand besitzt mit dem eine sofortige Ladung oder Entladung der parasitären Kapazität möglich ist. Im Bild 6.8 ist der sprunghafte Anstieg der Ausgangsspannung des Pulswechselrichters (Terminal voltage) ersichtlich, der im Prinzip unabhängig von der Stromhöhe ist. Daraus lässt sich ableiten, dass die aufgezeigten Einflüsse von C_p vernachlässigt werden können.



(a) Control signals for IGBTs



(b) Terminal voltage

Bild 6.8: Messung des Anstieges der Spannung u_1 am Umrichter Ausgang aus [1]

d) Schaltzustand während dem Zeitintervall $t_2 \leq t < t_{2v}$ (siehe Bild 6.6)

Am Anfang dieses Zeitintervalls zum Zeitpunkt t_2 öffnet der obere Schalter ($S1_o$) und der Strom wechselt den Pfad auf die untere Freilaufdiode ($D1_o$). Damit ist die Spannung $u_1 = -u_D(i_1)$. Somit wird die untere parasitäre Kapazität von der Spannung $U_{zk} - u_S(i_1)$ auf $-u_D(i_1)$

entladen. Dem entgegen erfolgt bei $C_{p,o}$ eine Aufladung der Spannung $u_{C_{p,o}}$ von $u_S(i_1)$ auf $U_{zk} + u_D(i_1)$.

Laut [1] ist beim umgekehrten Strompfadwechsel von IGBT ($S_{1,o}$) zu Freilaufdiode ($D_{1,u}$) ein anderes Phänomen beobachtbar. Durch das Abschalten des IGBT verhält sich die Ausgangsspannung am Spannungszwischenkreisumrichter wie im Bild 6.9. Durch den Effekt der parasitären Kapazität kommt es zu einem Abstieg der Spannung u_1 , dessen Steigung signifikant von der Stromhöhe abhängig ist.

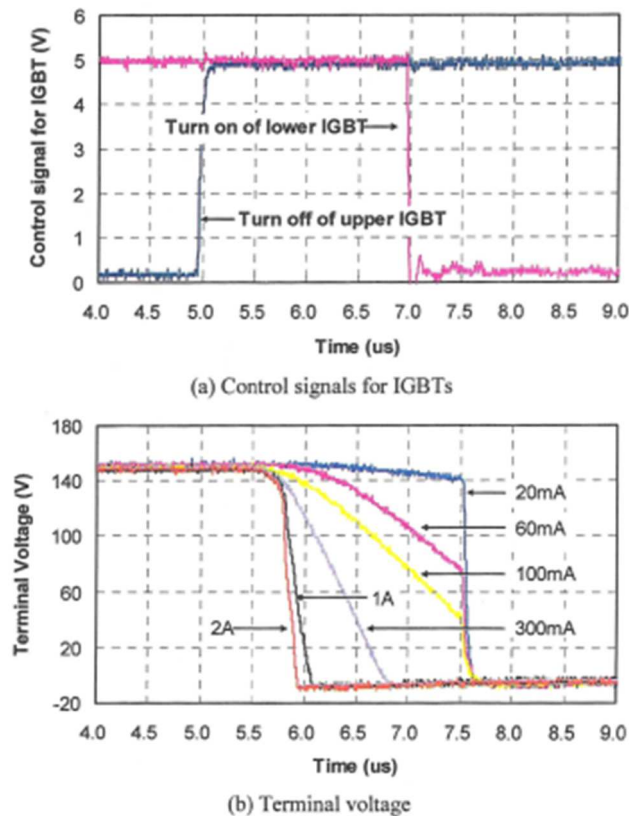


Bild 6.9: Gemessener Abfall der Ausgangsspannung u_1 am Pulswechselrichter aus [1]

e) Schaltzustand während dem Zeitintervall $t_2 \leq t < T_a$ (siehe Bild 6.7)

Der untere Schalter ($S_{1,o}$) ist zum Zeitpunkt t_{2v} geschlossen und der Strom fließt weiterhin über die untere Freilaufdiode. Auf die Spannungszustände der parasitären Kapazität hat dies keine Auswirkung. Die Spannung am Ausgang des Pulswechselrichters ist wie im vorigen Zeitintervall $u_1 = -u_D(i_1)$. Die parasitäre Kapazität hat in diesem Zeitabschnitt keine Auswirkung auf die Ausgangsspannung u_1 des Spannungszwischenkreisumrichters.

6.1.2 Strom negativ

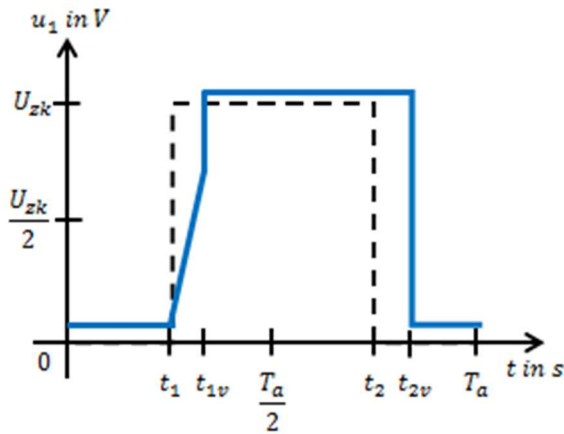


Bild 6.10: PWM mit parasitärer Kapazität $i_1 < 0$

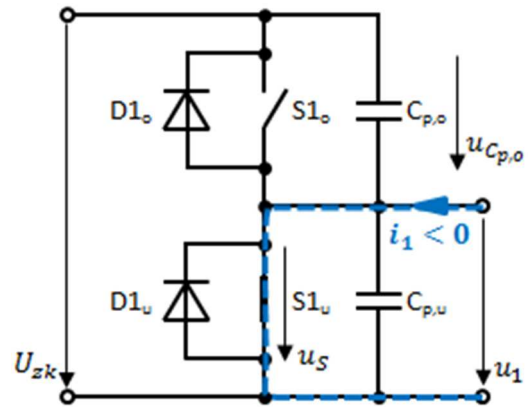


Bild 6.11: $0 \leq t < t_1$

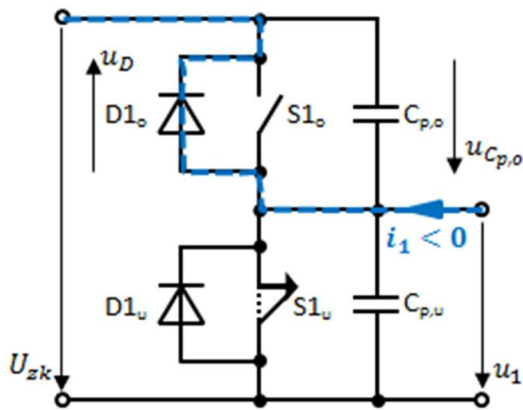


Bild 6.12: $t_1 \leq t < t_{1v}$

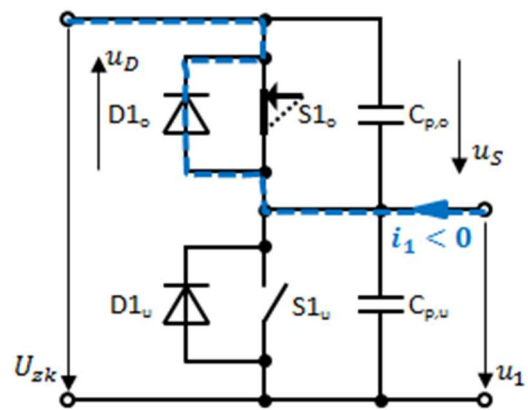


Bild 6.13: $t_{1v} \leq t < t_2$

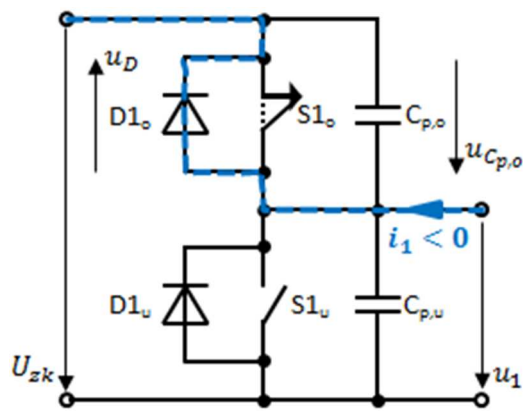


Bild 6.14: $t_2 \leq t < t_{2v}$

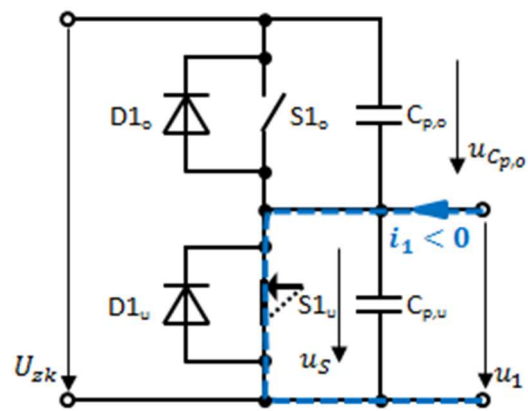


Bild 6.15: $t_{2v} \leq t < T_a$

Die Analyse des Verhaltens der Halbbrücke bei negativem Strom i_1 ist im Bild 6.10 bis Bild 6.15 graphisch abgebildet. Dabei beschreibt Bild 6.10 den Spannungspuls u_1 am Ausgang des Umrichters sowie Bild 6.11 bis Bild 6.15 die zum jeweiligen Zeitintervall zugehörigen Schalterzustände der Halbbrücke.

a) Schaltzustand während dem Zeitintervall $0 \leq t < t_1$ (siehe Bild 6.11)

In diesem Zeitintervall ist der untere Schalter (S_{1u}) geschlossen und der negative Strom i_1 fließt über S_{1u} . Die Spannung am Pulswechselrichterausgang beträgt $u_1 = u_S(i_1)$. Damit ist die untere parasitäre Kapazität ($C_{p,u}$) auch auf diese Spannung geladen. $C_{p,o}$ ist auf $u_{C_{p,o}} = U_{zk} - u_S(i_1)$ aufgeladen. In diesem Fall hat C_p keinen relevanten Einfluss auf u_1 .

b) Schaltzustand während dem Zeitintervall $t_1 \leq t < t_{1v}$ (siehe Bild 6.12)

Zum Zeitpunkt t_1 öffnet der untere Schalter (S_{1o}) und der negative Strom i_1 wechselt seinen Pfad auf die obere Freilaufdiode (D_{1o}). Die Spannung am Ausgang des Spannungszwischenkreisumrichters beträgt nun $u_1 = U_{zk} + u_D(i_1)$. Wie aus der obigen Diskussion zum positiven Strom i_1 bekannt ist, wird beim Wechsel des Stromes von IGBT zur Freilaufdiode die parasitäre Kapazität nicht unverzüglich geladen oder entladen. In [1] lässt sich durch Messungen nachweisen, wie das Abschalten des IGBT zu einem Anstieg der Spannung u_1 führt. Die Messergebnisse sind im Bild 6.16 dargestellt. Die Abhängigkeit von der Stromhöhe im Funktionsverlauf von u_1 ist ersichtlich. Die untere parasitäre Kapazität wird auf die Spannung $u_1 = U_{zk} + u_D(i_1)$ geladen und die obere parasitäre Kapazität auf $u_{C_{p,o}} = -u_D(i_1)$ entladen.

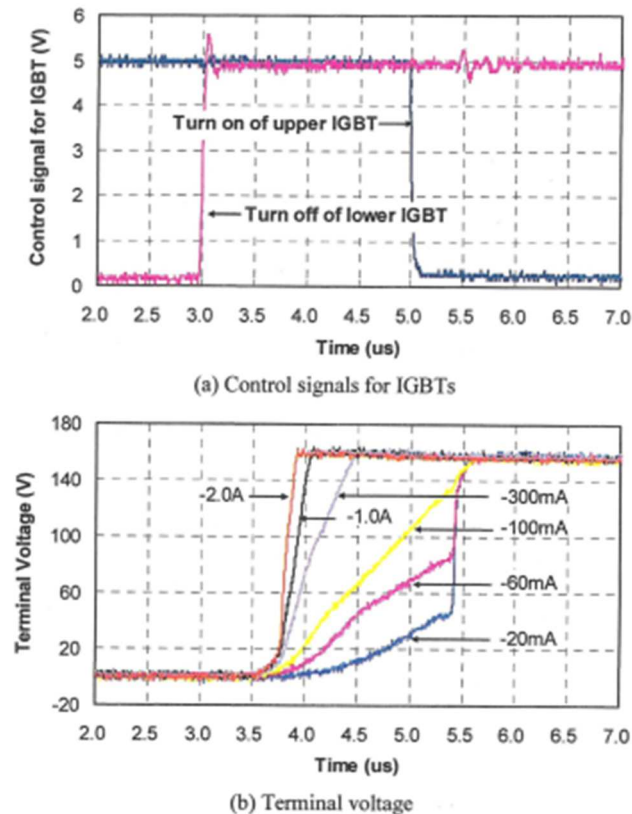


Bild 6.16: Gemessener Anstieg der Ausgangsspannung u_1 aus [1]

c) Schaltzustand während dem Zeitintervall $t_{1v} \leq t < t_2$ (siehe Bild 5.10)

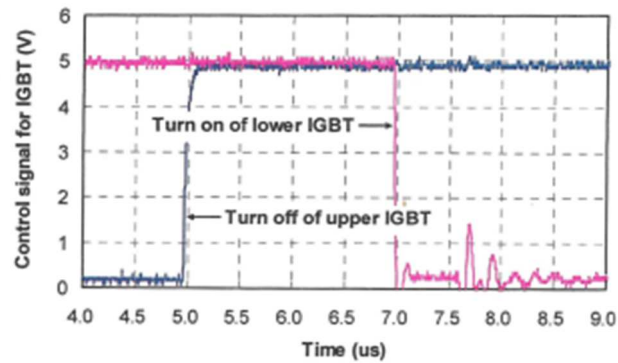
Am Anfang des Zeitabschnitts t_{1v} ist der obere Schalter ($S1_o$) geschlossen und der Strom fließt weiterhin über die obere Freilaufdiode ($D1_o$). Die Spannung am Ausgang des Spannungszwischenkreisumrichters beträgt $u_1 = U_{zk} + u_D(i_1)$. Die Spannungen an den parasitären Kapazitäten bleiben gleich wie in b). Durch das Einschalten von $S1_o$ ist im Bild 6.16 ersichtlich, dass bei kleinen Strömen die Spannung u_1 dadurch prompt auf $u_1 = U_{zk} + u_D(i_1)$ springt.

d) Schaltzustand während dem Zeitintervall $t_2 \leq t < t_{2v}$ (siehe Bild 6.14)

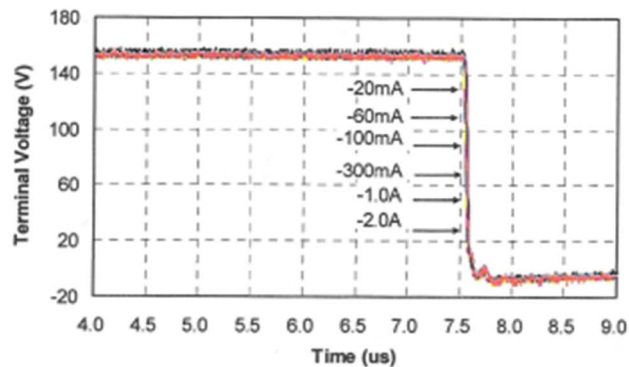
Zu Beginn dieses Intervalls t_2 öffnet der obere Schalter. Der Pfad des Stromes bleibt gleich über die obere Freilaufdiode ($D1_o$) sowie die Spannung am Umrichterausgang $u_1 = U_{zk} - u_D(i_1)$. Die Spannungen an den parasitären Kapazitäten bleiben somit auch unverändert.

e) Schaltzustand während dem Zeitintervall $t_2 \leq t < T_a$ (siehe Bild 6.15)

Zum Zeitpunkt t_{2v} ist der untere Schalter ($S1_u$) geschlossen und der negative Strom i_1 wechselt den Pfad von der oberen Freilaufdiode ($D1_o$) zum unteren IGBT ($S1_u$). Damit wird die Spannung wieder zu $u_1 = u_S(i_1)$. Beim Einschalten des IGBT erfolgt der Sprung von $u_1 = U_{zk} + u_D(i_1)$ auf $u_1 = u_S(i_1)$ ohne Beeinflussung der parasitären Kapazität. Aus [1] ist auch für diesen Fall eine Messung zum Beweis im Bild 6.17 dargestellt.



(a) Control signals for IGBTs



(b) Terminal voltage

Bild 6.17: Messung des Abstieges von u_1 aus [1]

6.1.3 Strom null

Wie im vorhergehenden Kapitel muss auch hier der Fall untersucht werden, wenn die Strommittelwerte \bar{i}_1 zwar größer bzw. kleiner als null sind, dies jedoch für die zugehörigen Momentanwerte i_1 nicht zutrifft. Im Bild 6.18 ist die Ausgangsspannung u_1 des Spannungszwischenkreisumrichters zu sehen. Der Strommittelwert ist dabei null. Wie aus dem Stromverlauf i_1 ersichtlich, sind die Werte in der ersten Hälfte der Pulsperiode negativ und in der zweiten positiv. Damit wird mittels der vorhergehenden Untersuchung klar, dass die parasitäre Kapazität C_p auf beide Verriegelungszeiten Einfluss hat.

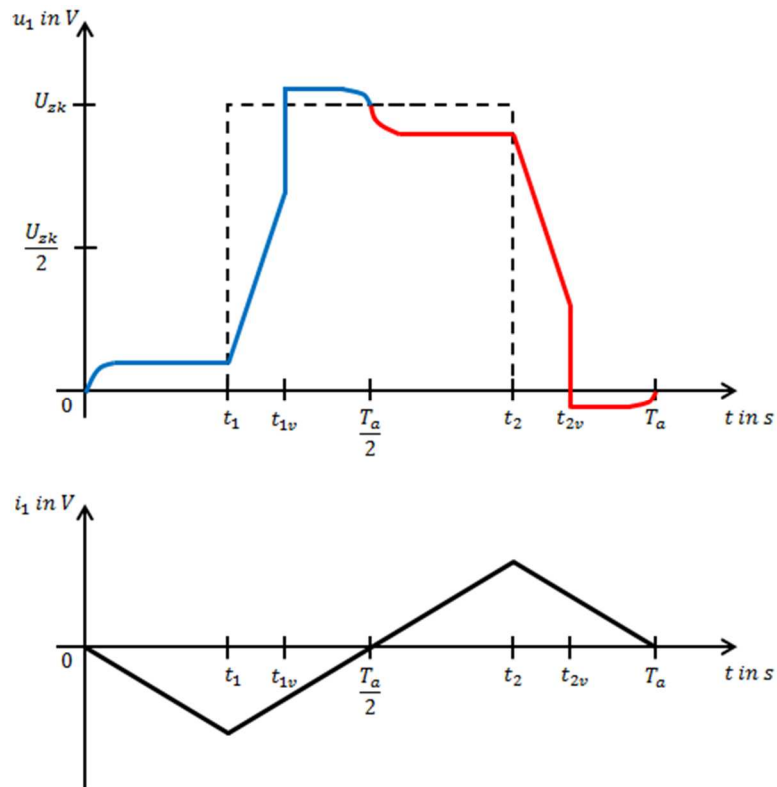


Bild 6.18: PWM-Puls bei $\bar{i}_1 = 0$ mit Berücksichtigung der parasitären Kapazität

6.2 Modellerweiterung für die Berücksichtigung der parasitären Kapazität

Aus den vorhergehenden Betrachtungen lässt sich zusammenfassend ableiten, dass die parasitäre Kapazität für positive Ströme in der Verriegelungszeit nach dem Schaltzeitpunkt t_2 wirksam ist. Für negative Ströme ist sie dies in der Totzeit nach dem Schaltzeitpunkt t_1 . Um diesen Einfluss der parasitären Kapazität im Modell zu berücksichtigen, wird während diesen Zeitintervallen C_p zugeschaltet.

Im nächst Abschnitt wird das mathematische Modell für die Berücksichtigung der parasitären Kapazität hergeleitet und im Anschluss die Implementierung in das MATLAB/Simulink-Modell gezeigt.

6.2.1 Mathematisches Modell mit der parasitären Kapazität

Für die Herleitung des mathematischen Modells, wird der Fall mit $i_1 > 0$ betrachtet. Aus Bild 6.1 und Bild 6.6 ergibt sich folgendes Ersatzschaltbild für die Halbbrücke mit der ohmsch-induktiven Last (siehe Bild 6.19).

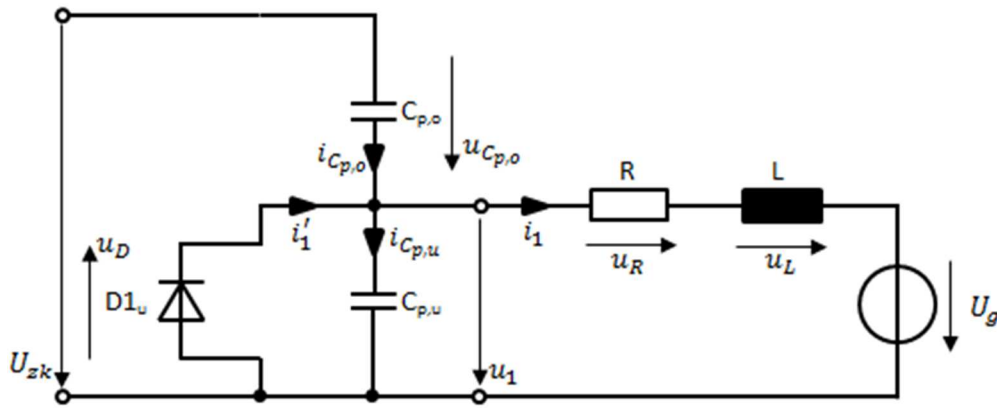


Bild 6.19: Ersatzschaltbild der Halbbrücke mit Last

Für die ohmsch-induktive Last ergibt sich wie in 3.4.1 berechnet, die Differentialgleichung (3.6)

$$\frac{di_1}{dt} = \frac{1}{L} \cdot (-i_1 \cdot R + u_1 - U_g). \quad (6.1)$$

Aus der Knotenregel für die Ströme lässt sich die Gleichung

$$i_1 + i_{C_{p,u}} - i'_1 - i_{C_{p,o}} = 0 \quad (6.2)$$

ableiten. Die Bauteilgleichungen für die parasitären Kapazitäten lauten

$$i_{C_{p,o}} = C_{p,o} \frac{du_{C_{p,o}}}{dt} \quad (6.3)$$

$$i_{C_{p,u}} = C_{p,u} \frac{du_1}{dt}. \quad (6.4)$$

Die Spannung an der unteren parasitären Kapazität entspricht der Spannung am Ausgang des Spannungszwischenkreisumrichters $u_1 = -u_D(i_1)$. Für die obere parasitäre Kapazität kann die Spannung $u_{C_{p,o}} = U_{zk} - u_1$ ausgedrückt werden und somit wird (6.3) zu

$$i_{C_{p,o}} = C_{p,o} \frac{dU_{zk}}{dt} - C_{p,o} \frac{du_1}{dt}. \quad (6.5)$$

Da die Zwischenkreisspannung konstant ist, fällt der Term mit U_{zk} weg und die Gleichung (6.5) wird zu

$$i_{C_{p,o}} = -C_{p,o} \frac{du_1}{dt}. \quad (6.6)$$

Mit den Gleichungen (6.4) und (6.6), eingesetzt in die Gleichung (6.2) erhält man

$$i_1 + C_{p,u} \frac{du_1}{dt} - i_1' + C_{p,o} \frac{du_1}{dt} = 0. \quad (6.7)$$

Für die parasitären Kapazitäten gilt $C_p = C_{p,u} = C_{p,o}$. Damit ergibt sich aus (6.7) die Differentialgleichung

$$\frac{du_1}{dt} = \frac{1}{2 \cdot C_p} (-i_1 + i_1'). \quad (6.8)$$

Für einen negativen Strom i_1 wird das Ersatzschaltbild von Pulswechselrichter und Last aus Bild 6.1 und Bild 6.4 zusammengesetzt. Es unterscheidet sich im Vergleich zu positiven Strom aus Bild 6.19 dadurch, dass die obere Freilaufdiode $D1_o$ den negativen Strom führt. Daraus ergibt sich das selbe mathematische Modell mit den Differentialgleichungen (6.1) und (6.8).

Interessant ist nun noch, wie der Strom i_1' ermittelt wird. Das allgemeine Ersatzschaltbild für die Berücksichtigung der parasitären Kapazität ist im Bild 6.20 dargestellt.

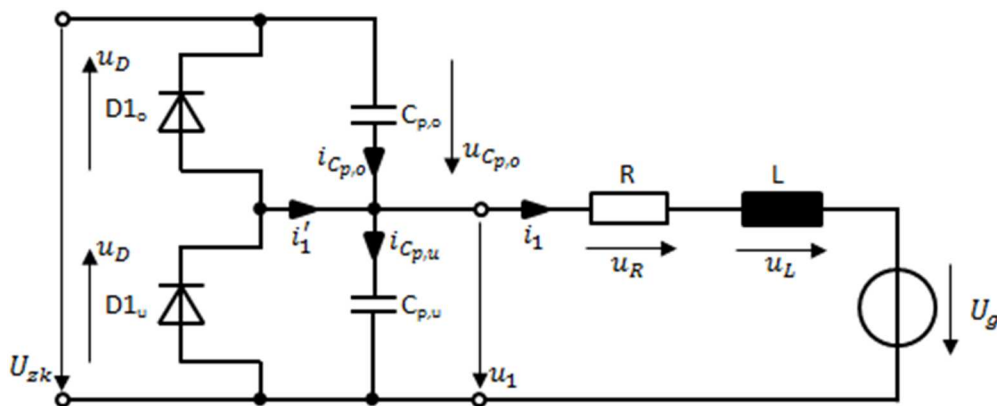


Bild 6.20: Ersatzschaltbild während der Verriegelungszeit mit Berücksichtigung von C_p

Der Strom i_1' fließt je nach Vorzeichen über eine der beiden Freilaufdioden. Die Kennlinie für diesen Schaltzustand ist bekannt und im Bild 5.33 dargestellt. Für die Berücksichtigung der Spannungsabfälle an den Freilaufdioden haben wir, in Abhängigkeit vom Strom i_1 , die Spannung an u_1 mittels einer Lookup-Tabelle bestimmt. Hier gehen wir den umgekehrten Weg und ermitteln in Abhängigkeit von u_1 den Strom i_1' . Im Bild 6.21 ist der Koppelplan für das mathematische Modell zu sehen.

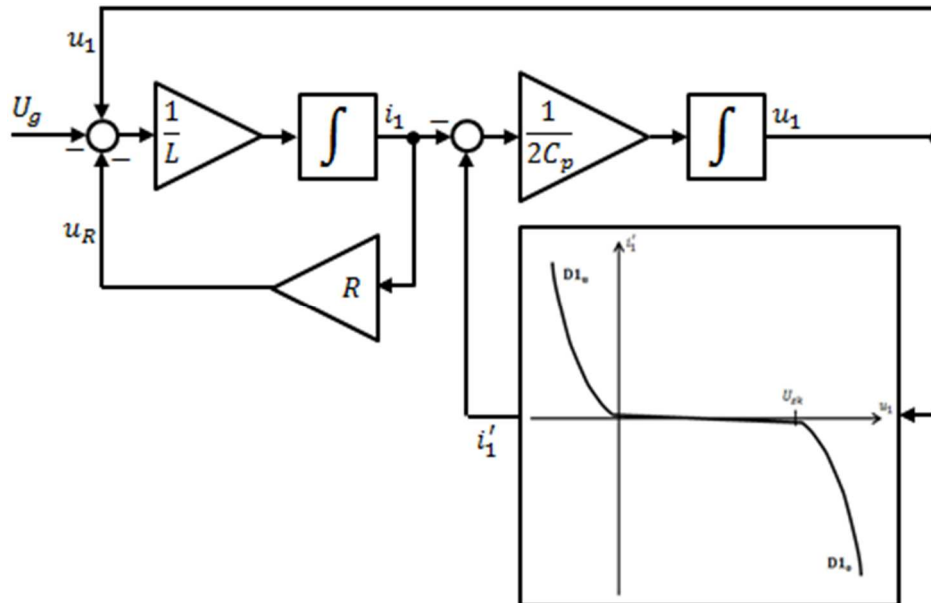


Bild 6.21: Koppelplan für mathematisches Modell mit parasitärer Kapazität

Im folgenden Abschnitt wird die konkrete Einbindung der parasitären Kapazität in das momentane MATLAB/Simulink-Modell beschrieben.

6.2.2 Implementierung in MATLAB/Simulink

Die prinzipielle Struktur des Simulink-Koppelplans mit den Blöcken „PWR“, „ohmsch-induktive Last“ und „Mittelwertmessung“ bleibt bestehen (siehe Bild 4.15). Erweitert wird nur der PWR-Block. Dabei wird das mathematische Modell für die parasitäre Kapazität passend eingefügt. Zusätzlich muss auch die S-Function umprogrammiert werden. Der Koppelplan des Spannungszwischenkreisumrichters ist im Bild 6.22 dargestellt.

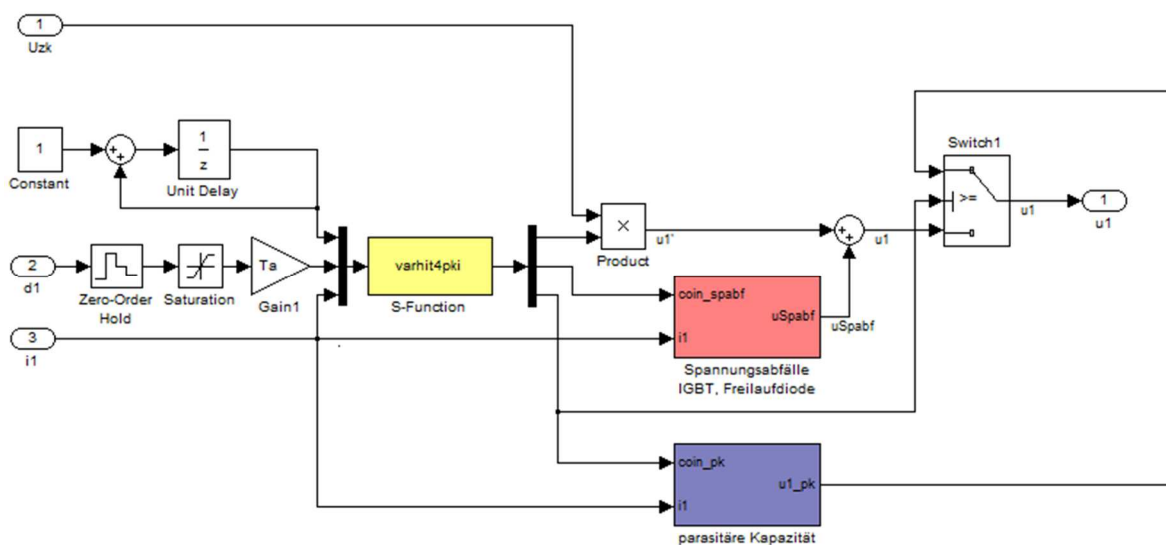


Bild 6.22: MATLAB/Simulink-Koppelplan für Spannungszwischenkreisumrichter-Block

Im Vergleich zu Bild 5.36 ist ein zusätzlicher Block für die Berücksichtigung der parasitären Kapazität zu sehen. Die S-Function hat einen zusätzlichen Eingang i_1 bzw.

Ausgang „coin_pk“ (*control input* für parasitäre Kapazität). Neu ist auch der Switch-Block, der bestimmt, welche Spannung u_1 am Ausgang auf die Last geschaltet wird. Es folgen nun genauere Auskünfte über die Modifizierungen.

S-Function

Für das Ausgangssignal „coin_pk“ wird der Strom i_1 benötigt. Dadurch erhält die S-Function einen dritten Eingang. Aus den obigen Diskussionen in 6.1.1 und 6.1.2 geht hervor, dass die parasitäre Kapazität nur in bestimmten Zeitabschnitten Auswirkungen auf die Spannung u_1 hat. Bei positivem Strom i_1 ist dies im Intervall $t_2 \leq t < t_{2v}$ und für negativen Strom in $t_1 \leq t < t_{1v}$. Um die beiden Fälle klar unterscheiden zu können wird „coin_pk“ für $i_1 > 0$ auf 1 bzw. für $i_1 < 0$ auf 2 gesetzt. Für die restliche Zeit in der Pulsperiode T_a bleibt „coin_pk“ auf null.

In Tabelle 3 ist der Ablauf während drei Pulsperioden für Eingänge ($u1$, $u2$ und $u3$), Ausgänge (sys) und Zustände ($x1$, $x2$) in der S-Function dokumentiert. In der sys -Spalte stehen in der Zeile für Output die drei Ausgangssignale der S-Function. Der Quellcode von „varhit4pki.m“ ist im Anhang A.5 zu finden.

Block mit parasitärer Kapazität

Im Bild 6.23 ist das mathematische Modell von C_p abgebildet. Im Vergleich mit Bild 6.21 kann der hintere Teil mit der parasitären Kapazität unverändert übernommen werden. Da C_p nur in einer der beiden Verriegelungszeiten tatsächlich wirksam ist, wird mit dem Switch und dem Steuersignal „coin_pk“ das mathematische Modell mit der parasitären Kapazität hinzugeschaltet. Ist „coin_pk“ gleich null, dann bleibt der Integratorausgang „u1_pk“ unverändert, da am Eingang des Integrators null anliegt. Hat „coin_pk“ die Werte 1 bzw. 2, wird das Modell für C_p aktiv. Dies kann aber nur zu den Zeitpunkten t_1 und t_2 passieren. Somit muss gewährleistet werden, dass die Spannung u_1 am Integratorausgang auf den passenden Wert gesetzt wird. Dadurch hat der Integrator zwei zusätzliche Eingänge, einen für den Reset und den anderen für den gewünschten Spannungswert u_1 , auf den der Integrator zum Resetzeitpunkt gesetzt wird. Der externe Reset des Integrators wurde so konfiguriert, dass er auf steigende Flanken reagiert. Wenn als „coin_pk“ von 0 auf 1 oder von 0 auf 2 springt, wird der Reset aktiviert und der Integrator wird auf den Wert des dritten Einganges (x_0) gelegt.

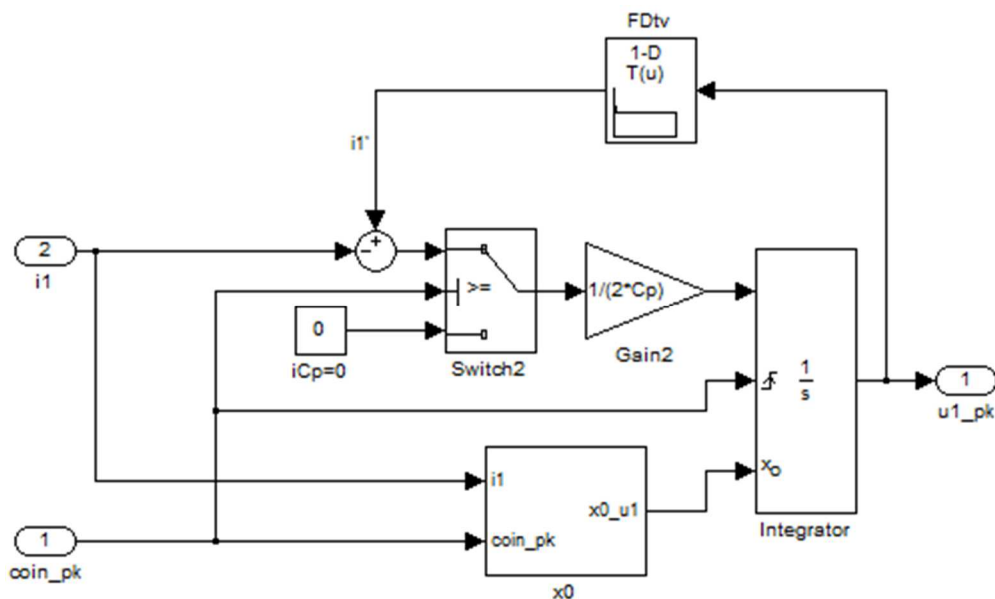


Bild 6.23: MATLAB/Simulink Koppelplan für parasitäre Kapazität

Der Spannungswert für u_1 wird mit dem Block „x0“ bestimmt. Dieser ist im Bild 6.24 dargestellt. In Tabelle 4 wird gezeigt, welchen Anfangswert u_1 in Abhängigkeit vom Stromvorzeichen besitzt. Daraus erkennen wir, dass der Integrator je nach Stromvorzeichen auf verschiedene Anfangswerte gesetzt werden muss.

Tabelle 4: t_0 und u_1 in Abhängigkeit vom Stromvorzeichen

	$i_1 > 0$	$i_1 < 0$
Anfangszeitpunkt t_0	$t_0 = t_2$	$t_0 = t_1$
$u_1(t_0)$	$U_{zk} - u_S(i_1)$	$u_S(i_1)$

Vor den Zeitpunkten bei denen die parasitäre Kapazität aktiv wird, fließt der Strom immer über einen der beiden IGBTs. In der Lookup-Tabelle „IGBT“ ist somit für positive als auch für negative Stromwerte die IGBT-Kennlinie $u_S(i_1)$ abgelegt.

Der Multiport Switch wird über „coin_pk“ gesteuert. Ist „coin_pk“ gleich null, dann ist auch $x0_u1$ gleich null. Für „coin_pk“ ist gleich 1, hat der Strom i_1 ein positives Vorzeichen. Wir erhalten dabei einen negativen Spannungsabfall $-u_S(i_1)$. Durch Addition der Zwischenkreisspannung erhalten wir den gewünschten Wert $x0_u1 = U_{zk} - u_S(i_1)$. Hat coin_pk den Wert 2, ist der Strom negativ. Dadurch erhalten wir einen positiven Spannungsabfall $u_S(i_1)$ als Anfangswert.

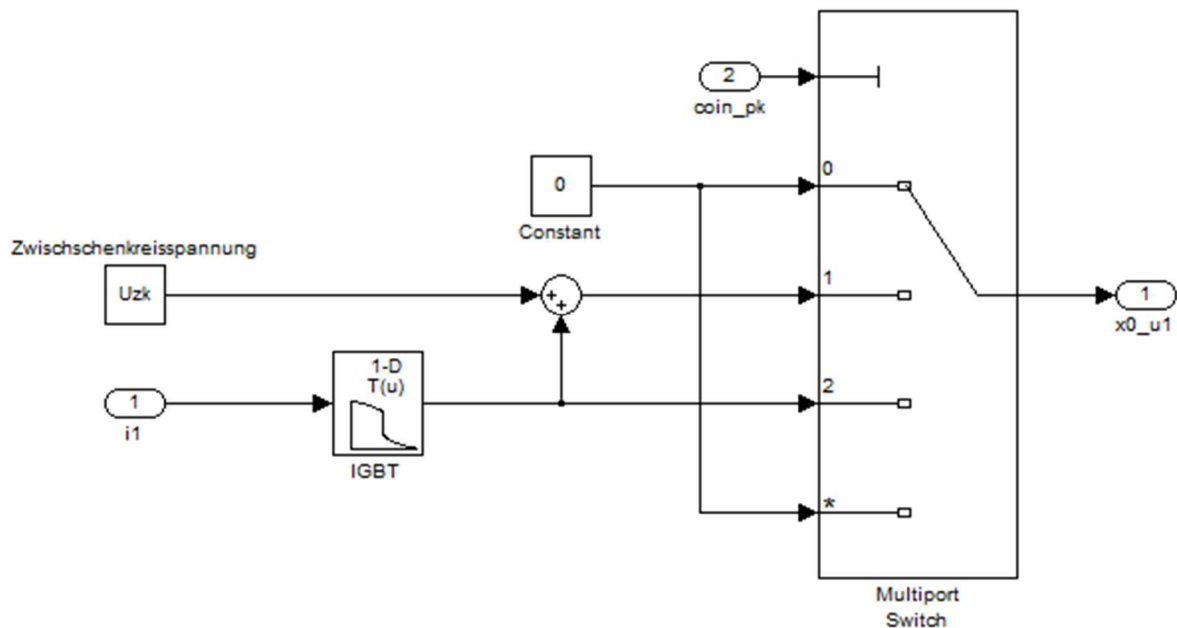


Bild 6.24: Anfangswert $u_1(0)$ für Integrator im Simulationsmodell

Somit ist gesichert, dass zu den Zeitpunkten t_1 und t_2 der Integrator im Bild 6.23 auf den entsprechenden Spannungswert u_1 aus dem vorhergehenden Zeitintervall gesetzt wird.

Mit Bild 6.22 wird nun die Zusammenschaltung des Koppelplans der parasitären Kapazität mit dem Modell aus Kapitel 5 erklärt. Der „Switch“ schaltet zwischen Spannung u_1 aus dem Modell mit Berücksichtigung der Spannungsabfälle und dem Modell mit der parasitären Kapazität. Wie vorhin schon erwähnt, passiert dies abhängig vom Vorzeichen des Stroms zu den Zeitpunkten t_1 und t_2 . Der „Switch“ wird mit dem Steuersignal „coin_pk“ geschaltet. Für „coin_pk“ gleich null, ist das Modell mit den Spannungsabfällen wirksam. Hat das Steuersignal „coin_pk“ den Wert 1 oder 2, dann folgt die Umschaltung auf die Ausgangsspannung der parasitären Kapazität. Mit den „Switch“-Blöcken aus Bild 6.22 und Bild 6.23 wird mit dem Signal „coin_pk“ das Ausgangssignal ($u1_pk$) vom Block „parasitäre Kapazität“ bzw. das Eingangssignal (i_1) zu- bzw. abgeschaltet.

6.2.3 Untersuchung der langen Simulationsdauer

Die ersten Testsimulationen mit diesem Modell führten zu sehr langen Ausführungszeiten (ca. 100 s). Dabei wurden folgende Parameterwerte verwendet: $T_a = 200 \mu\text{s}$, $U_{zk} = 100 \text{ V}$, $R = 0,3 \Omega$, $L = 10\text{mH}$, $t_v = 10 \mu\text{s}$, $R_{tv} = 500 \text{ k}\Omega$, $d_1 = 0,5$ und $U_g = 48 \text{ V}$. Die parasitäre Kapazität wurde willkürlich mit $C_p = 5\text{nF}$ angenommen. Es wurde der Solver „ode45“ mit der maximalen Schrittweite $T_a/2$ bzw. der relativen Toleranz $1e-7$ verwendet. Die Simulation wurde für 1500 Pulsperioden, d.h. 0,3s, durchgeführt.

Aufgrund des Zuschaltens der parasitären Kapazität, besitzt das Modell eine weitere Zustandsgröße. Dies ist die Spannung u_1 am Ausgang des Pulswechselrichters die auch der Spannung an der parasitären Kapazität entspricht. Aus Gleichung (6.1) und (6.8) ergibt sich folgende Zustandsdifferentialgleichungen.

$$\begin{bmatrix} \frac{di_1}{dt} \\ \frac{du_1}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R_{tv}}{L} & \frac{1}{L} \\ \frac{-1}{2 \cdot C_p} & 0 \end{bmatrix} \cdot \begin{bmatrix} i_1 \\ u_1 \end{bmatrix} + \begin{bmatrix} \frac{1}{L} & 0 \\ 0 & \frac{1}{2 \cdot C_p} \end{bmatrix} \cdot \begin{bmatrix} U_g \\ i_1' \end{bmatrix} \quad (6.9)$$

Da die parasitäre Kapazität nur während der Verriegelungszeit zugeschaltet wird, steht in (6.9) R_{tv} statt dem Lastwiderstand R . Damit erhalten wir für die Eigenwerte

$$\lambda_1 \approx -5 \cdot 10^7 \text{ bzw. } \lambda_2 \approx -2 \cdot 10^2 \quad (6.10)$$

Es ist ersichtlich, dass der erste Eigenwert um den Faktor 250000 größer ist als der zweite Eigenwert. Somit handelt es sich hierbei um ein sogenanntes steifes System. Die Induktivität der Last bzw. die parasitäre Kapazität haben ein extrem unterschiedliches Verhalten. Das Laden und Entladen der Kapazität (Spannung u_1) passiert um ein vielfaches (Faktor 250000) schneller als das Ansteigen bzw. Abklingen des Stromes i_1 bei der Induktivität. Der Solver „ode45“ passt die Schrittweite an die Zustandsgröße u_1 an, dadurch wird diese sehr klein. Durch die vielen Berechnungsschritte kommt es zu längeren Ausführungszeiten der Simulation.

Um diese Herausforderung zu bewältigen, stellt MATLAB/Simulink spezielle Solver (ode15s, ode23s, ode23t und ode23tb) für steife Systeme zur Verfügung. Ein Vergleich aller sieben Solver mit variabler Schrittweite ist in Tabelle 5 dokumentiert. Dabei gelten dieselben Parameterwerte wie zuvor, mit jeweils variiertem Solver.

Tabelle 5: Vergleich der Solver

	Simulationsdauer	Anzahl der Stützstellen	\bar{u}_1	\bar{i}_1
<i>ode45</i>	~ 102 s	2621808	48,0384 V	0,12869 A
<i>ode23</i>	~ 91 s	3582333	48,0386 V	0,12871 A
<i>ode113</i>	~ 127 s	5461417	48,0387 V	0,12871 A
<i>ode15s</i>	~ 11 s	357677	48,0386 V	0,12835 A
<i>ode23s</i>	~ 25 s	337552	48,0386 V	0,12871 A
<i>ode23t</i>	~ 14 s	529513	48,0386 V	0,12871 A
<i>ode23tb</i>	~ 14 s	407692	48,0386 V	0,12871 A

Aus dem obigen Vergleich ist ersichtlich, dass die Simulationsergebnisse der Mittelwerte von Spannung und Strom für alle Solver fast identisch sind. Große Unterschiede gibt es in der Simulationsdauer bzw. bei der Anzahl der Stützstellen. Die ersten drei Solver (*ode45*, *ode23* und *ode113*) haben im Vergleich zu den anderen die ca. zehnfache Simulationsdauer.

Für die weiteren Simulationen in dieser Arbeit wurde der Solver *ode15s* ausgewählt, da er die geringste Simulationsdauer besitzt. Mit der relativen Toleranz von $1e-6$ (vorher $1e-7$) wird die Simulationsdauer noch geringer (ca. 7 s) bei gleichbleibender Genauigkeit.

6.3 Testen des Modells mit parasitärer Kapazität

Hier wird nun nachgewiesen, dass die vorhin aufgezeigten Einflüsse der parasitären Kapazität C_p mit diesem Modell simuliert werden können. Um dies anschaulich zu machen, wurden die Verriegelungszeit t_v bzw. die Spannungsabfälle für IGBT und die Freilaufdiode vergrößert. Dadurch werden deren Effekte besser ersichtlich. Der Wert der parasitären Kapazität wird willkürlich auf $C_p = 10\text{nF}$ festgelegt.

6.3.1 Strom positiv

Für den Fall $i_1 > 0$ ist der simulierte PWM-Verlauf im Bild 6.25 abgebildet. Dabei wird im Vergleich mit Bild 6.2 sichtbar, dass das Modell das geforderte Verhalten berücksichtigt. In der Verriegelungszeit $t_1 \leq t < t_{1v}$ ist keine Auswirkung der parasitären Kapazität ersichtlich, da die Ausgangsspannung des Spannungszwischenkreisumrichters unabhängig vom Strom i_1 sprunghaft ansteigt. In der Totzeit $t_2 \leq t < t_{2v}$ wird das Modell für C_p zugeschaltet und der stromabhängige Abstieg der Spannung u_1 ist ersichtlich. Im Bild 6.26 in dieser Zeitabschnitt genauer betrachtet. Daraus leitet sich ab, dass für den Strommittelwert $\bar{i}_1 = 10\text{ A}$ die parasitäre Kapazität praktisch keinen Einfluss mehr hat.

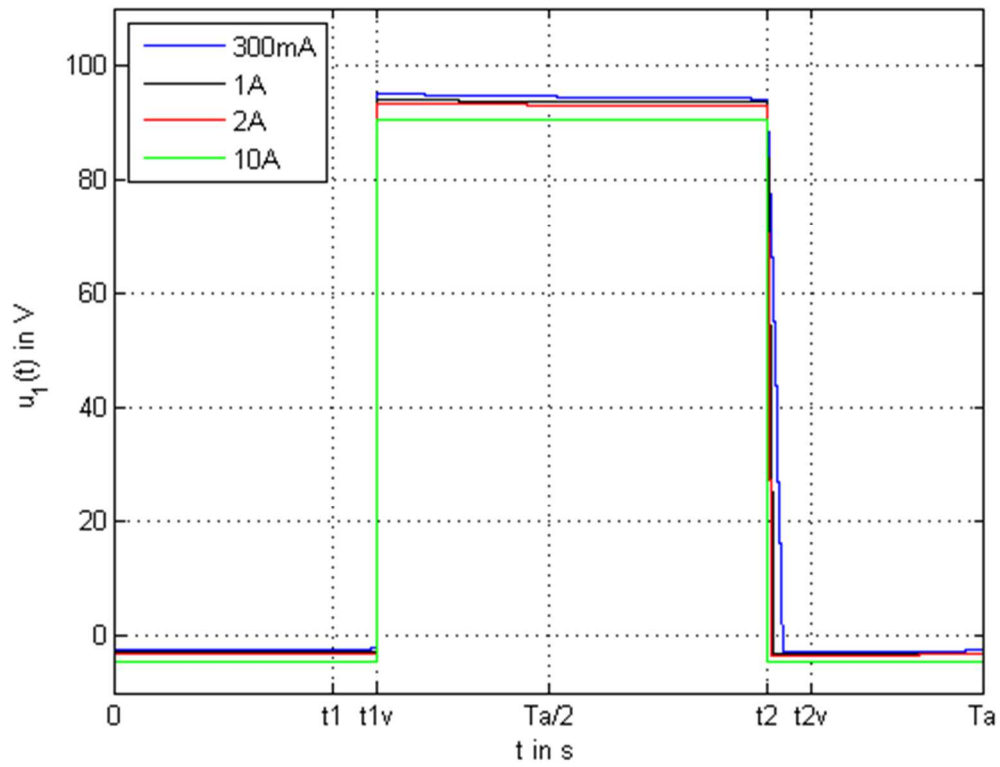


Bild 6.25: Verlauf der Ausgangsspannung u_1 bei positiven Strömen

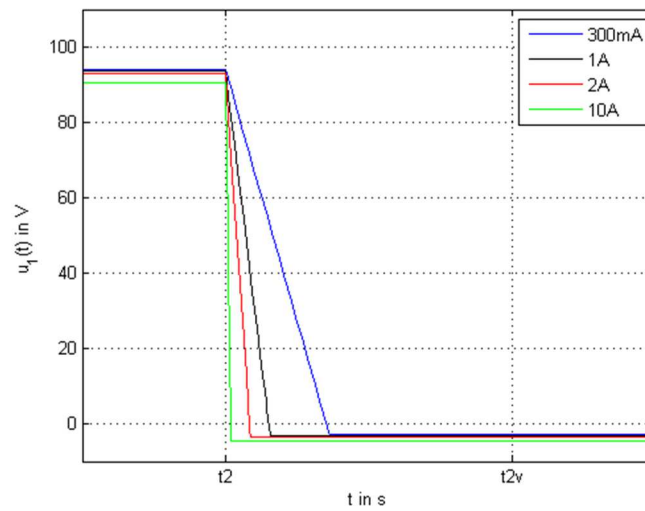


Bild 6.26: Ausschnitt der Verriegelungszeit in dem die parasitäre Kapazität wirksam ist

6.3.2 Strom negativ

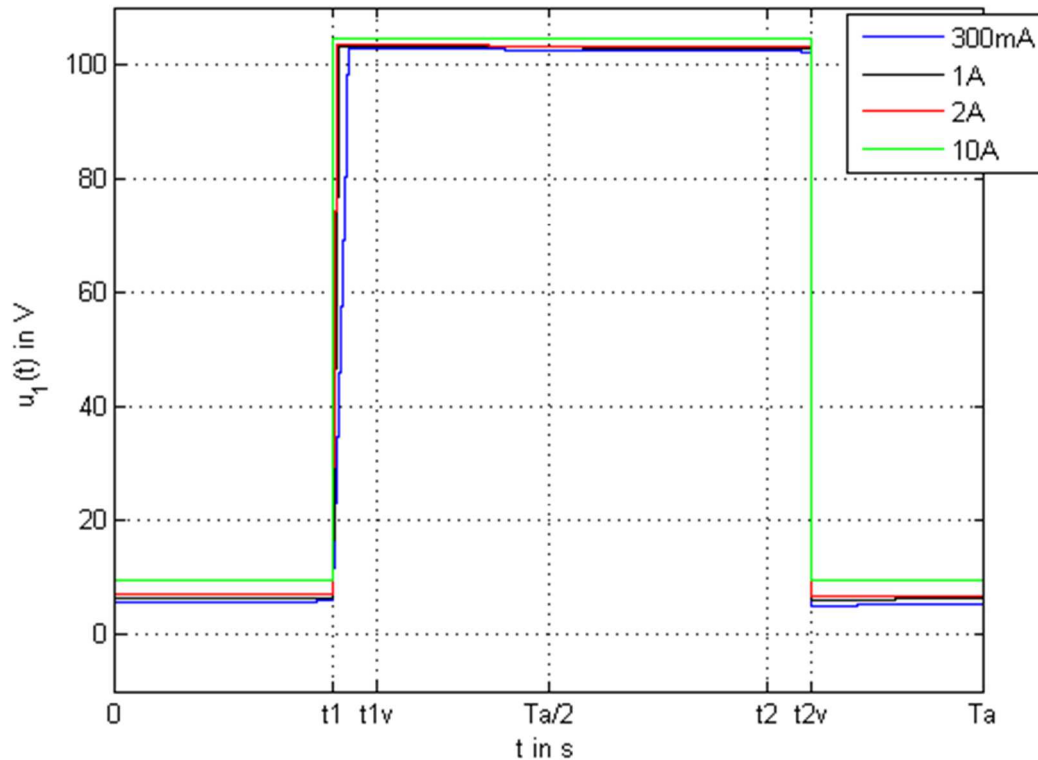


Bild 6.27: Verlauf der Ausgangsspannung u_1 bei negativen Strömen

Im Bild 6.27 ist die Simulation mit Berücksichtigung der parasitären Kapazität für $i_1 < 0$ dargestellt. Verglichen mit Bild 6.10 wird auch hier das geforderte Modellverhalten mit dem Simulationsmodell erreicht. In diesem Fall ist in der Verriegelungszeit $t_2 \leq t < t_{2v}$ keine Auswirkung von C_p zu sehen, d.h. die Spannung u_1 am Pulswechselrichter Ausgang verläuft sprungförmig zum Zeitpunkt t_{2v} . Im Zeitabschnitt $t_1 \leq t < t_{1v}$ ist die parasitäre Kapazität wirksam. Dieser Bereich im PWM-Signal ist im Bild 6.28 herausgehoben und der nicht-sprungförmige Übergang der Spannung u_1 ist ersichtlich. Die Steigung von u_1 ist dabei von der Höhe des Stromes i_1 abhängig.

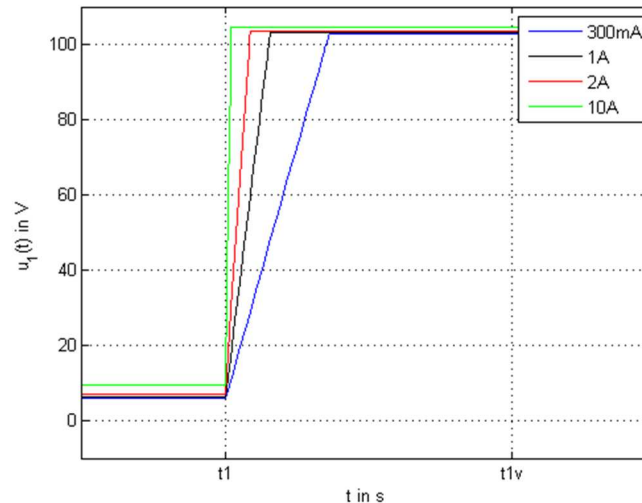


Bild 6.28: Ausschnitt der Verriegelungszeit in dem die parasitäre Kapazität wirksam ist

6.3.3 Strom null

Für den speziellen Fall des Strommittelwerts $\bar{i}_1 = 0$ ist der Funktionsverlauf der Spannung u_1 im Bild 6.29 abgebildet. In diesem Fall sind auch die Strommittelwerte enthalten, bei denen zwar $\bar{i}_1 > 0$ bzw. $\bar{i}_1 < 0$, jedoch die Momentanwerte i_1 nicht zwingend positiv bzw. negativ sein müssen.

In diesem Fall ist in beiden Verriegelungszeiten die parasitäre Kapazität im Modell zugeschaltet. Der Vergleich mit Bild 6.18 bestätigt, dass das Simulationsmodell die gewünschten Verhältnisse nachbilden kann. Ist der Strommittelwert null, sind die Steigungen der Spannung u_1 in den beiden Totzeiten identisch. Für sehr kleine positive Strommittelwerte werden im Zeitintervall $t_1 \leq t < t_{1v}$ die negativen Stromwerte i_1 immer kleiner und dadurch der Übergang von u_1 flacher. In der anderen Verriegelungszeit werden die positiven Stromwerte i_1 größer und damit der Funktionsverlauf von u_1 steiler. Die Verschiebung des Stromnulldurchgangs ist darin auch ersichtlich.

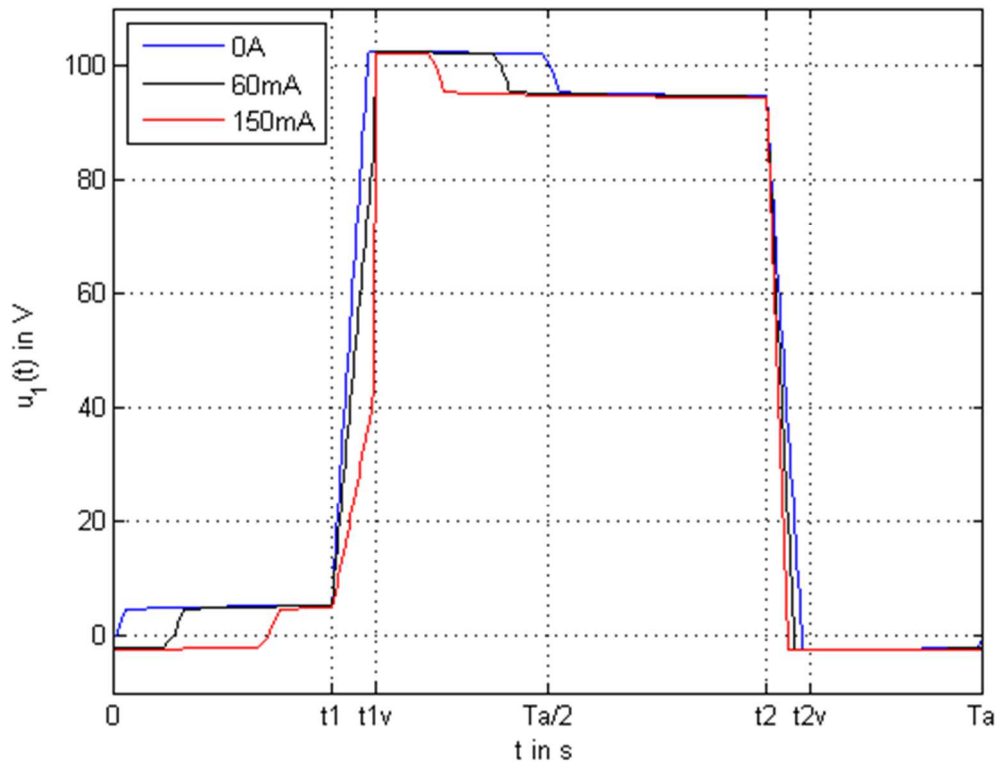


Bild 6.29: Spannungspuls bei kleinen Strömen

6.4 Validierung des Modells mit parasitärer Kapazität

In diesem Abschnitt wird das MATLAB/Simulink-Modell wie in den vorhergehenden Kapiteln mit der Messung verglichen. Zuvor wird die parasitäre Kapazität C_p festgelegt.

6.4.1 Größenordnung der parasitäre Kapazität

Es stellt sich nun die Frage wie der Parameter von der parasitären Kapazität zu wählen ist. Im Datenblatt (siehe Anhang B) ist für den IGBT eine Kapazität $C_{CHC} = 1,4\text{nF}$ angegeben und es steht zusätzlich „AC-side“ dabei. Ob es sich dabei um die parasitäre Kapazität handelt, konnte nicht nachvollzogen werden. Um den Wertebereich von C_p grob einzuschränken und deren Auswirkungen auf die Umrichternichtlinearität aufzuzeigen, kommt es hier zu einer Variation ähnlich wie für die Totzeit. Dabei wurde die Nichtlinearität des Spannungswischenkreisumrichters mit fünf verschiedenen Parameterwerten für die parasitäre Kapazität $C_p = \{1\text{nF}, 5\text{nF}, 10\text{nF}, 15\text{nF}, 20\text{nF}\}$ bestimmt.

6.4.2 Vergleich der Simulation mit der Messung

Der Vergleich zwischen der Messung und den Simulationsergebnissen des mathematischen Modells mit Berücksichtigung der parasitären Kapazität ist im Bild 6.30 dargestellt. Die Verriegelungszeit wurde mit $t_v = 2 \mu\text{s}$ festgelegt und in diesem Vergleich nicht mehr variiert. Es ist ersichtlich, dass die Modellierung der parasitären Kapazität die Abweichungen aus dem vorhergehenden Modell im Bild 5.39 fast zu Gänze eliminiert. Die Variation der parasitären Kapazität macht den speziellen Wert von $C_p \approx 10\text{nF}$ erkenntlich. Aus den simulierten Umrichternichtlinearität ist ersichtlich, dass im Bereich $-10 \text{ A} < \bar{i}_1 < 10 \text{ A}$ der Einfluss von C_p sehr hoch ist. Ausgenommen ist darin der steile Kennlinienverlauf um den Koordinatenursprung. Im Strombereich $|\bar{i}_1| > 10 \text{ A}$, hat die parasitäre Kapazität nur mehr marginale Wirkung und das Modell mit Berücksichtigung der Verriegelungszeit bzw. den Spannungsabfällen an IGBT und Freilaufdiode ist dominant.

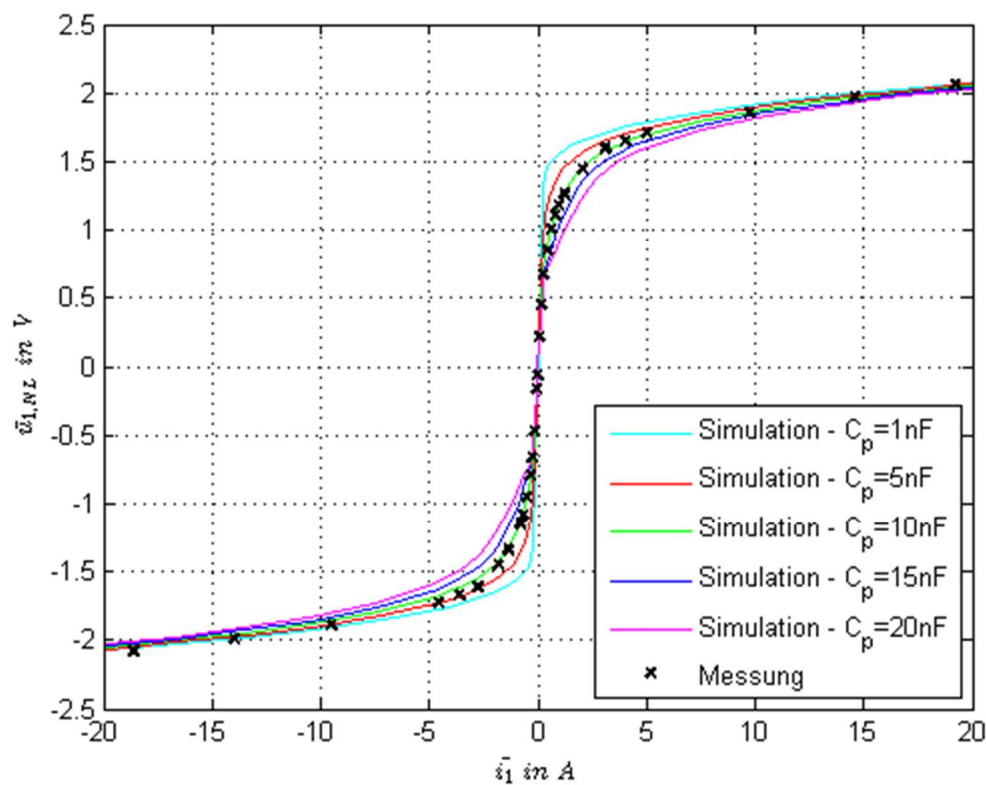


Bild 6.30: Vergleich der Umrichternichtlinearität zwischen Messung und Simulation

Im Bild 6.31 ist der Ausschnitt aus der Umrichternichtlinearität im Strombereich $-5 \text{ A} < \bar{i}_1 < 5 \text{ A}$ abgebildet. Daraus sind die Schlussfolgerungen aus der vorhergehenden Diskussion genauer ersichtlicher.

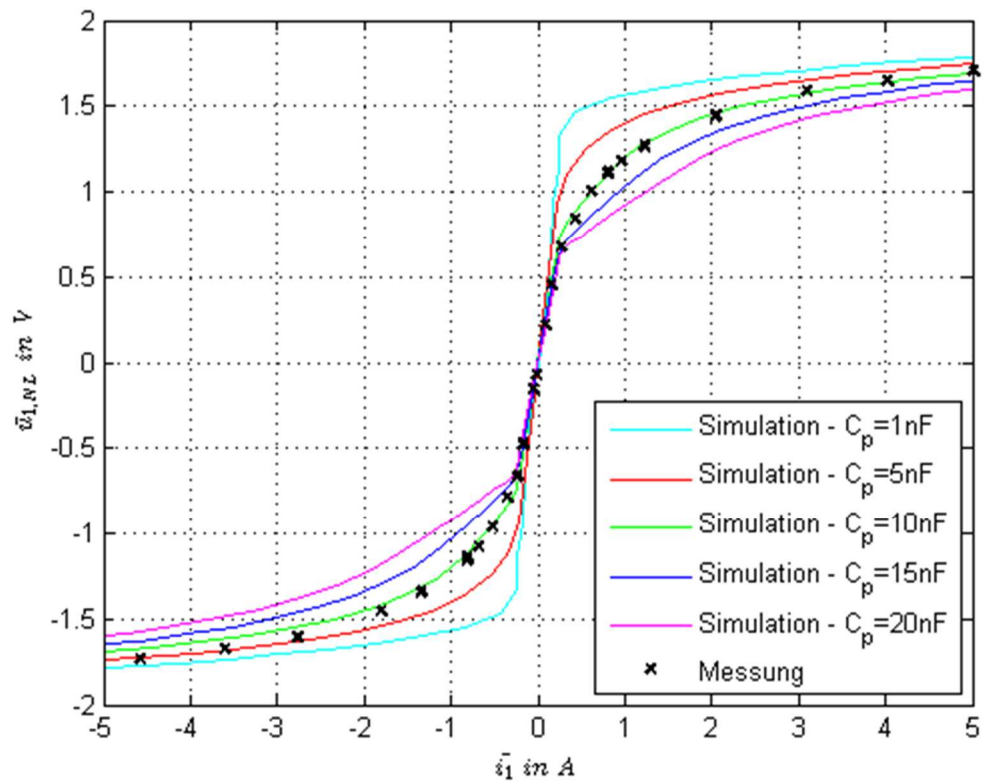


Bild 6.31: Ausschnitt der Umrichternichtlinearität unter Berücksichtigung der parasitären Kapazität

Im nachfolgenden letzten Kapitel dieser Arbeit erfolgen eine kompakte Zusammenfassung der gewonnenen Erkenntnisse und ein Vergleich der durchgeführten Modellierungsschritte.

7 Resümee und Ausblick

Das Ziel dieser Arbeit, ein Simulationsmodell eines Spannungszwischenkreisumrichters in MATLAB/Simulink zu erstellen, wurde mit der Berücksichtigung dreier wesentlicher Einflussfaktoren erfolgreich durchgeführt. Eingehend untersucht wurden dabei die Verriegelungszeit, die Spannungsabfälle an den leistungselektronischen Bauelementen (IGBT bzw. Freilaufdiode) und die parasitäre Kapazität. In die Betrachtung der Verriegelungszeit einbegriffen ist das so genannte Stromlücken.

Ausgangslage für die Modellierung war das ideale Modell eines Spannungszwischenkreisumrichters. Dieses wurde im ersten Schritt um die Verriegelungszeit erweitert. Zusätzlich wurde das Problem des Stromlückens (zero clamping effect) berücksichtigt. Es konnte gezeigt werden, dass der standardmäßige Solver ode45 mit einer ausreichend kleinen relativen Toleranz ($1e-6$) dieses System mit dem Lastwiderstand $R = 0,3 \Omega$ und dem um ein Vielfaches größeren $R_{tv} = 500 \text{ k}\Omega$ (Widerstandsapproximation des Sperrbereichs der beiden Freilaufdioden) effizient simuliert. Damit konnte der nichtlineare Spannungsabfall am Pulswechselrichter grob approximiert werden.

Der Vergleich der Simulationsergebnisse mit der Messung für die drei Zwischenkreisspannungen ($U_{zk} = \{100 \text{ V}, 200 \text{ V}, 300 \text{ V}\}$) ist im Bild 7.1 bis Bild 7.3 dargestellt. Im jeweiligen Bild ist die Umrichternichtlinearität im Strombereich $-20 \text{ A} \leq \bar{i}_1 \leq 20 \text{ A}$ und der Übergang durch den Ursprung dargestellt. In den unteren Subplots ist der Einfluss der Verriegelungszeit bzw. des Stromlückens deutlich zu sehen. Daraus ist auch die Größe des Stromrippels ersichtlich, dieser beträgt für $U_{zk} = 100 \text{ V}$ $0,5 \text{ A}$, für $U_{zk} = 200 \text{ V}$ 1 A und für $U_{zk} = 300 \text{ V}$ $1,5 \text{ A}$. Der Stromripple wird dabei als Spitze-Spitze-Wert angegeben ($i_{1,pp}$).

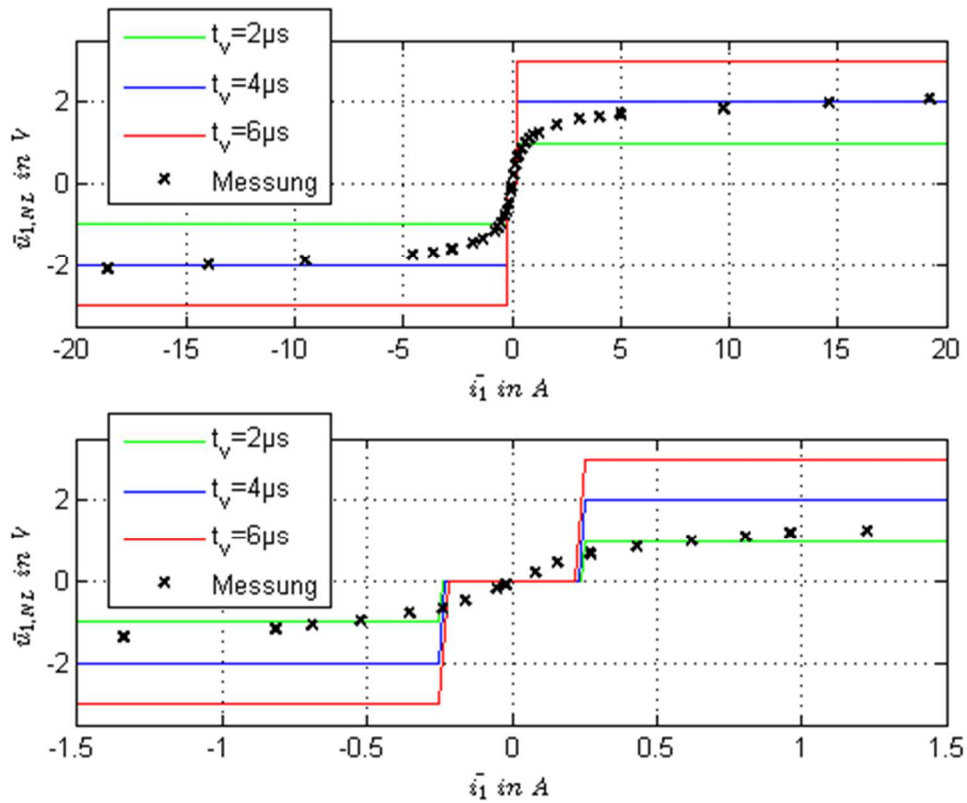


Bild 7.1: Simulationsergebnisse bzw. Messung der Umrichternichtlinearität bei $U_{zk} = 100$ V

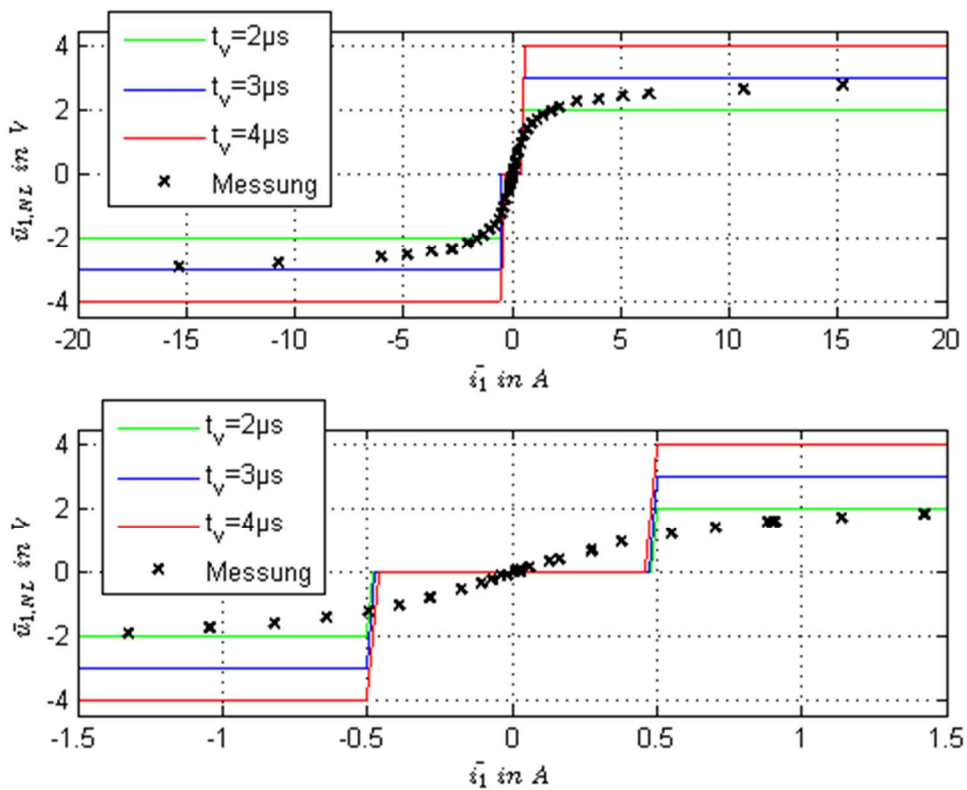


Bild 7.2: Simulationsergebnisse bzw. Messung der Umrichternichtlinearität bei $U_{zk} = 200$ V

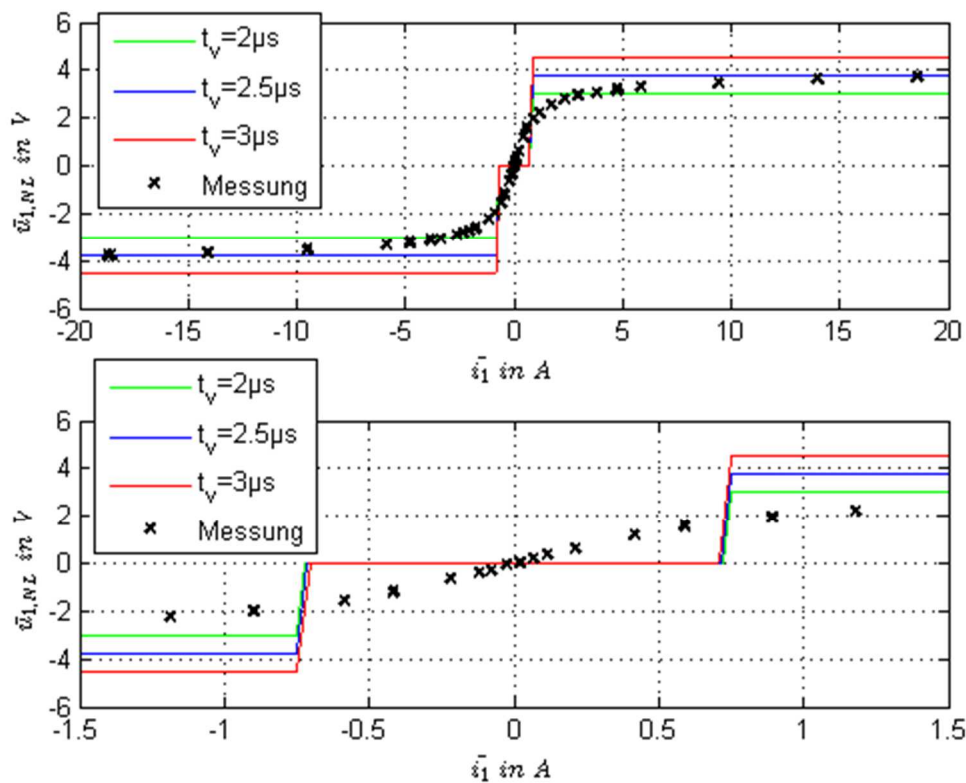


Bild 7.3: Simulationsergebnisse bzw. Messung der Umrichternichtlinearität bei $U_{zk} = 300 \text{ V}$

Im Bild 7.2 ist der Stromrippel $i_{1,pp} = 1 \text{ A}$ ($U_{zk} = 200 \text{ V}$). In der unteren Abbildung ist ersichtlich, dass die Verriegelungszeit im Strombereich $-0,5 \text{ A} < \bar{i}_1 < 0,5 \text{ A}$ keinen Spannungsabfall verursacht. In diesem Fall haben aufgrund des Stromrippels der Momentanwertverlauf i_1 positive bzw. negative Werte (Stromnulldurchgänge). Damit hat der Strom für beide Verriegelungszeiten unterschiedliches Vorzeichen und es tritt kein Spannungsabfall, verursacht durch die Totzeit, auf. Außerhalb dieses Strombereichs ist der Stromrippel vollständig über bzw. unter der Zeitachse ($i_1 = 0 \text{ A}$) und damit ist der vollständige Einfluss der Verriegelungszeit ersichtlich. Den Übergang zwischen diesen beiden Fällen bildet der steile Anstieg vor $\bar{i}_1 = 0,5 \text{ A}$. In diesem Bereich tritt das Stromlücken zu Tage, da der Augenblickswertverlauf i_1 des Stromes null wird.

Im zweiten Schritt der Modellerweiterung konnte mit der Berücksichtigung der Spannungsabfälle am IGBT und der Freilaufdiode eine wesentlich bessere Approximation der Umrichternichtlinearität erreicht werden. Die Modifikation des Modells wurde mit der Einbindung der Durchlasskennlinien von IGBT und Freilaufdiode einfach durchgeführt. Der Sperrbereich konnte somit vernachlässigt werden. Eine besondere Herausforderung stellte es dar, die passenden Kennlinien für die beiden leistungselektronischen Bauteile zu bekommen, da die Datenblätter im Strombereich $\pm 20 \text{ A}$ keine Daten enthielten. Die beiden Durchlasskennlinien konnten somit aus den vorhandenen Messungen ermittelt.

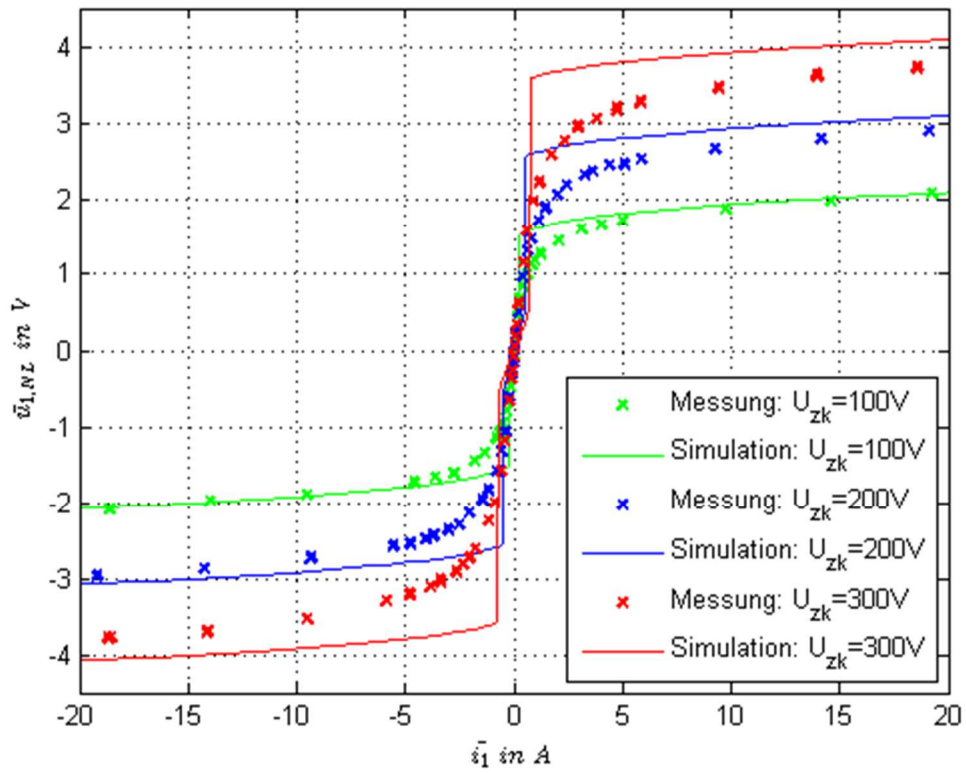


Bild 7.4: Messungen und Simulationsergebnisse für $U_{zk} = \{100 \text{ V}, 200 \text{ V}, 300 \text{ V}\}$ mit $t_v = 2 \mu s$

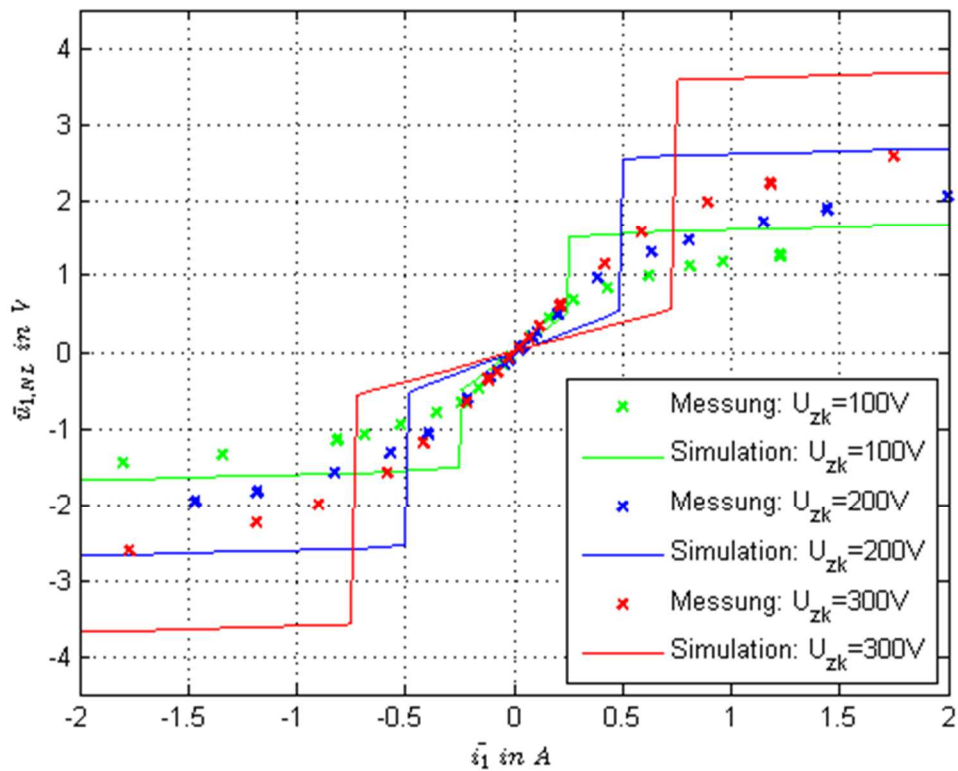


Bild 7.5: Übergangsbereich der Messungen und Simulationsergebnisse für $U_{zk} = \{100 \text{ V}, 200 \text{ V}, 300 \text{ V}\}$

Im Bild 7.4 und Bild 7.5 sind die Simulationsverläufe der Umrichternichtlinearität im direkten Vergleich mit den Messungen zusammengefasst. Aus Bild 7.4 ist ersichtlich, dass mit Berücksichtigung der Spannungsabfälle an IGBT und Freilaufdiode, der Kurvenverlauf der Messungen bereits passabel nachgebildet werden kann ($\bar{i}_1 < -10$ A bzw. $\bar{i}_1 > 10$ A). Die Verriegelungszeit beträgt dabei $t_v = 2 \mu\text{s}$. Bei den Zwischenkreisspannungen für 200 V und 300 V sind Abweichungen vom Messverlauf zu erkennen. Da die Durchlasskennlinien aus den Messungen mit $U_{zk} = 100$ V bestimmt wurden wird vermutet, dass die Kennlinien in diesem Strombereich von der Zwischenkreisspannung abhängig sind. Dadurch ergeben sich die vorhin erwähnten Abweichungen.

Im Übergangsbereich der Umrichternichtlinearität, dargestellt im Bild 7.5, ist im Vergleich zum vorhergehenden Modell der zusätzliche Spannungsabfall ersichtlich. Markant zu sehen ist auch der steile Anstieg bzw. Abfall im Kurvenverlauf aufgrund der Verriegelungszeit.

In der letzten Erweiterung des Modells wurde die parasitäre Kapazität implementiert. Aufgrund der parasitären Kapazität im Umrichtermodell, wird der Rechenaufwand größer und die Simulation dauert länger (steifes System). Um die Simulationsdauer zu senken, wurden die Simulationen in diesem Modell mit dem Solver ode15s (relative Toleranz $1e-6$) durchgeführt. Weiters wurde das Modell so strukturiert, dass die parasitäre Kapazität nur dann zugeschaltet wird, wenn ihr Einfluss nicht vernachlässigbar ist. Dies ist nur während der Verriegelungszeiten der Fall. In welcher der beiden Verriegelungszeiten dies geschieht, hängt vom Stromvorzeichen ab.

Der zusammengefasste Vergleich der Messung mit den Simulationen ist für das entwickelte Modell im Bild 7.6 bis Bild 7.8 dargestellt. Mit der Berücksichtigung der parasitären Kapazität konnten die verbliebenen Abweichungen der Simulation von der Messung im Strombereich ± 5 A eliminiert werden. Die Bereiche mit großer Steigung im simulierten Verlauf der Kennlinie aufgrund der Verriegelungszeit sind damit nicht mehr vorhanden. Durch die Variation der Zwischenkreisspannung kommt es in der Simulation, bei gleichbleibender parasitärer Kapazität $C_p = 10\text{nF}$, zu Abweichungen gegenüber den Messkurven. Durch Veränderung der parasitären Kapazität konnte der Simulationsverlauf an die Messung angepasst werden. Für die Simulationen mit unterschiedlichen Zwischenkreisspannungen ergeben sich die parasitären Kapazitäten $C_{p,100} = 10\text{pF}$, $C_{p,200} = 7\text{pF}$ und $C_{p,300} = 6\text{pF}$.

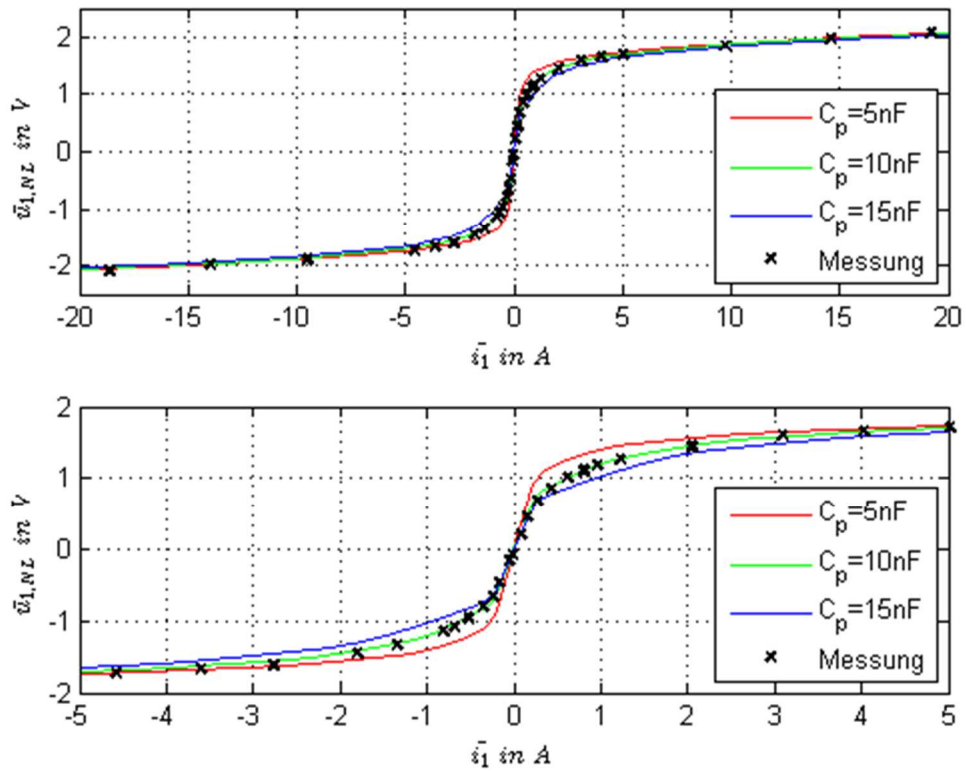


Bild 7.6: Simulationsergebnisse bzw. Messung der Umrichternichtlinearität bei $U_{zk} = 100\text{ V}$

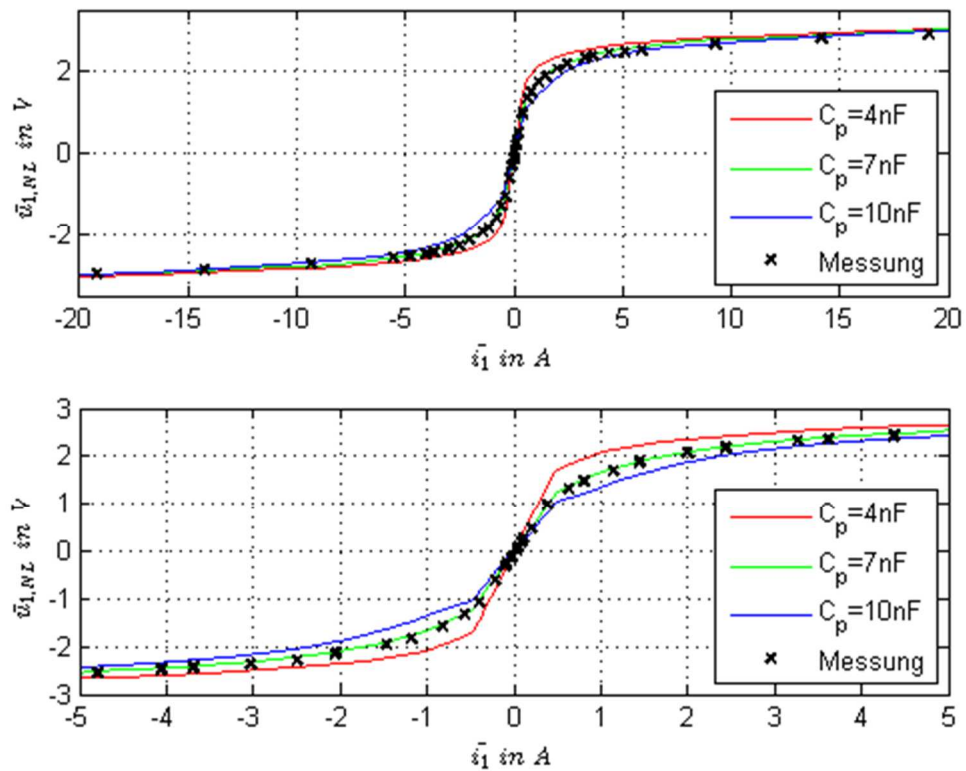


Bild 7.7: Simulationsergebnisse bzw. Messung der Umrichternichtlinearität bei $U_{zk} = 200\text{ V}$

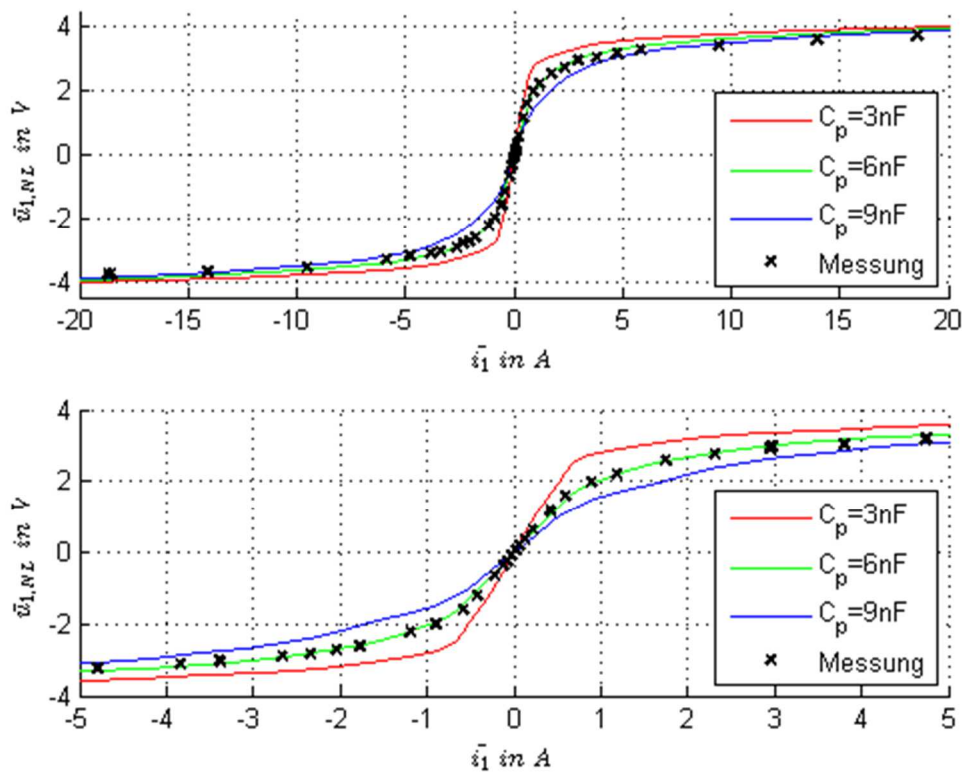


Bild 7.8: Simulationsergebnisse bzw. Messung der Umrichternichtlinearität bei $U_{zk} = 300$ V

Aufbauend auf den gewonnenen Erkenntnissen dieser Arbeit eröffnen sich neue Aufgaben zur weiteren Forschung am Modell des Spannungszwischenkreisumrichters. Besonders wichtig ist hier eine Erweiterung auf alle drei Halbbrücken. Damit lässt sich ein dreiphasiger Spannungszwischenkreisumrichter mit Berücksichtigung der Umrichternichtlinearität simulieren. Da eine Simulation momentan, für einen Zeitraum von 0,3 Sekunden, ca. 7 Sekunden dauert, böte ein „Level 2 S-Function“, eventuell die Möglichkeit die Ausführungszeit zu verkürzen. Im zukünftigen Simulationsmodell ist es wichtig, das Absinken der Zwischenkreisspannung für steigende Ströme zu berücksichtigen. Um dem entgegenzuwirken, wurde in dieser Arbeit für jeden Messpunkt die Zwischenkreisspannung auf den geforderten Wert nachgeregelt. Dadurch konnte in der Simulation die Zwischenkreisspannung konstant vorgegeben werden. Zudem ist es interessant, den Einfluss von Parameterschwankungen (Durchlasskennlinien, parasitäre Kapazität) und Temperatur auf das Modell zu untersuchen. Somit könnte eine noch exaktere Modellierung erreicht werden.

Literaturverzeichnis

- [1] Gong, L. M. ; Zhu, Z.Q.: »Modeling and Compensation of Inverter Nonlinearty Effects in Carrier Signal Injection-Based Sensorless Control Methods from Positive Sequence Carrier Current Distortion«. In: *Energy Conversion Congress and Exposition (ECCE)*, Atlanta, 2010

- [2] Hwang, Sean-Hwan ; Kim, Jang-Mok: »Dead Time Compensation Method for Voltage-Fed PWM Inverter«. In: *IEEE Transactions on Energy Conversion*, Bd. 25, Nr. 1, März 2010

- [3] Jeong, Seung-Gi ; Park, Min-Ho: »The Analysis and Compensation of Dead-Time Effects in PWM Inverters«. In: *IEEE Transactions on Industrial Electronics*, Bd. 38, Nr. 2, April 1991

- [4] Schröder, Dierk: *Leistungselektronische Schaltungen : Funktion, Auslegung und Anwendung*. 3., überarbeitete und erweiterte Aufl. Berlin : Springer, 2012

- [5] Michel, Manfred: *Leistungselektronik : Einführung in Schaltungen und deren Verhalten*. 5., bearbeitete und ergänzte Aufl. Berlin : Springer, 2011

- [6] Specovius, Joachim: *Grundkurs Leistungselektronik : Bauelemente, Schaltungen und Systeme*. 5., aktualisierte und überarbeitete Aufl. Wiesbaden : Vieweg+Teubner, 2011

- [7] Probst, Uwe: *Leistungselektronik für Bachelors : Grundlagen und praktische Anwendungen*. 2., aktualisierte und erweiterte Aufl. München : Carl Hanser, 2011

- [8] Krischan, Klaus: *Stromrichtertechnik : Skriptum zur Vorlesung*. Graz : Technische Universität Graz, ca. 2004

- [9] Brosch, Peter F.: *Moderne Stromrichterantriebe : Leistungselektronik und Maschinen*. 3., überarbeitete und erweiterte Aufl. Würzburg : Vogel, 1998

- [10] The MathWorks, Inc.: Simulink® : Developing S-Functions. sfunctions.pdf. (2014)

- [11] Schröder, Dierk: *Leistungselektronische Bauelemente*. 2. Aufl. Berlin : Springer, 2006

A Anhang

A.1 MATLAB-Funktion „varhit2ideal.m“

```
1 function [sys,x0,str,ts] = varhit2ideal(t,x,u,flag,Ta)
2
3 % With M-file S-functions, you can define you own ordinary differential
4 % equations (ODEs), discrete system equations, and/or just about
5 % any type of algorithm to be used within a Simulink block diagram.
6 %
7 % The general form of an M-File S-function syntax is:
8 %     [SYS,X0,STR,TS,SIMSTATECOMPLIANCE] = SFUNC(T,X,U,FLAG,P1,...,Pn)
9 %
10 % What is returned by SFUNC at a given point in time, T, depends on the
11 % value of the FLAG, the current state vector, X, and the current
12 % input vector, U.
13 %
14 % FLAG    RESULT          DESCRIPTION
15 % -----
16 % 0       [SIZES,X0,STR,TS] Initialization, return system sizes in SYS,
17 %         initial state in X0, state ordering strings
18 %         in STR, and sample times in TS.
19 % 1       DX              Return continuous state derivatives in SYS.
20 % 2       DS              Update discrete states SYS = X(n+1)
21 % 3       Y               Return outputs in SYS.
22 % 4       TNEXT           Return next time hit for variable step sample
23 %         time in SYS.
24 % 5                       Reserved for future (root finding).
25 % 9       []              Termination, perform any cleanup SYS=[].
26 %
27 % The state vectors, X and X0 consists of continuous states followed
28 % by discrete states.
29 %
30 % Optional parameters, P1,...,Pn can be provided to the S-function and
31 % used during any FLAG operation.
32 %
33 % When SFUNC is called with FLAG = 0, the following information
34 % should be returned:
35 %
36 %     SYS(1) = Number of continuous states.
37 %     SYS(2) = Number of discrete states.
38 %     SYS(3) = Number of outputs.
39 %     SYS(4) = Number of inputs.
40 %             Any of the first four elements in SYS can be specified
41 %             as -1 indicating that they are dynamically sized. The
42 %             actual length for all other flags will be equal to the
43 %             length of the input, U.
44 %     SYS(5) = Reserved for root finding. Must be zero.
45 %     SYS(6) = Direct feedthrough flag (1=yes, 0=no). The s-function
46 %             has direct feedthrough if U is used during the FLAG=3
47 %             call. Setting this to 0 is akin to making a promise that
48 %             U will not be used during FLAG=3. If you break the promise
49 %             then unpredictable results will occur.
50 %     SYS(7) = Number of sample times. This is the number of rows in TS.
51 %
52 %     X0      = Initial state conditions or [] if no states.
53 %
54 %     STR     = State ordering strings which is generally specified as [].
55 %
56 %     TS     = An m-by-2 matrix containing the sample time
57 %             (period, offset) information. Where m = number of sample
58 %             times. The ordering of the sample times must be:
59 %
60 %             TS = [0      0,      : Continuous sample time.
```



```

61 %           0      1,      : Continuous, but fixed in minor step
62 %                               sample time.
63 %           PERIOD OFFSET, : Discrete sample time where
64 %                               PERIOD > 0 & OFFSET < PERIOD.
65 %           -2      0];    : Variable step discrete sample time
66 %                               where FLAG=4 is used to get time of
67 %                               next hit.
68 %
69 %           There can be more than one sample time providing
70 %           they are ordered such that they are monotonically
71 %           increasing. Only the needed sample times should be
72 %           specified in TS. When specifying more than one
73 %           sample time, you must check for sample hits explicitly by
74 %           seeing if
75 %           abs(round((T-OFFSET)/PERIOD) - (T-OFFSET)/PERIOD)
76 %           is within a specified tolerance, generally 1e-8. This
77 %           tolerance is dependent upon your model's sampling times
78 %           and simulation time.
79 %
80 %           You can also specify that the sample time of the S-function
81 %           is inherited from the driving block. For functions which
82 %           change during minor steps, this is done by
83 %           specifying SYS(7) = 1 and TS = [-1 0]. For functions which
84 %           are held during minor steps, this is done by specifying
85 %           SYS(7) = 1 and TS = [-1 1].
86 %
87 %           SIMSTATECOMPLIANCE = Specifies how to handle this block when saving and
88 %           restoring the complete simulation state of the
89 %           model. The allowed values are: 'DefaultSimState',
90 %           'HasNoSimState' or 'DisallowSimState'. If this value
91 %           is not specified, then the block's compliance with
92 %           simState feature is set to 'UnknownSimState'.
93 %
94 %           Copyright 1990-2009 The MathWorks, Inc.
95 %           $Revision: 1.18.2.4 $
96
97 zeig=0; % {0,1} 1...Anzeige von Parameterwerten im Command Window
98
99 switch flag
100
101 %~~~~~
102 % Initialization %
103 %~~~~~
104 case 0
105     [sys,x0,str,ts]=mdlInitializeSizes;
106
107 %~~~~~
108 % Update %
109 %~~~~~
110 case 2
111     sys=mdlUpdate(t,x,u,Ta,zeig);
112
113 %~~~~~
114 % Outputs %
115 %~~~~~
116 case 3
117     sys=mdlOutputs(t,x,u,Ta,zeig);
118
119 %~~~~~
120 % GetTimeOfNextVarHit %

```



```

121  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
122  case 4
123      sys=mdlGetTimeOfNextVarHit(t,x,u,Ta,zeig);
124
125  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
126  % Terminate %
127  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
128  case 9
129      sys=mdlTerminate(t,x,u);
130
131  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
132  % Unhandled flags %
133  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
134  case 1
135      sys = [];
136
137  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
138  % Unexpected flags %
139  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
140  otherwise
141      error(['Unhandled flag = ',num2str(flag)]);
142
143  end
144 % end varhit2ideal
145
146 %-----
147 % mdlInitializeSizes
148 % Return the sizes, initial conditions, and sample times for the S-function.
149 %-----
150 function [sys,x0,str,ts]=mdlInitializeSizes
151
152 % call simsizes for a sizes structure, fill it in and convert it to a
153 % sizes array
154 sizes = simsizes;
155
156 sizes.NumContStates = 0;
157 sizes.NumDiscStates = 1; % x...speichert Pulsperiode
158 sizes.NumOutputs = 1; % SFUNCout
159 sizes.NumInputs = 2; % u(1)...Zähler für Pulsperioden (SFUNCin1)
160                   % u(2)...Einschaltdauer tein (SFUNCin2)
161 sizes.DirFeedthrough = 1;
162 sizes.NumSampleTimes = 1; % at least one sample time is needed
163
164 sys = simsizes(sizes);
165
166 % initialize the initial conditions
167 x0 = -1;
168
169 % str is always an empty matrix
170 str = [];
171
172 % initialize the array of sample times
173 ts = [-2 0]; % Variable step discrete sample time
174 %           where FLAG=4 is used to get time of
175 %           next hit.
176 % end mdlInitializeSizes
177
178 %-----
179 % mdlUpdate
180 % Handle discrete state updates, sample time hits, and major time step

```

```
181 % requirements.
182 %-----
183 function sys=mdlUpdate(t,x,u,Ta,zeig)
184
185 if x~=u(1) % x ungleich SFUNCin1
186     sys=u(1); % SFUNCin1 wird in x gespeichert
187 else
188     sys=x; % x bleibt unverändert
189 end
190
191 if zeig==1 % Ausgabe in Command Window
192 disp('Update');
193 disp([t; x; u; sys]); % sys...update Pulsperiode
194 end
195 % end mdlUpdate
196
197 %-----
198 % mdlOutputs
199 % Return the block outputs.
200 %-----
201 function sys=mdlOutputs(t,x,u,Ta,zeig)
202
203 if x~=u(1) % x ungleich SFUNCin1
204     if u(2)==Ta % tein=Ta
205         sys=1; % SFUNCout=1
206     else
207         sys=0; % SFUNCout=0
208     end
209 else
210     if t<(u(1)*Ta+Ta/2) % momentaner Zeitpunkt kleiner Ta/2
211         sys=1; % SFUNCout=1 zum Zeitpunkt t1
212     else
213         sys=0; % SFUNCout=1 zum Zeitpunkt t2
214     end
215 end
216
217 if zeig==1 % Ausgabe in Command Window
218 disp('Output');
219 disp([t; x; u; sys]); % sys...SFUNCout
220 end
221 % end mdlOutputs
222
223 %-----
224 % mdlGetTimeOfNextVarHit
225 % Return the time of the next hit for this block. Note that the result is
226 % absolute time.
227 %-----
228 function sys=mdlGetTimeOfNextVarHit(t,x,u,Ta,zeig)
229
230 if x~=u(1) % x ungleich SFUNCin1
231     if (0<u(2)) && (u(2)<Ta) % 0<tein<Ta
232         sys=u(1)*Ta+(Ta-u(2))/2; % Berechnung Zeitpunkt t1
233     else
234         sys=(u(1)+1)*Ta; % nächster Aufruf am Anfang der nächste Pulsperiode
235     end
236 else
237     if t<(u(1)*Ta+Ta/2); % momentaner Zeitpunkt kleiner Ta/2
238         sys=u(1)*Ta+(Ta+u(2))/2; % Berechnung Zeitpunkt t2
239     else
240         sys=(u(1)+1)*Ta; % nächster Aufruf am Anfang der nächste Pulsperiode
```

```
241     end
242 end
243
244 if zeig==1 % Ausgabe in Command Window
245 disp('NextHit');
246 disp([t; x; u; sys]); % sys...nächster Zeitpunkt für Aufruf der S-Function
247 end
248 % end mdlGetTimeOfNextVarHit
249
250 %-----
251 % mdlTerminate
252 % Perform any end of simulation tasks.
253 %-----
254 function sys=mdlTerminate(t,x,u)
255
256 sys = [];
257 % end mdlTerminate
```

A.2 MATLAB-Funktion „varhit4tvi.m“

```

1 function [sys,x0,str,ts] = varhit4tvi(t,x,u,flag,Ta,tv)
2
3 % Hilfe zu "M-file S-functions" siehe varhit2ideal.m
4
5 zeig=1; % {0,1} 1...Anzeige von Parameterwerten im Command Window
6
7 switch flag
8
9     %%%%%%%%%%%%%%%%%%%%%%%%%
10    % Initialization %
11    %%%%%%%%%%%%%%%%%%%%%%%%%
12    case 0
13        [sys,x0,str,ts]=mdlInitializeSizes;
14
15    %%%%%%%%%
16    % Update %
17    %%%%%%%%%
18    case 2
19        sys=mdlUpdate(t,x,u,Ta,tv,zeig);
20
21    %%%%%%%%%
22    % Outputs %
23    %%%%%%%%%
24    case 3
25        sys=mdlOutputs(t,x,u,Ta,tv,zeig);
26
27    %%%%%%%%%%%%%%%%%%%%%%%%%
28    % GetTimeOfNextVarHit %
29    %%%%%%%%%%%%%%%%%%%%%%%%%
30    case 4
31        sys=mdlGetTimeOfNextVarHit(t,x,u,Ta,tv,zeig);
32
33    %%%%%%%%%
34    % Terminate %
35    %%%%%%%%%
36    case 9
37        sys=mdlTerminate(t,x,u);
38
39    %%%%%%%%%%%%%%%%%%%%%%%%%
40    % Unhandled flags %
41    %%%%%%%%%%%%%%%%%%%%%%%%%
42    case 1
43        sys = [];
44
45    %%%%%%%%%%%%%%%%%%%%%%%%%
46    % Unexpected flags %
47    %%%%%%%%%%%%%%%%%%%%%%%%%
48    otherwise
49        error(['Unhandled flag = ',num2str(flag)]);
50
51 end
52 % end varhit4tvi
53
54 %-----
55 % mdlInitializeSizes
56 % Return the sizes, initial conditions, and sample times for the S-function.
57 %-----
58 function [sys,x0,str,ts]=mdlInitializeSizes
59
60 % call simsizes for a sizes structure, fill it in and convert it to a

```

```

61 % sizes array
62 sizes = simsizes;
63
64 sizes.NumContStates = 0;
65 sizes.NumDiscStates = 2; % x(1)...speichert Pulsperiode
66                       % x(2)...flag für Verriegelungszeit tv
67 sizes.NumOutputs    = 1; % SFUNCout
68 sizes.NumInputs     = 3; % u(1)...Zähler für Pulsperioden (SFUNCin1)
69                       % u(2)...Einschaltdauer tein (SFUNCin2)
70                       % u(3)...Augenblickswert des Stroms i1(t) (SFUNCin3)
71 sizes.DirFeedthrough = 1;
72 sizes.NumSampleTimes = 1; % at least one sample time is needed
73
74 sys = simsizes(sizes);
75
76 % initialize the initial conditions
77 x0 = [-1 0];
78
79 % str is always an empty matrix
80 str = [];
81
82 % initialize the array of sample times
83 ts = [-2 0]; % Variable step discrete sample time
84 %           where FLAG=4 is used to get time of
85 %           next hit.
86 % end mdlInitializeSizes
87
88 %-----
89 % mdlUpdate
90 % Handle discrete state updates, sample time hits, and major time step
91 % requirements.
92 %-----
93 function sys=mdlUpdate(t,x,u,~,tv,zeig)
94
95 sys=zeros(2,1);
96 if x(1)~=u(1) % x(1) ungleich SFUNCin1
97     sys(1)=u(1); % SFUNCin1 wird in x(1) gespeichert
98     sys(2)=0; % tv nicht aktiv
99 else
100    sys(1)=x(1); % x(1) bleibt unverändert
101    if tv>0 && x(2)==0
102        sys(2)=1; % tv aktiv für t1 und t2
103    elseif tv>0 && x(2)==-1
104        sys(2)=0; % tv nicht aktiv für t1v und t2v
105    end
106 end
107
108 if zeig==1 % Ausgabe in Command Window
109 disp('Update');
110 disp([t; x; u; sys]); % sys(1)...update Pulsperiode
111                       % sys(2)...update flag tv
112 end
113 % end mdlUpdate
114
115 %-----
116 % mdlOutputs
117 % Return the block outputs.
118 %-----
119 function sys=mdlOutputs(t,x,u,Ta,tv,zeig)
120

```

```

121 if x(1)~=u(1) % x(1) ungleich SFUNCin1
122     if u(2)==Ta % tein=Ta
123         sys=1; % SFUNCout=1
124     else
125         sys=0; % SFUNCout=0
126     end
127 else
128     if t<(u(1)*Ta+Ta/2) % momentaner Zeitpunkt kleiner Ta/2
129         if tv>0 && x(2)==0 % Zeitpunkt t1
130             if (u(3)>0) % i1>0
131                 sys=0; % SFUNCout=0
132             elseif (u(3)<0) % i1<0
133                 sys=1; % SFUNCout=1
134             elseif (u(3)==0) % i1==0
135                 sys=0.5; % SFUNCout=0.5
136             end
137         else
138             sys=1; % zum Zeitpunkt t1v SFUNCout=1
139         end
140     else
141         if tv>0 && x(2)==0 % Zeitpunkt t2
142             if (u(3)>0) % i1>0
143                 sys=0; % SFUNCout=0
144             elseif (u(3)<0) % i1<0
145                 sys=1; % SFUNCout=1
146             elseif (u(3)==0) % i1==0
147                 sys=0.5; % SFUNCout=0.5
148             end
149         else
150             sys=0; % zum Zeitpunkt t2v SFUNCout=0
151         end
152     end
153 end
154
155 if zeig==1 % Ausgabe in Command Window
156 disp('Output');
157 disp([t; x; u; sys]); % sys...SFUNCout
158 end
159 % end mdlOutputs
160
161 %=====
162 % mdlGetTimeOfNextVarHit
163 % Return the time of the next hit for this block. Note that the result is
164 % absolute time.
165 %=====
166 function sys=mdlGetTimeOfNextVarHit(t,x,u,Ta,tv,zeig)
167
168 if x(1)~=u(1) % x(1) ungleich SFUNCin1
169     if (0<u(2)) && (u(2)<Ta) % 0<tein<Ta
170         sys=u(1)*Ta+(Ta-u(2))/2; % Berechnung Zeitpunkt t1
171     else
172         sys=(u(1)+1)*Ta; % nächster Aufruf am Anfang der nächste Pulsperiode
173     end;
174 else
175     if t<(u(1)*Ta+Ta/2); % momentaner Zeitpunkt kleiner Ta/2
176         if tv>0 && x(2)==0 % Bedingung für tv
177             sys=t+tv; % Berechnung Zeitpunkt t1v
178         else
179             sys=u(1)*Ta+(Ta+u(2))/2; % Berechnung Zeitpunkt t2
180         end

```

```
181     else
182         if tv>0 && x(2)==0 % Bedingung für tv
183             sys=t+tv; % Berechnung Zeitpunkt t2v
184         else
185             sys=(u(1)+1)*Ta; % nächster Aufruf am Anfang d. nächste Pulsperiode
186         end
187     end
188 end
189
190 if zeig==1 % Ausgabe in Command Window
191 disp('NextHit');
192 disp([t; x; u; sys]); % sys...nächster Zeitpunkt für Aufruf der S-Function
193 end
194 % end mdlGetTimeOfNextVarHit
195
196 %-----
197 % mdlTerminate
198 % Perform any end of simulation tasks.
199 %-----
200 function sys=mdlTerminate(t,x,u)
201
202 sys = [];
203 % end mdlTerminate
```


A.3 MATLAB-Funktion „varhit4tvii.m“

```

1 function [sys,x0,str,ts] = varhit4tvii(t,x,u,flag,Ta,tv)
2
3 % Hilfe zu "M-file S-functions" siehe varhit2ideal.m
4
5 zeig=0; %(0,1) 1...Anzeige von Parameterwerten im Command Window
6
7 switch flag
8
9 %%%%%%%%%%%%%%%%%%%%%%%%%%%
10 % Initialization %
11 %%%%%%%%%%%%%%%%%%%%%%%%%%%
12 case 0
13     [sys,x0,str,ts]=mdlInitializeSizes;
14
15 %%%%%%%%%%%%%%%%%%%%%%%%%%%
16 % Update %
17 %%%%%%%%%%%%%%%%%%%%%%%%%%%
18 case 2
19     sys=mdlUpdate(t,x,u,Ta,tv,zeig);
20
21 %%%%%%%%%%%%%%%%%%%%%%%%%%%
22 % Outputs %
23 %%%%%%%%%%%%%%%%%%%%%%%%%%%
24 case 3
25     sys=mdlOutputs(t,x,u,Ta,tv,zeig);
26
27 %%%%%%%%%%%%%%%%%%%%%%%%%%%
28 % GetTimeOfNextVarHit %
29 %%%%%%%%%%%%%%%%%%%%%%%%%%%
30 case 4
31     sys=mdlGetTimeOfNextVarHit(t,x,u,Ta,tv,zeig);
32
33 %%%%%%%%%%%%%%%%%%%%%%%%%%%
34 % Terminate %
35 %%%%%%%%%%%%%%%%%%%%%%%%%%%
36 case 9
37     sys=mdlTerminate(t,x,u);
38
39 %%%%%%%%%%%%%%%%%%%%%%%%%%%
40 % Unhandled flags %
41 %%%%%%%%%%%%%%%%%%%%%%%%%%%
42 case 1
43     sys = [];
44
45 %%%%%%%%%%%%%%%%%%%%%%%%%%%
46 % Unexpected flags %
47 %%%%%%%%%%%%%%%%%%%%%%%%%%%
48 otherwise
49     error(['Unhandled flag = ',num2str(flag)]);
50
51 end
52 % end varhit4tvii
53
54 %-----
55 % mdlInitializeSizes
56 % Return the sizes, initial conditions, and sample times for the S-function.
57 %-----
58 function [sys,x0,str,ts]=mdlInitializeSizes
59
60 % call simsizes for a sizes structure, fill it in and convert it to a

```



```

61 % sizes array
62 sizes = simsizes;
63
64 sizes.NumContStates = 0;
65 sizes.NumDiscStates = 2; % x(1)...speichert Pulsperiode
66                       % x(2)...flag für Verriegelungszeit tv
67 sizes.NumOutputs    = 2; % SFUNCout1
68                       % SFUNCout2...Steuersignal für tv {0,1}
69 sizes.NumInputs     = 2; % u(1)...Zähler für Pulsperioden (SFUNCin1)
70                       % u(2)...Einschaltdauer tein (SFUNCin2)
71 sizes.DirFeedthrough = 1;
72 sizes.NumSampleTimes = 1; % at least one sample time is needed
73
74 sys = simsizes(sizes);
75
76 % initialize the initial conditions
77 x0 = [-1 0];
78
79 % str is always an empty matrix
80 str = [];
81
82 % initialize the array of sample times
83 ts = [-2 0]; % Variable step discrete sample time
84 %           where FLAG=4 is used to get time of
85 %           next hit.
86 % end mdlInitializeSizes
87
88 %-----
89 % mdlUpdate
90 % Handle discrete state updates, sample time hits, and major time step
91 % requirements.
92 %-----
93 function sys=mdlUpdate(t,x,u,~,tv,zeig)
94
95 sys=zeros(2,1);
96 if x(1)~=u(1) % x(1) ungleich SFUNCin1
97     sys(1)=u(1); % SFUNCin1 wird in x(1) gespeichert
98     sys(2)=0; % tv nicht aktiv
99 else
100     sys(1)=x(1); % x(1) bleibt unverändert
101     if tv>0 && x(2)==0
102         sys(2)=1; % tv aktiv für t1 und t2
103     elseif tv>0 && x(2)==-1
104         sys(2)=0; % tv nicht aktiv für t1v und t2v
105     end
106 end
107
108 if zeig==1 % Ausgabe in Command Window
109 disp('Update');
110 disp([t; x; u; sys]); % sys(1)...update Pulsperiode
111                       % sys(2)...update flag tv
112 end
113 % end mdlUpdate
114
115 %-----
116 % mdlOutputs
117 % Return the block outputs.
118 %-----
119 function sys=mdlOutputs(t,x,u,Ta,tv,zeig)
120

```

```

121 sys=zeros(2,1);
122 if x(1)~=u(1) % x(1) ungleich SFUNCin1
123     if u(2)==Ta % tein-Ta
124         sys(1)=1; sys(2)=0; % SFUNCout1=1, SFUNCout2=0
125     else
126         sys(1)=0; sys(2)=0; % SFUNCout1=0, SFUNCout2=0
127     end
128 else
129     if t<(u(1)*Ta+Ta/2) % momentaner Zeitpunkt kleiner Ta/2
130         if tv>0 && x(2)==0 % Zeitpunkt t1
131             sys(1)=0; sys(2)=1; % SFUNCout1=0, SFUNCout2=1
132         else % Zeitpunkt t1v
133             sys(1)=1; sys(2)=0; % SFUNCout1=1, SFUNCout2=0
134         end
135     else
136         if tv>0 && x(2)==0 % Zeitpunkt t2
137             sys(1)=0; sys(2)=1; % SFUNCout1=0, SFUNCout2=1
138         else % Zeitpunkt t2v
139             sys(1)=0; sys(2)=0; % SFUNCout1=0, SFUNCout2=0
140         end
141     end
142 end
143
144 if zeig==1 % Ausgabe in Command Window
145 disp('Output');
146 disp([t; x; u; sys]); % sys...SFUNCout1, SFUNCout2
147
148 end
149 % end mdlOutputs
150
151 %-----
152 % mdlGetTimeOfNextVarHit
153 % Return the time of the next hit for this block. Note that the result is
154 % absolute time.
155 %-----
156 function sys=mdlGetTimeOfNextVarHit(t,x,u,Ta,tv,zeig)
157
158 if x(1)~=u(1) % x(1) ungleich SFUNCin1
159     if (0<u(2)) && (u(2)<Ta) % 0<tein<Ta
160         sys=u(1)*Ta+(Ta-u(2))/2; % Berechnung Zeitpunkt t1
161     else
162         sys=(u(1)+1)*Ta; % nächster Aufruf am Anfang der nächste Pulsperiode
163     end
164 else
165     if t<(u(1)*Ta+Ta/2); % momentaner Zeitpunkt kleiner Ta/2
166         if tv>0 && x(2)==0 % Bedingung für tv
167             sys=t+tv; % Berechnung Zeitpunkt t1v
168         else
169             sys=u(1)*Ta+(Ta+u(2))/2; % Berechnung Zeitpunkt t2
170         end
171     else
172         if tv>0 && x(2)==0 % Bedingung für tv
173             sys=t+tv; % Berechnung Zeitpunkt t2v
174         else
175             sys=(u(1)+1)*Ta; % nächster Aufruf am Anfang d. nächste Pulsperiode
176         end
177     end
178 end
179
180 if zeig==1 % Ausgabe in Command Window

```

```
181 disp('NextHit');
182 disp([t; x; u; sys]); % sys...nächster Zeitpunkt für Aufruf der S-Function
183 end
184 % end mdlGetTimeOfNextVarHit
185
186 %-----
187 % mdlTerminate
188 % Perform any end of simulation tasks.
189 %-----
190 function sys=mdlTerminate(t,x,u)
191
192 sys = [];
193 % end mdlTerminate
```

A.4 MATLAB-Funktion „varhit4spabf.m“

```

1 function [sys,x0,str,ts] = varhit4spabf(t,x,u,flag,Ta,tv)
2
3 % Hilfe zu "M-file S-functions" siehe varhit2ideal.m
4
5 zeig=0; % {0,1} 1...Anzeige von Parameterwerten im Command Window
6
7 switch flag
8
9     %%%%%%%%%%%%%%%%%%%%%%%%%
10    % Initialization %
11    %%%%%%%%%%%%%%%%%%%%%%%%%
12    case 0
13        [sys,x0,str,ts]=mdlInitializeSizes;
14
15    %%%%%%%%%%%%%%%%%%%%%%%%%
16    % Update %
17    %%%%%%%%%%%%%%%%%%%%%%%%%
18    case 2
19        sys=mdlUpdate(t,x,u,Ta,tv,zeig);
20
21    %%%%%%%%%%%%%%%%%%%%%%%%%
22    % Outputs %
23    %%%%%%%%%%%%%%%%%%%%%%%%%
24    case 3
25        sys=mdlOutputs(t,x,u,Ta,tv,zeig);
26
27    %%%%%%%%%%%%%%%%%%%%%%%%%
28    % GetTimeOfNextVarHit %
29    %%%%%%%%%%%%%%%%%%%%%%%%%
30    case 4
31        sys=mdlGetTimeOfNextVarHit(t,x,u,Ta,tv,zeig);
32
33    %%%%%%%%%%%%%%%%%%%%%%%%%
34    % Terminate %
35    %%%%%%%%%%%%%%%%%%%%%%%%%
36    case 9
37        sys=mdlTerminate(t,x,u);
38
39    %%%%%%%%%%%%%%%%%%%%%%%%%
40    % Unhandled flags %
41    %%%%%%%%%%%%%%%%%%%%%%%%%
42    case 1
43        sys = [];
44
45    %%%%%%%%%%%%%%%%%%%%%%%%%
46    % Unexpected flags %
47    %%%%%%%%%%%%%%%%%%%%%%%%%
48    otherwise
49        error(['Unhandled flag = ',num2str(flag)]);
50
51 end
52 % end varhit4spabf
53
54 %-----
55 % mdlInitializeSizes
56 % Return the sizes, initial conditions, and sample times for the S-function.
57 %-----
58 function [sys,x0,str,ts]=mdlInitializeSizes
59
60 % call simsizes for a sizes structure, fill it in and convert it to a

```

```

61 % sizes array
62 sizes = simsizes;
63
64 sizes.NumContStates = 0;
65 sizes.NumDiscStates = 2; % x(1)...speichert Pulsperiode
66 % x(2)...flag für Verriegelungszeit tv
67 sizes.NumOutputs = 2; % SFUNCout1
68 % SFUNCout2...Steuersignal coin_spabf {1,2,3}
69 sizes.NumInputs = 2; % u(1)...Zähler für Pulsperioden (SFUNCin1)
70 % u(2)...Einschaltdauer tein (SFUNCin2)
71 sizes.DirFeedthrough = 1;
72 sizes.NumSampleTimes = 1; % at least one sample time is needed
73
74 sys = simsizes(sizes);
75
76 % initialize the initial conditions
77 x0 = [-1 0];
78
79 % str is always an empty matrix
80 str = [];
81
82 % initialize the array of sample times
83 ts = [-2 0]; % Variable step discrete sample time
84 % where FLAG=4 is used to get time of
85 % next hit.
86 % end mdlInitializeSizes
87
88 %-----
89 % mdlUpdate
90 % Handle discrete state updates, sample time hits, and major time step
91 % requirements.
92 %-----
93 function sys=mdlUpdate(t,x,u,~,tv,zeig)
94
95 sys=zeros(2,1);
96 if x(1)~=u(1) % x(1) ungleich SFUNCin1
97     sys(1)=u(1); % SFUNCin1 wird in x(1) gespeichert
98     sys(2)=0; % tv nicht aktiv
99 else
100     sys(1)=x(1); % x(1) bleibt unverändert
101     if tv>0 && x(2)==0
102         sys(2)=1; % tv aktiv für t1 und t2
103     elseif tv>0 && x(2)==-1
104         sys(2)=0; % tv nicht aktiv für t1v und t2v
105     end
106 end
107
108 if zeig==1 % Ausgabe in Command Window
109 disp('Update');
110 disp([t; x; u; sys]); % sys(1)...update Pulsperiode
111 % sys(2)...update flag tv
112 end
113 % end mdlUpdate
114
115 %-----
116 % mdlOutputs
117 % Return the block outputs.
118 %-----
119 function sys=mdlOutputs(t,x,u,Ta,tv,zeig)
120

```

```

121 sys=zeros(2,1);
122 if x(1)~=u(1) % x(1) ungleich SFUNCin1
123     if u(2)~=Ta % tein-Ta
124         sys(1)=1; sys(2)=3; % SFUNCout1=1, SFUNCout2=3
125     else
126         sys(1)=0; sys(2)=1; % SFUNCout1=0, SFUNCout2=1
127     end
128 else
129     if t<(u(1)*Ta+Ta/2) % momentaner Zeitpunkt kleiner Ta/2
130         if tv>0 && x(2)==0 % Zeitpunkt t1
131             sys(1)=0; sys(2)=2; % SFUNCout1=0, SFUNCout2=2
132         else % Zeitpunkt t1v
133             sys(1)=1; sys(2)=3; % SFUNCout1=1, SFUNCout2=3
134         end
135     else
136         if tv>0 && x(2)==0 % Zeitpunkt t2
137             sys(1)=0; sys(2)=2; % SFUNCout1=0, SFUNCout2=2
138         else % Zeitpunkt t2v
139             sys(1)=0; sys(2)=1; % SFUNCout1=0, SFUNCout2=1
140         end
141     end
142 end
143
144 if zeig==1 % Ausgabe in Command Window
145 disp('Output');
146 disp([t; x; u; sys]); % sys...SFUNCout1, SFUNCout2
147 end
148 % end mdlOutputs
149
150 %-----
151 % mdlGetTimeOfNextVarHit
152 % Return the time of the next hit for this block. Note that the result is
153 % absolute time.
154 %-----
155 function sys=mdlGetTimeOfNextVarHit(t,x,u,Ta,tv,zeig)
156
157 if x(1)~=u(1) % x(1) ungleich SFUNCin1
158     if (0<u(2)) && (u(2)<Ta) % 0<tein<Ta
159         sys=u(1)*Ta+(Ta-u(2))/2; % Berechnung Zeitpunkt t1
160     else
161         sys=(u(1)+1)*Ta; % nächster Aufruf am Anfang der nächste Pulsperiode
162     end
163 else
164     if t<(u(1)*Ta+Ta/2); % momentaner Zeitpunkt kleiner Ta/2
165         if tv>0 && x(2)==0 % Bedingung für tv
166             sys=t+tv; % Berechnung Zeitpunkt t1v
167         else
168             sys=u(1)*Ta+(Ta+u(2))/2; % Berechnung Zeitpunkt t2
169         end
170     else
171         if tv>0 && x(2)==0 % Bedingung für tv
172             sys=t+tv; % Berechnung Zeitpunkt t2v
173         else
174             sys=(u(1)+1)*Ta; % nächster Aufruf am Anfang d. nächste Pulsperiode
175         end
176     end
177 end
178
179 if zeig==1 % Ausgabe in Command Window
180 disp('NextHit');

```

```
181 disp([t; x; u; sys]); % sys...nächster Zeitpunkt für Aufruf der S-Function
182 end
183 % end mdlGetTimeOfNextVarHit
184
185 %-----
186 % mdlTerminate
187 % Perform any end of simulation tasks.
188 %-----
189 function sys=mdlTerminate(t,x,u)
190
191 sys = [];
192 % end mdlTerminate
```


A.5 MATLAB-Funktion „varhit4pk.m“

```

1 function [sys,x0,str,ts] = varhit4pk(t,x,u,flag,Ta,tv)
2
3 % Hilfe zu "M-file S-functions" siehe varhit2ideal.m
4
5 zeig=0; % {0,1} 1...Anzeige von Parameterwerten im Command Window
6
7 switch flag
8
9 %%%%%%%%%%%%%%%%%%%%%%%%%
10 % Initialization %
11 %%%%%%%%%%%%%%%%%%%%%%%%%
12 case 0
13     [sys,x0,str,ts]=mdlInitializeSizes;
14
15 %%%%%%%%%%%%%%%%%%%%%%%%%
16 % Update %
17 %%%%%%%%%%%%%%%%%%%%%%%%%
18 case 2
19     sys=mdlUpdate(t,x,u,Ta,tv,zeig);
20
21 %%%%%%%%%%%%%%%%%%%%%%%%%
22 % Outputs %
23 %%%%%%%%%%%%%%%%%%%%%%%%%
24 case 3
25     sys=mdlOutputs(t,x,u,Ta,tv,zeig);
26
27 %%%%%%%%%%%%%%%%%%%%%%%%%
28 % GetTimeOfNextVarHit %
29 %%%%%%%%%%%%%%%%%%%%%%%%%
30 case 4
31     sys=mdlGetTimeOfNextVarHit(t,x,u,Ta,tv,zeig);
32
33 %%%%%%%%%%%%%%%%%%%%%%%%%
34 % Terminate %
35 %%%%%%%%%%%%%%%%%%%%%%%%%
36 case 9
37     sys=mdlTerminate(t,x,u);
38
39 %%%%%%%%%%%%%%%%%%%%%%%%%
40 % Unhandled flags %
41 %%%%%%%%%%%%%%%%%%%%%%%%%
42 case 1
43     sys = [];
44
45 %%%%%%%%%%%%%%%%%%%%%%%%%
46 % Unexpected flags %
47 %%%%%%%%%%%%%%%%%%%%%%%%%
48 otherwise
49     error(['Unhandled flag = ',num2str(flag)]);
50
51 end
52 % end varhit4pk
53
54 %=====
55 % mdlInitializeSizes
56 % Return the sizes, initial conditions, and sample times for the S-function.
57 %=====
58 function [sys,x0,str,ts]=mdlInitializeSizes
59
60 % call sizens for a sizes structure, fill it in and convert it to a

```



```

61 % sizes array
62 sizes = simsizes;
63
64 sizes.NumContStates = 0;
65 sizes.NumDiscStates = 2; % x(1)...speichert Pulsperiode
66 % x(2)...flag für Verriegelungszeit tv
67 sizes.NumOutputs = 3; % SFUNCout1
68 % SFUNCout2...Steuersignal coin_spabf {1,2,3}
69 % SFUNCout3...Steuersignal coin_pk {0,1,2}
70 sizes.NumInputs = 3; % u(1)...Zähler für Pulsperioden (SFUNCin1)
71 % u(2)...Einschaltdauer tein (SFUNCin2)
72 % u(3)...Augenblickswert des Stroms i1(t) (SFUNCin3)
73 sizes.DirFeedthrough = 1;
74 sizes.NumSampleTimes = 1; % at least one sample time is needed
75
76 sys = simsizes(sizes);
77
78 % initialize the initial conditions
79 x0 = [-1 0];
80
81 % str is always an empty matrix
82 str = [];
83
84 % initialize the array of sample times
85 ts = [-2 0]; % Variable step discrete sample time
86 % where FLAG=4 is used to get time of
87 % next hit.
88 % end mdlInitializeSizes
89
90 %-----
91 % mdlUpdate
92 % Handle discrete state updates, sample time hits, and major time step
93 % requirements.
94 %-----
95 function sys=mdlUpdate(t,x,u~,tv,zeig)
96
97 sys=zeros(2,1);
98 if x(1)~=u(1) % x(1) ungleich SFUNCin1
99     sys(1)=u(1); % SFUNCin1 wird in x(1) gespeichert
100     sys(2)=0; % tv nicht aktiv
101 else
102     sys(1)=x(1); % x(1) bleibt unverändert
103     if tv>0 && x(2)==0
104         sys(2)=1; % tv aktiv für t1 und t2
105     elseif tv>0 && x(2)==1
106         sys(2)=0; % tv nicht aktiv für t1v und t2v
107     end
108 end
109
110 if zeig==1 % Ausgabe in Command Window
111 disp('Update');
112 disp([t; x; u; sys]); % sys(1)...update Pulsperiode
113 % sys(2)...update flag tv
114 end
115 % end mdlUpdate
116
117 %-----
118 % mdlOutputs
119 % Return the block outputs.
120 %-----

```

```

121 function sys=mdlOutputs(t,x,u,Ta,tv,zeig)
122
123 sys=zeros(3,1);
124 if x(1)~=u(1) % x(1) ungleich SFUNCin1
125     if u(2)==Ta % tein=Ta
126         sys(1)=1; sys(2)=3; sys(3)=0; % SFUNCout1=1, coin_spabf=3, coin_pk=0
127     else
128         sys(1)=0; sys(2)=1; sys(3)=0; % SFUNCout1=1, coin_spabf=1, coin_pk=0
129     end
130 else
131     if t<(u(1)*Ta+Ta/2) % momentaner Zeitpunkt kleiner Ta/2
132         if tv>0 && x(2)==0 % Zeitpunkt t1
133             sys(1)=0; sys(2)=2; % SFUNCout1=0, coin_spabf=2
134             if (u(3)>0) % il>0
135                 sys(3)=0; % coin_pk=0
136             elseif (u(3)<0) % il<0
137                 sys(3)=2; % coin_pk=2
138             elseif (u(3)==0) % il==0
139                 sys(3)=0; % coin_pk=0
140             end
141         else % Zeitpunkt tlv
142             sys(1)=1; sys(2)=3; sys(3)=0; % SFUNCout1=1, coin_spabf=3, coin_pk=0
143         end
144     else
145         if tv>0 && x(2)==0 % Zeitpunkt t2
146             sys(1)=0; sys(2)=2; % SFUNCout1=0, coin_spabf=2
147             if (u(3)>0) % il>0
148                 sys(3)=1; % coin_pk=1
149             elseif (u(3)<0) % il<0
150                 sys(3)=0; % coin_pk=0
151             elseif (u(3)==0) % il==0
152                 sys(3)=0; % coin_pk=0
153             end
154         else % Zeitpunkt t2v
155             sys(1)=0; sys(2)=1; sys(3)=0; % SFUNCout1=0, coin_spabf=1, coin_pk=0
156         end
157     end
158 end
159
160 if zeig==1 % Ausgabe in Command Window
161 disp('Output');
162 disp([t; x; u; sys]); % sys...SFUNCout1, coin_spabf, coin_pk
163 end
164 % end mdlOutputs
165
166 %-----
167 % mdlGetTimeOfNextVarHit
168 % Return the time of the next hit for this block. Note that the result is
169 % absolute time.
170 %-----
171 function sys=mdlGetTimeOfNextVarHit(t,x,u,Ta,tv,zeig)
172
173 if x(1)~=u(1) % x(1) ungleich SFUNCin1
174     if (0<u(2)) && (u(2)<Ta) % 0<tein<Ta
175         sys=u(1)*Ta+(Ta-u(2))/2; % Berechnung Zeitpunkt t1
176     else
177         sys=(u(1)+1)*Ta; % nächster Aufruf am Anfang der nächste Pulsperiode
178     end;
179 else
180     if t<(u(1)*Ta+Ta/2); % momentaner Zeitpunkt kleiner Ta/2

```

```
181     if tv>0 && x(2)==0 % Bedingung für tv
182         sys=t+tv; % Berechnung Zeitpunkt t1v
183     else
184         sys=u(1)*Ta+(Ta+u(2))/2; % Berechnung Zeitpunkt t2
185     end
186 else
187     if tv>0 && x(2)==0 % Bedingung für tv
188         sys=t+tv; % Berechnung Zeitpunkt t2v
189     else
190         sys=(u(1)+1)*Ta; % nächster Aufruf am Anfang d. nächste Pulsperiode
191     end
192 end
193 end
194
195 if zeig==1 % Ausgabe in Command Window
196 disp('NextHit');
197 disp([t; x; u; sys]); % sys...nächster Zeitpunkt für Aufruf der S-Function
198 end
199 % end mdlGetTimeOfNextVarHit
200
201 %=====
202 % mdlTerminate
203 % Perform any end of simulation tasks.
204 %=====
205 function sys=mdlTerminate(t,x,u)
206
207 sys = [];
208 % end mdlTerminate
```

B Datenblatt (IGBT, Freilaufdiode)

SKiiP 342GDL120-4DU



SKiiP[®] 2

7-pack - integrated intelligent Power System

Power section - 3 phase bridge

SKiiP 342GDL120-4DU

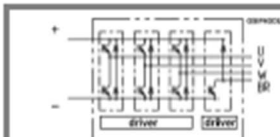
Power section features

- SKiiP technology inside
- CAL diode technology
- Integrated current sensor
- Integrated temperature sensor
- Integrated heat sink
- IEC 60721-3-3 (humidity) class 3K3/IE32 (SKiiP[®] 2 System)
- IEC 60068-1 (climate) 40/125/56
- UL recognized file no. E63532

1) with assembly of suitable MKP capacitor per terminal

Absolute Maximum Ratings		$T_s = 25\text{ }^\circ\text{C}$ unless otherwise specified	
Symbol	Conditions	Values	Units
IGBT			
V_{CES}		1200	V
$V_{CC}^{(1)}$	Operating DC link voltage	900	V
V_{GES}		± 20	V
I_C	$T_s = 25\text{ (70) }^\circ\text{C}$	300 (225)	A
Inverse diode			
$I_F = -I_C$	$T_s = 25\text{ (70) }^\circ\text{C}$	300 (225)	A
I_{FSM}	$T_J = 160\text{ }^\circ\text{C}$; $t_p = 10\text{ ms}$; sin.	2160	A
t_H (Diode)	Diode, $T_J = 160\text{ }^\circ\text{C}$, 10 ms	23	ks
T_J (T_{stg})		-40 (-25) ... +160 (125)	$^\circ\text{C}$
V_{sot}	AC, 1 min. (mainterminals to heat sink)	3000	V

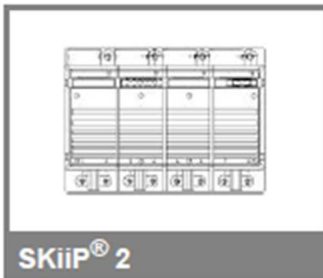
Characteristics		$T_s = 25\text{ }^\circ\text{C}$ unless otherwise specified				
Symbol	Conditions	min.	typ.	max.	Units	
IGBT						
V_{CEsat}	$I_C = 260\text{ A}$, $T_J = 25\text{ (125) }^\circ\text{C}$		2,6 (3,1)	3,1	V	
V_{CEO}	$T_J = 25\text{ (125) }^\circ\text{C}$		1,2 (1,3)	1,6 (1,6)	V	
r_{CE}	$T_J = 25\text{ (125) }^\circ\text{C}$		6,3 (7)	6,3 (8,1)	m Ω	
I_{CES}	$V_{GE} = 0\text{ V}$, $V_{CE} = V_{CES}$, $T_J = 25\text{ (125) }^\circ\text{C}$		(15)	0,4	mA	
$E_{on} + E_{off}$	$I_C = 260\text{ A}$, $V_{CC} = 600\text{ V}$ $T_J = 125\text{ }^\circ\text{C}$, $V_{CC} = 900\text{ V}$			75	mJ	
$R_{CC} + EE'$	terminal chip, $T_J = 125\text{ }^\circ\text{C}$		0,6		m Ω	
L_{CE}	top, bottom		15		nH	
C_{CHC}	per phase, AC-side		1,4		nF	
Inverse diode						
$V_F = V_{EC}$	$I_F = 260\text{ A}$, $T_J = 25\text{ (125) }^\circ\text{C}$		2,1 (2)	2,6	V	
V_{FO}	$T_J = 25\text{ (125) }^\circ\text{C}$		1,3 (1)	1,4 (1,1)	V	
r_F	$T_J = 25\text{ (125) }^\circ\text{C}$		3,3 (4)	4,6 (5,2)	m Ω	
E_{rr}	$I_C = 260\text{ A}$, $V_{CC} = 600\text{ V}$ $T_J = 125\text{ }^\circ\text{C}$, $V_{CC} = 900\text{ V}$			10	mJ	
				12	mJ	
Mechanical data						
M_{dc}	DC terminals, SI Units		6	8	Nm	
M_{ac}	AC terminals, SI Units		13	15	Nm	
w	SKiiP [®] 2 System w/o heat sink		3,5		kg	
w	heat sink		8,5		kg	
Thermal characteristics (P16 heat sink; 275 m³/h); "r" reference to temperature sensor						
$R_{th(j-c)}$	per IGBT			0,09	K/W	
$R_{th(j-d)}$	per diode			0,25	K/W	
$R_{th(s-a)}$	per module			0,036	K/W	
Z_{th}	R_{th} (mKW) (max. values)		tau(s)			
		1	2	3	4	
$Z_{th(j-c)}$		10	69	11	0	1
$Z_{th(j-d)}$		28	193	30	0	1
$Z_{th(s-a)}$		1,7	24	7,6	2,6	494
						165
						20
						0,03



Case S 5

* The specifications of our components may not be considered as an assurance of component characteristics. Components have to be tested for the respective application. Adjustments may be necessary. The use of SEMIKRON products in life support appliances and systems is subject to prior specification and written approval by SEMIKRON. We therefore strongly recommend prior consultation of our personal.

SKiiP 342GDL120-4DU



Absolute Maximum Ratings		$T_a = 25\text{ °C}$ unless otherwise specified	
Symbol	Conditions	Values	Units
V_{S1}	stabilized 15 V power supply	18	V
V_{S2}	unstabilized 24 V power supply	30	V
V_{IH}	input signal voltage (high)	$16 + 0,3$	V
dv/dt	secondary to primary side	75	kV/ μ s
V_{SO10}	input / output (AC, r.m.s., 2s)	3000	Vac
V_{SO12}	output 1 / output 2 (AC, r.m.s., 2s)	1500	Vac
f_{sw}	switching frequency	20	kHz
f_{out}	output frequency for I_{OLC} ; sin.	1	kHz
T_{op} (T_{stg})	operating / storage temperature	- 40 ... + 85	°C

7-pack - integrated intelligent Power System

7-pack

integrated gate driver - 3 phase bridge
SKiiP 342GDL120-4DU

Gate driver features

- CMOS compatible inputs
- Wide range power supply
- Integrated circuitry to sense phase current, heat sink temperature and DC-bus voltage (option)
- Short circuit protection
- Over current protection
- Over voltage protection (option)
- Power supply protected against under voltage
- Interlock of top/bottom switch
- Isolation by transformer
- IEC 60068-1 (climate) 40/85/56

Characteristics		$(T_a = 25\text{ °C})$			
Symbol	Conditions	min.	typ.	max.	Units
V_{S1}	supply voltage stabilized	14,4	15	15,6	V
V_{S2}	supply voltage non stabilized	20	24	30	V
I_{S1}	$V_{S1} = 15\text{ V}$	$410 + 550 \cdot f_{sw} + 3,6 \cdot I_{AC} / A$			mA
I_{S2}	$V_{S2} = 24\text{ V}$	$300 + 390 \cdot f_{sw} + 2,6 \cdot I_{AC} / A$			mA
V_{IT+}	input threshold voltage (High)			12,3	V
V_{IT-}	input threshold voltage (Low)	4,6			V
R_{IN}	input resistance	10			k Ω
$t_{STON/O}$	input-output turn-on propagation time			1,5	μ s
$t_{SLO/O}$	input-output turn-off propagation time			1,4	μ s
$t_{MEM/RESET}$	error memory reset time	9			μ s
t_{TD}	top / bottom switch : interlock time	2,3			μ s
$I_{ANALOGOUT}$	8 V corresponds to max. current of 15 V supply voltage (available when supplied with 24 V)	300			A
$I_{VOUTMAX}$	output current at pin 13/20/22/24/26			50	mA
I_{OLMAX}	output current at pin 13/20/22/24/26			5	mA
V_{OL}	logic low output voltage			0,6	V
V_{OH}	logic high output voltage			30	V
I_{TRIPSC}	over current trip level ($I_{ANALOGOUT} = 10\text{ V}$)	376			A
I_{TRIPLG}	ground fault protection	87			A
T_{tp}	over temperature protection	110		120	°C
U_{DCTRIP}	trip level of U_{DC} -protection ($U_{ANALOGOUT} = 9\text{ V}$); (option)	900			V

For electrical and thermal design support please use SEMISEL.
Access to SEMISEL is via SEMIKRON website <http://www.semikron.com>.

* The specifications of our components may not be considered as an assurance of component characteristics. Components have to be tested for the respective application. Adjustments may be necessary. The use of SEMIKRON products in life support appliances and systems is subject to prior specification and written approval by SEMIKRON. We therefore strongly recommend prior consultation of our personal.

