

Masterarbeit

MA 654

**Konfigurierbare integrierte digitale
Flankenregelung von externen und
internen hochvoltfesten
MOS-Transistoren in automobilen
Applikationen**

zur Erlangung des akademischen Grades

Dipl.-Ing.

vorgelegt dem Institut für Elektronik
an der Technischen Universität Graz
(Leiter: Dipl.-Ing. Dr. Wolfgang Pribyl),
durchgeführt bei Infineon Technologies Austria AG, Villach

Johannes Janschitz, BSc

15. März 2010

Betreuer Infineon Technologies AG:

Dipl.-Ing. Herwig Wappis

Dipl.-Ing.(FH) Jürgen Schäfer

Betreuer TU Graz:

Ass. Prof. Dipl.-Ing. Dr. Peter Söser

Kurzfassung

Konfigurierbare integrierte digitale Flankenregelung von externen und internen hochvoltfesten MOS-Transistoren in automobilen Applikationen

In integrierten Systemen werden Baugruppen nicht nur hinsichtlich funktionaler Eigenschaften implementiert, sondern bei solchen Anwendungen steht auch die gegenseitige Beeinflussung durch Abgabe bzw. Aufnahme von Störsignalen zwischen elektronischen Komponenten im Vordergrund. Ein EMV gerechtes Design spielt eine immer wichtigere Rolle in modernen Technologien, die mit hochvoltfesten Bauelementen entsprechend hohe Spannungspegel schalten, um Energie aber auch Information zu übertragen. Dadurch resultieren Spannungs- bzw. Stromflanken mit hohen Anstiegsgeschwindigkeiten, welche wiederum potentielle Quellen für EMV-Emissionen sind.

Abschwächende Maßnahmen gegen diese Störungen sollen möglichst mit integriert werden, um einerseits Kosten gering zu halten, aber auch um Applikationen in puncto Zuverlässigkeit nicht negativ zu beeinflussen.

Eine Variante dies zu bewerkstelligen zeigt die vorliegende Arbeit, bei der durch Regelung die Anstiegsgeschwindigkeit konstant gehalten und je nach Bedarf vorgegeben wird, um Störaussendungen gering zu halten.

Ein Fallbeispiel soll auch zeigen, wie durch den Einsatz digitaler Regler, Schaltapplikationen vielseitiger anwendbar werden, bzw. wie dadurch der Konfigurationsaufwand auf der Kundenseite möglichst gering gehalten werden kann.

Abstract

Configurable integrated digital slope control of external and internal high voltage MOS-switches in automotive applications

The design of integrated circuits focuses not only on functionality, but also on the emission and absorption of electromagnetic signals between electronic components as an essential factor. An EMC compatible design is playing an increasingly significant role in modern technologies that involve high voltage devices operating at high currents and voltages to transfer energy or information. The resulting switching slopes with transitions at high slew rates are a potential source of interference.

Where possible, measures to attenuate electromagnetic interference should be integrated within module design in order to minimize costs and ensure that the measures do not negatively affect reliability.

One solution to achieve this is described in this thesis, where the slew rate is controlled by regulation loops and can be adjusted as required. The case study presented here shows how the use of a digital control system results in a switch circuit that can be used more flexibly with less configuration effort required by the customer.

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

EIDESSTÄTTLICHE 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 entnommene Stellen als solche kenntlich gemacht habe.

Graz, am

.....
(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

.....
date

.....
(signature)

Danksagungen

Die vorliegende Arbeit entstand während meiner Zeit als Diplomand bei Infineon Technologies Austria. Ich danke allen, die durch ihre Unterstützung zum Gelingen dieser Arbeit beigetragen haben. Besonders bedanken möchte ich mich bei:

- meinen Betreuern Herwig und Jürgen, die mich während der Durchführung der Arbeit nach allen Möglichkeiten unterstützt haben. In zahlreichen Gesprächen konnte ich viel dazu lernen und auch wenn ich mich beim Nachgehen von Ideen zu weit abseits befand, wurde ich von ihnen wieder in die richtige Richtung zurück geholt.
- meinem Betreuer und Mentor seitens des Institutes für Elektronik der Technischen Universität Graz, Dr. Peter Söser für seine Unterstützung und die Korrektur der Arbeit.
- Dirk Hammerschmidt, für viele Anregungen und so manche weitreichende Diskussion.
- Stefan, David, Andi, Linda und den anderen Kollegen der Dienststelle für die tolle Zusammenarbeit und das freundliche Arbeitsklima.
- Steffi, für die Hilfe bei der Korrektur.

Natürlich möchte ich mich an dieser Stelle auch bei meinen Eltern Roswitha und Johann für die großartige Unterstützung während meiner gesamten Ausbildung bedanken.

Graz, 15. März 2010

Johannes Janschitz

Inhaltsverzeichnis

Inhaltsverzeichnis	IX
Abbildungsverzeichnis	XI
Tabellenverzeichnis	XV
Abkürzungsverzeichnis	XVII
Formelzeichen und Symbole	XIX
1 Motivation	1
1.1 Ziele der Arbeit	2
2 EMV in der Automobil-Elektronik	3
2.1 Anforderungen	3
2.2 Kopplungsarten	4
2.3 Voraussetzung für Emissionen/Störaussendungen	6
2.4 Elektromagnetische Störungen durch dU/dt -Effekte	7
2.5 Theoretische Analyse	8
2.6 Signalform mit optimalem Spektrum	12
3 Schaltverhalten von Leistungs-MOS-Transistoren	17
3.1 Kapazitäten des MOS-Transistors	18
3.2 Transiente Analyse des Schaltverhaltens	20
3.3 Aktive Gate Regelung	21
4 Fallbeispiel: Flankenregelung für LIN-Transmitter	23
4.1 Der LIN Bus	23
4.1.1 EMV Verhalten	23
4.2 Ansätze zur Flankenregelung	24
4.3 Beschreibung der Strecke	24
4.4 Konzept zur Flankenregelung	26
4.5 Implementierung für den Normal-Modus	34
4.6 Schaltung des Gate Treibers	35
4.7 Stabilität des Regelkreises	39
4.8 Simulationsergebnisse	40
4.9 Layout	43

4.10	Abschließende Betrachtungen	45
5	Digitale Flankenregelung	47
5.1	Leistungsregelung mit Pulsweiten-Modulation (PWM)	47
5.2	PWM geregelte Systeme im KFZ	48
5.3	Einspritzsysteme im Automobil	48
5.4	Spezifikation der Flanke	49
5.5	Ausgangsstruktur High-Side Schalter	49
5.6	Konzepte für die Regelung	51
5.7	Systemdefinition und Dimensionierung	53
5.8	Strecke des betrachteten Systems	58
5.8.1	Modell mit 3 Zustandsvariablen	61
5.8.2	Berechnung des Stellwertes	65
5.8.3	Modell der Zylinderspule	67
5.9	Regelalgorithmus	67
5.10	Das Stellglied	73
5.11	Dimensionierung der Messkapazität	76
5.12	Auflösung und Wahl der Abtastzeit	78
5.13	Stabilitätsbetrachtungen	82
5.14	Vorschlag für die Implementierung des Reglers	83
5.15	Modell & Simulation der Nachlaufregelung	86
5.15.1	Modell des A/D Umsetzers	86
5.15.2	Dimensionierung für einen Leistungs-MOSFET	87
6	Zusammenfassung und Ausblick	97
6.1	Kapitelübersicht	97
6.2	Ergebnisse und Diskussion	98
6.3	Ausblick	100
	Anhang	101
	Literaturverzeichnis	119

Abbildungsverzeichnis

2.1	Elektromagnetisches Umfeld im Kraftfahrzeug	4
2.2	Galvanische Kopplung	4
2.3	Kapazitive Kopplung im Kabelbaum	5
2.4	Induktive Kopplung im Kabelbaum	5
2.5	Unterteilung der EMV Störaussendungen	7
2.6	Unsymmetrisches Trapezsignal	8
2.7	Vergleich diverser Spektren	9
2.8	Berechnetes Spektrum mit einhüllender Approximation	11
2.9	Optional caption for list of figures	12
2.10	Fourierreihe der Trapezkurve	13
2.11	Optional caption for list of figures	13
2.12	Optional caption for list of figures	14
2.13	Optional caption for list of figures	14
2.14	Überlagerte Spektren aus 2.13(b) und 2.11(b)	15
2.15	Optional caption for list of figures	15
3.1	Lateraler DMOS	17
3.2	Querschnitt eines lateralen n-Kanal DMOS-Transistors	18
3.3	Kapazitäten des MOS Transistors	19
3.4	Zeitlicher Verlauf des Schaltverhaltens	20
3.5	Prinzip für aktive Gate Regelung	21
3.6	Kompromiss zwischen verbesserten EMV-Eigenschaften und Verlustleistung	21
4.1	Schematische Darstellung der Strecke	24
4.2	Kleinsignalersatzschaltbild der Strecke	25
4.3	Darstellung des Regelkreises	27
4.4	Konzept zur Flankenregelung des LIN Bus'	27
4.5	Prinzip der Gate-Regelung	29
4.6	zeitlicher Verlauf der Regelgrößen	30
4.7	Schaltungskonzept für LIN Gate Control	31
4.8	Stromspiegel als P Glied	31
4.9	Darstellung des Regelkreises	33
4.10	Schaltung des LIN Reglers	37
4.11	Schaltung des LIN Schalters	38
4.12	Bode Diagramm der Rückkopplungsschleife	39
4.13	Simulationsergebnisse im Normal-Modus	40

4.14	Zeitlicher Verlauf des LIN-Bus Signales im <i>NORMAL-Slope-Mode</i> für $-40^\circ, 27^\circ, 85^\circ$, und 175°C	41
4.15	Zeitlicher Verlauf des LIN-Bus Signales im <i>LOW-Slope-Mode</i> für $-40^\circ, 27^\circ, 85^\circ$, und 175°C	41
4.16	Zeitlicher Verlauf des LIN-Bus Signales im <i>FAST-Slope-Mode</i> für $-40^\circ, 27^\circ, 85^\circ$, und 175°C	42
4.17	Layout des LIN Gate-Driver Moduls	43
4.18	Top-Layout LIN Modul	44
4.19	LIN Flanke mit verschiedenen Weiten des Schalt-Transistors	45
5.1	<i>High-Side</i> Treiber Konzept	50
5.2	Konzept mit <i>Slew rate Control</i>	51
5.3	Konzept für adaptive Regelung der <i>Slew rate</i>	52
5.4	Konzept für Regelung mit 1 Bit ADC	52
5.5	Systemdefinition des Regelkreises	53
5.6	Darstellung der minimalen Schrittweite zur Korrektur des Fehlers	56
5.7	Schematische Darstellung der Strecke	58
5.8	Kleinsignal-ESB für vereinfachte Spannungsübertragung	58
5.9	Bode-Diagramm für das Übertragungsverhalten von U_g nach U_s	59
5.10	Kleinsignal-ESB für Modell mit Stromeingang	61
5.11	Bode-Diagramm des Modells sowie einer Simulation im Vergleich	63
5.12	Bode-Diagramm des Übertragungsverhaltens der Ströme	64
5.13	Gegenüberstellung der Auflösungen von ADC und DAC in Hinblick auf Grenzyklus-Schwingungen	66
5.14	Basis des Regelfehlers, (a) direkt ,(b) nachlaufend	68
5.15	Ablauf der nachlaufenden Regelung	68
5.16	Mathematische Beschreibung des Standardregelkreises	70
5.17	Approximation der Integration mit der Rechteckregel	72
5.18	Kapazitäten des DMOS Transistors	74
5.19	Typisches Gate-Ladungs-Diagramm für externe MOSFETs	74
5.20	Digital/Analog Strom Wandler	75
5.21	Auflösung der Flanke	79
5.22	Beschränkungen durch den digitalen Regler	80
5.23	AD/DA-Umsetzer Auflösung und Abtastrate für diverse Architektur	81
5.24	Struktur des Standardregelkreises mit Umformung durch den A/D Um- setzer	82
5.25	Struktur nach Umformung der Größen	83
5.26	Struktur für die Untersuchung der offene Schleife	83
5.27	Block Diagramm der Reglereinheit	84
5.28	Konzeptbild für die PI-Regler Implementierung	85
5.29	Modell Architektur des 4 Bit Flash A/D Umsetzers	86
5.30	Bode-Diagramm $P(q)/q$	90
5.31	Bode-Diagramm $L(q)$	91
5.32	Sprungantwort des geregelten Systems	92
5.33	Wurzelortskurve des Vorwärtszweigs	93

5.34	Sprungantworten für diverse Parameter	94
5.35	Vergleich der Fehlerverläufe mit und ohne Grenzyklus-Schwingungen .	95
5.36	Transiente Simulation der Drain-Source-Spannung (blau), V_{GS} (rot) und I_{FB} (grün) im eingeregelter Zustand	95
5.37	Transiente Simulation der Drain-Source-Spannung (blau), V_{GS} (grün) und I_{DS} (rot), $t_r = 360$ ns	96
5.38	Transiente Simulation mit 4 Bit ADC	96
.1	Toplevel Schematic der digitalen Regelung	102

Tabellenverzeichnis

2.1	Einhüllende Approximation für Frequenzspektren von Trapezsignalen	10
4.1	Betriebsmodi für LIN Transmitter	29
5.1	Spezifikation der Flanke	49
5.2	Auflösung bei verschiedenen Bits und Fehlerbereichen	55
5.3	Bekannte Größen für die Dimensionierung	87

Abkürzungsverzeichnis

- ADC** Analog/Digital Converter/Umsetzer, [51](#), [53](#), [54](#), [57](#), [65](#), [66](#), [68](#), [70](#), [79](#), [80](#), [86](#), [94](#), [99](#), [100](#)
- AFS** Active Front Steering, [48](#)
- CAN** Bussystem; Controller Area Network, [1](#), [23](#)
- DAC** Digital/Analog Converter/Umsetzer, [51](#), [54](#), [65](#), [66](#), [68](#), [75](#), [82](#), [99](#)
- DMOS** Leistungstransistor; Double-diffused MOS Transistor, [17](#), [18](#), [34](#), [45](#), [73](#), [75](#), [94](#)
- EFI** Electronic Fuel Injection, [1](#)
- EMC** Elektromagnetische Verträglichkeit; Electromagnetic Compatibility, [3](#)
- EMV** Elektromagnetische Verträglichkeit; Electromagnetic Compatibility EMC, [3](#), [10](#), [23](#), [76](#), [97](#), [98](#)
- EPS** Electric Power Steering, [1](#), [48](#)
- ESB** Ersatzschaltbild, [25](#), [59](#), [98](#)
- ESD** Elektrostatische Entladungen; Electrostatic Discharge, [36](#), [43](#), [97](#)
- FKL** Frequenzkennlinien, [28](#), [89](#)
- FOX** Feldoxid, [18](#)
- LIN** Bussystem; Local Interconnect Network, [1](#), [23](#), [34–36](#), [40](#), [43](#), [45](#), [47](#), [98](#)
- LSB** Least Significant Bit, [56](#), [57](#), [64](#), [88](#)
- LZI** Lineares zeitinvariantes System; Linear Time Invariant System, [45](#), [63](#), [68](#)

MOSFET Metall-Oxid Feldeffekttransistor; Metall Oxide Semiconductor Transistor, [17–19](#), [47](#), [74](#), [97](#), [98](#)

PCB Printed Circuit Board, [1](#)

PI Proportional-Integral Regler, [69](#), [85](#), [98](#), [99](#)

PID Proportional-Integral-Derivativer Regler, [79](#)

PWM Pulsweiten-Modulation, [16](#), [48](#), [50](#)

SPT Smart Power Technology, [98](#)

STI Shallow Trench Isolation, [18](#)

Formelzeichen und Symbole

Symbol ¹	Erklärung	Einheit
B	Bandbreite	Hz
C_{iss}	Eingangskapazität; Input capacitance	F
C_{rss}	Rückwirkungskapazität; Reverse transfer capacitance.	F
C_{oss}	Ausgangskapazität; Output capacitance	F
d	Puls-Perioden-Verhältnis	
δ_M	Messbereich	%
ΔV_{ADC}	Stufenspannung des A/D Wandlers	V
f_g	Grenzfrequenz	Hz
g_{ds}	Ausgangsleitwert (Kleinsignal)	$\frac{1}{\Omega}$
g_m	Transkonduktanz (Übertragungsleitwert)	$\frac{1}{\Omega}$
g_{mb}	Transkonduktanz (Substrat)	$\frac{1}{\Omega}$
GND	Bezugspotential, Masse	V
K_P	Proportionalfaktor	
K_L	Induktionskoeffizient	$\frac{1}{m}$
M_{12}	Gegeninduktivität	H
N_{ADC}	Auflösung des A/D Wandlers	Bit
N_{DAC}	Auflösung des D/A Wandlers	Bit
$Q_G(th)$	Ladung um V_{TH} zu erreichen	C
Q_{SW}	Schaltladung	C
Q_G	Gate-Ladung gesamt	C
r_{ds}	Kleinsignalausgangswiderstand	Ω
R_{on}	Widerstand eines eingeschalteten MOST	Ω
SR	Slew Rate	$\frac{V}{s}$
T	Temperatur	$K, (^{\circ}C)$
T_0	Referenztemperatur ($300K$)	K

¹Die verwendeten Konstanten und Symbole wurden an die als sinnvoll befundene Notation von Univ.-Prof. Dr.-Ing. Kurt Hoffmann [Hof06] angepasst

Symbol ¹	Erklärung	Einheit
T_R	t_r/T , relative Anstiegszeit	
T_F	t_f/T , relative Abfallzeit	
T_A	Abtastzeit; Diskretisierungszeit	s
T_N	Nachstellzeit	s
t_d	Verzögerungszeit	s
t_f	Abfallzeit einer Flanke	s
t_r	Anstiegszeit einer Flanke	s
t_s	Einschwingzeit (settling-time)	s
Ω_C	Transitfrequenz	Hz
U_{DC}	Spannungsabfall entlang der Drain-Extension	V
U_{DD}	positive Versorgungsspannung	V
U_{DS}	Drain-Source-Spannung	V
$U_{DS,sat}$	Drain-Source-Sättigungsspannung	V
μ_0	Permeabilität des leeren Raumes	$\frac{Vs}{Am}$
$V_R^2(f)$	spektrale Rauschleistungsdichte	$\frac{V^2}{Hz}$
$V_{R(rms)}$	äquivalente Rauschspannung	V
V_{TH}	Schwellenspannung des Transistors	V

Physikalische Konstanten¹

Konstante	Bedeutung	Zahlenwert
k	Boltzmann-Konstante	$1,38 \cdot 10^{-23} \frac{Ws}{K}$
q	Elementarladung	$1,602 \cdot 10^{-19} As$
$\phi_T = \frac{kT}{q}$	Temperaturspannung	26 mV bei 300 K
μ_0	Permeabilität des leeren Raumes	$4 \cdot \pi \cdot 10^{-7}$
ϵ_0	Dielektrizitätskonstante des Vakuums	$8,854 \cdot 10^{-14} \frac{F}{cm}$
ϵ_{ox}	relative Dielektrizitätskonstante von Siliziumdioxid	3,9 $\frac{Vs}{Am}$

1 Motivation

Aufgrund der hohen Standards bezüglich Zuverlässigkeit und Effizienz werden im automobilen Bereich immer mehr Vorgänge elektronisch gesteuert und überwacht. In solchen integrierten Hochvoltapplikationen werden Leistungstransistoren als Schalter zur Abtrennung bzw. Zuführung von Energie eingesetzt. Diese Transistoren besitzen einen komplexen Aufbau, da sie wesentlich höheren Spannungen standhalten müssen, als es in Standard CMOS Schaltungen üblich ist.

In Applikationen wie beispielsweise Electric Power Steering ([EPS](#)) oder Electronic Fuel Injection ([EFI](#)), aber auch in Bussystemen wie dem Controller Area Network ([CAN](#)) oder dem Local Interconnect Network ([LIN](#)) werden Lasten mit Schalttopologien geschaltet. Aufgrund dieser Schaltvorgänge entstehen Spannungsflanken mit hoher Anstiegsgeschwindigkeit (dU/dt), die potentielle Quellen für EMV Störungen darstellen.

Da im Auto diese hochintegrierten Applikationen sehr dicht aneinander verbaut werden, ist es so von enormer Wichtigkeit, störende Einflüsse so gering wie möglich zu halten und sofern möglich durch abschwächende Maßnahmen zu dämpfen.

Eine effektive Methode ist es, diese Schaltflanken auf ein konstantes dU/dt zu regeln [[Kag07](#)],[[IK04](#)].

In Anwendungen wie elektronisch gesteuerten Einspritzsystemen [EFI](#) werden Magnetventile zur Einspritzung der Kraftstoffmenge in die Brennkammer eingesetzt. Die Größe dieser Ventile variiert mit der Anzahl der Zylinder und so muss auch der Leistungstransistor, der die Stromzufuhr der Magnetspule regelt, angepasst werden. Diese Applikationen werden auf den größten Transistor hin dimensioniert und der Schaltvorgang wird mit passiven Filtern gedämpft. Diese Filter werden auf Printed Circuit Board ([PCB](#)) Ebene eingebaut, was hohen finanziellen Aufwand bedeutet.

Auch für Applikationen, bei denen die Leistungstransistoren mit integriert werden, besteht die Nachfrage nach einer modularen Lösung, um eine Flankenregelung für unterschiedliche Schaltergrößen zu realisieren.

1.1 Ziele der Arbeit

In der vorliegenden Arbeit sollen Ansätze zur Flankenregelung mit analogen und digitalen Methoden untersucht werden, vor allem auch wie und mit welchen Mitteln eine Lösung auf digitaler Ebene realisiert werden kann, um auf sich ändernde Beschaltung hinsichtlich der Transistorparameter mit möglichst wenig Aufwand reagieren zu können. So soll die Notwendigkeit der oben beschriebenen Filter entfallen, um Kosten auf Systemebene zu sparen.

2 EMV in der Automobil-Elektronik

2.1 Anforderungen

Elektromagnetische Verträglichkeit (**EMV**) oder der äquivalente englische Begriff Electromagnetic Compatibility (**EMC**) gewinnt immer mehr an Wichtigkeit in der Automobilindustrie. Ein Automobil stellt ein sogenanntes elektrisches Inselsystem dar, welches grob eingeteilt aus der energiezuführenden Komponente (Generator), der energiespeichernden Komponente (Batterie) sowie aus der verteilenden Komponente, dem Kabelbaum besteht. Aus dieser zugegeben eher rudimentären Beschreibung lässt sich bereits erahnen, dass genau diese *Verteilung von Energie/ Information* enormes Potential in Bezug auf Störanfälligkeit bzw. die Erzeugung von Störungen aufweist. Immerhin besteht der innere Aufbau bzw. die Steuerung eines modernen Autos aus zahlreichen elektronischen Komponenten, welche nicht nur jeweils passend versorgt werden müssen, sondern auch miteinander kommunizieren, um bei Bedarf auf eine vorbestimmte Weise in Abläufe eingreifen zu können. Wie in [Krü04] beschrieben, gibt es zahlreiche Störquellen wie:

- Magnetventile
- Schalter
- digitale Datenübertragung
- Zündanlagen (Einspritzsysteme)
- Sende- und Empfangseinrichtungen
- getaktete Regelkreise
- Motoren / Generatoren

Die angeführten Störquellen können *galvanisch, kapazitiv, induktiv* sowie *elektromagnetisch* in eine Störsenke einkoppeln und in dieser unerwünschte Effekte hervorrufen, welche die Funktion negativ beeinflussen. Über diese Übertragungswege werden Störungen also zu den Störsenken gebracht.



Abb. 2.1: Elektromagnetisches Umfeld im Kraftfahrzeug

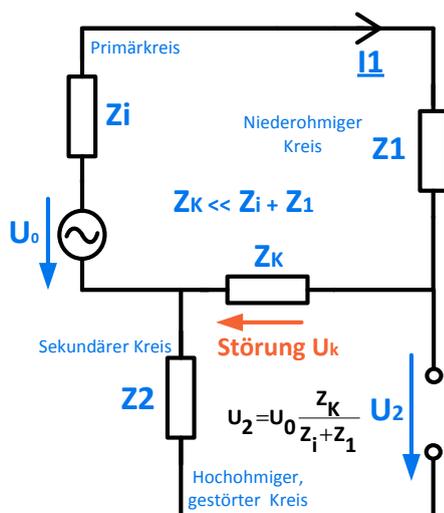
Mögliche Störsenken sind:

- Kabel
- Sensoren
- elektronische Steuergeräte
- Lampen
- Infotainment

Bezogen auf den oben erwähnten Energie-/Informationstransport im Auto sollen an dieser Stelle diverse Kopplungsarten betrachtet werden, die in einem Kabelbaum auftreten.

2.2 Kopplungsarten

Galvanische Kopplung



Die galvanische Kopplung ist eine rein leitungsgebundene Kopplungsart, wobei die Störung aufgrund eines Spannungsabfalls an gemeinsamen Impedanzen auftritt. Diese äußert sich typischerweise bei Systemen mit gemeinsamen Masse-, Versorgungs- oder Busleitungen, wo der Strom eines Schaltungsteils eine Spannungsdifferenz über der Masseimpedanz verursacht, welche dann wiederum in einem Sekundärkreis Auswirkungen hat. Um Störungen zu vermeiden, muss darauf geachtet werden, dass die übergekoppelte Spannung kleiner als die auftretende Signalspannung bleibt

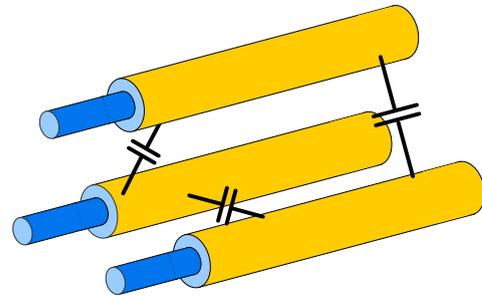
Abb. 2.2: Galvanische Kopplung

[Gon05].

Kapazitive

Kopplung

Da der Kabelbaum eines Autos aus eng aneinander liegenden, parallel geführten Leitungen besteht, die über eine Länge von mehreren Metern verlaufen können, entstehen zwischen diesen Leitungen Kapazitäten, über die Störungen eingekoppelt werden (Abbildung 2.3). Je geringer der Abstand und je größer die Fläche zwischen zwei Kabeln ist, desto größer wird die Kapazität und auch der Einfluss der Kopplung.

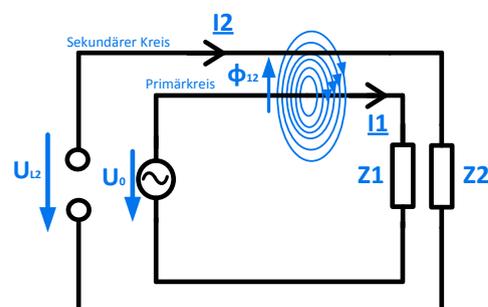


Die Ursache für die kapazitive Kopplung liegt in der Influenzwirkung des elektrischen Feldes. Voraussetzung für eine Störbeeinflussung sind Quellen mit hohen Spannungsänderungen. Störungen dieser Art treten in hochohmigen Lasten auf, wobei sich die Wirkung dieser Kopplungsart mit steigender Frequenz erhöht [Gon05].

Abb. 2.3: Kapazitive Kopplung im Kabelbaum

Induktive Kopplung

Ähnlich wie Kapazitäten ergeben sich auch Induktivitäten in einem Kabelbaum und Störungen werden induktiv eingekoppelt (ausgesendet). Die Ursache hierzu liegt im magnetischen Feld zwischen zwei Leitern und ist als Induktion bekannt.



Störungen werden durch hohe zeitlich veränderliche Ströme, also entweder Wechselströme oder Ströme mit hohen Anstiegsgeschwindigkeiten, verursacht. Wie in Abbildung 2.4 veranschaulicht, tritt im sekundären Kreis eine Spannung U_{L2} auf, die aufgrund des induzierten Stromes I_2 , der durch die Eigenim-

Abb. 2.4: Induktive Kopplung im Kabelbaum

pedanz Z_2 begrenzt wird, an der Last abfällt.

$$u_{L2} = -M_{12} \frac{di_1}{dt} = \frac{d\phi_{12}}{dt} \quad (2.1)$$

Wobei hier M_{12} die Gegeninduktivität und ϕ_{12} den magnetischen Fluss bezeichnen. Für rein sinusförmige Betrachtungen gilt

$$U_{L2} = -j\omega M_{12} I_1. \quad (2.2)$$

Vernachlässigt man die Rückwirkung des sekundären Kreises auf den ersten, ergibt sich

$$I_2 = \frac{j\omega M_{12} I_1}{j\omega L_2} = \frac{M_{12}}{L_2} I_1. \quad (2.3)$$

Somit wird laut Gleichung (2.3) der induzierte Sekundärstrom aus dem Verhältnis von Gegen- zu Eigeninduktivität bestimmt [Gon05].

Elektromagnetische Kopplung

Von elektromagnetischer Kopplung spricht man laut [Gon05] erst, wenn die betrachteten geometrischen Abmessungen nicht mehr klein gegenüber der Wellenlänge der höchsten zu berücksichtigenden Frequenz sind. Ab einer Größe von $\frac{\lambda}{10}$ wird die Laufzeit schon so stark beeinflusst, dass die im Frequenzbereich betrachteten Phaseneinflüsse bereits wesentlich stören.

2.3 Voraussetzung für Emissionen/Störaussendungen

An dieser Stelle stellt sich die Frage: Warum sollten Schaltflanken in Bezug auf zeitliche Änderungen nicht unendlich steil sein? Ausgehend von dem Sachverhalt, dass man bedingt durch physikalische Beschränkungen, wie beispielsweise der Leitungsimpedanz oder der Lastimpedanz, keine unendlich steilen Flanken erhalten wird, geht man einen Schritt weiter und betrachtet was passieren würde, wenn man einen Schaltvorgang in definierter Zeit durchführt. Das auftretende Problem resultiert direkt aus den Maxwell'schen Gleichungen und kann als elektromagnetische Abstrahlung kategorisiert werden. Schnelle Spannungsflanken bringen auch schnelle Änderungen des Stromes mit

sich, um das Potential auszugleichen. Somit erhält man elektrische Ladung. Der Grund für elektromagnetische Emission ist eine zeitliche Änderung der Geschwindigkeit mit der besagte Ladung bewegt wird. Die Voraussetzung für Abstrahlung ist also eine von Null verschiedene Beschleunigung der Ladungsträger [Sau99].

Um die Beschleunigung der Ladungsträger so gering wie möglich zu halten, muss also auch die Spannung- bzw. Stromänderung begrenzt werden! Schnelle Schaltflanken bzw. hohe Taktraten generieren andernfalls elektromagnetische Störungen in benachbarten Elementen wie beispielsweise dem Bordnetz.

2.4 Elektromagnetische Störungen durch dU/dt -Effekte

Elektromagnetische Störungen werden zu Analysezwecken in verschiedene Klassen eingeteilt. Eine erste Einteilung wird in *gestrahlte* und *leitungsgebundene* Störaussendung gemacht. Unter gestrahlter Störaussendung versteht man elektromagnetische Störsignale, die sich durch den Raum ausbreiten, wohingegen unter leitungsgebundener Störaussendung die selben Störungen bezeichnet werden, welche sich aber nun durch einen elektrischen Leiter fortbewegen. Dies kann beispielsweise eine Versorgungsleitung, eine Masseleitung aber auch eine übliche Verbindung zwischen Baugruppen sein [Krü04]. Diese kabelgebundenen Störaussendungen können nun weiter in asymmetrische (*common mode*) und symmetrische (*differential mode*) Aussendungen unterteilt werden.

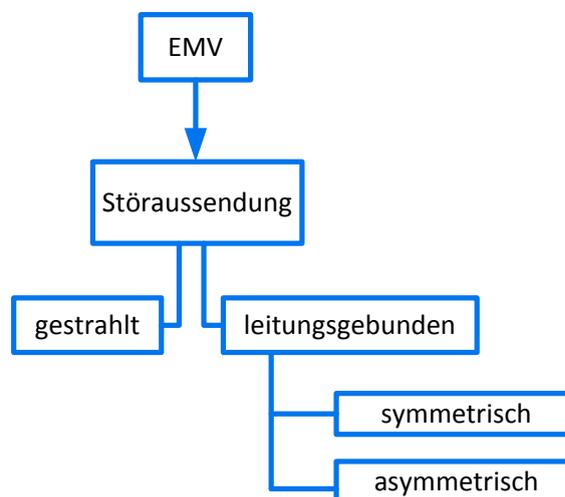


Abb. 2.5: Unterteilung der EMV Störaussendungen

Symmetrische Störungen sind meist hochfrequente Stromanteile, welche dem eigentlichen Nutzsignal überlagert sind, und durch hohe dI/dt Schaltflanken entstehen. Asymmetrische Störungen sind im Allgemeinen auf Masse bezogen und werden primär durch hohe dU/dt Flanken im Zusammenspiel mit parasitären Kapazitäten hervorgerufen [Kag07].

2.5 Theoretische Analyse

In Schalterapplikationen werden überwiegend digitale Pegel als Steuersignale verwendet (0 und 1). Typischerweise entstehen durch aufeinander folgende, sich laufend ändernde Pegel rechteckförmige Signale. Eigentlich sind es keine wirklichen Rechtecksignale, sondern die Signalform ähnelt einem Trapez mit einer steigenden und mit einer fallenden Flanke. Die Übergangszeiten, um vom LOW auf den HIGH Zustand zu kommen, sind also von endlicher Länge (Abbildung 2.6). Das Amplitudenspektrum eines unsymmetrischen

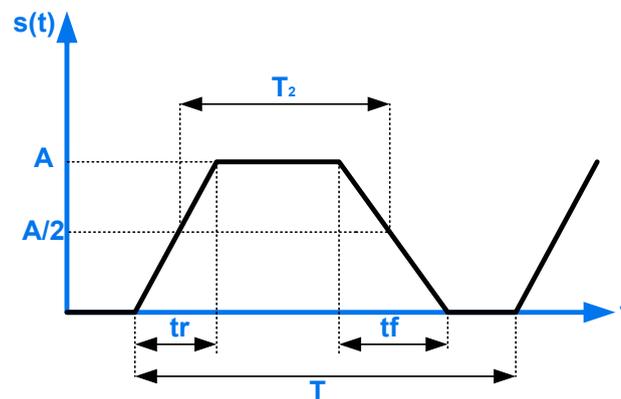


Abb. 2.6: Unsymmetrisches Trapezsignal

trischen Rechtecksignals mit der Amplitude A und dem Puls-Perioden-Verhältnis d^1 (d ist das Verhältnis von T_2 zu T) ist gegeben durch [Nag97]:

$$S(k) = \frac{A}{2\pi k} |3\sin(\pi kd) - \sin(\pi k(2-d))|, \quad (2.4)$$

wobei k die Harmonische an der Stelle bezeichnet. Für das Spektrum eines trapezförmigen Signals ergibt sich laut [Nag97]:

$$S(k) = \frac{A}{\pi k} |\text{sinc}(\pi k T_R) e^{j\pi kd} - \text{sinc}(\pi k T_F) e^{-j\pi kd}| \quad (2.5)$$

¹engl. Duty Cycle

Die Funktion sinc ist folgend definiert (2.6):

$$\text{sinc} : \mathbb{R} \rightarrow \mathbb{R}, x \mapsto \text{sinc}(x) = \begin{cases} \frac{\sin x}{x} & \text{für alle } x \neq 0 \\ 1 & \text{für } x = 0 \end{cases} \quad (2.6)$$

$T_R = tr/T$, relative Anstiegszeit,
 $T_F = tf/T$, relative Abfallzeit,
 $d = T2/T$, relative Pulsweite (duty cycle).

In Abbildung 2.7 sind drei verschiedene Spektren von Rechteck- bzw. Trapezsignalen dargestellt. Das Rechtecksignal besitzt die steilsten Flanken (größtes dU/dt) und weist auch die größte Bandbreite auf. Die Amplitude fällt mit -20 dB/Dekade ab. Hier besteht auch das größte Potential für hochfrequente Störungen. Das Trapezsignal mit den schnellen Flanken ($tr = tf = 10$ ns) fällt ebenfalls mit -20 dB/Dekade, zeigt jedoch einen Knick bei ungefähr $1,5$ kHz und fällt dann weiter mit -40 dB/Dekade. Die Bandbreite ist hier bereits geringer. Schließlich betrachtet man das Trapezsignal mit den langsamen Flanken ($tr = tf = 600$ ns) und kann erkennen, dass sich besagter Knick weiter nach links verschoben hat, wodurch sich die Bandbreite massiv reduziert. Die theoretischen Ansätze werden in [CH92] ausführlich beschrieben.

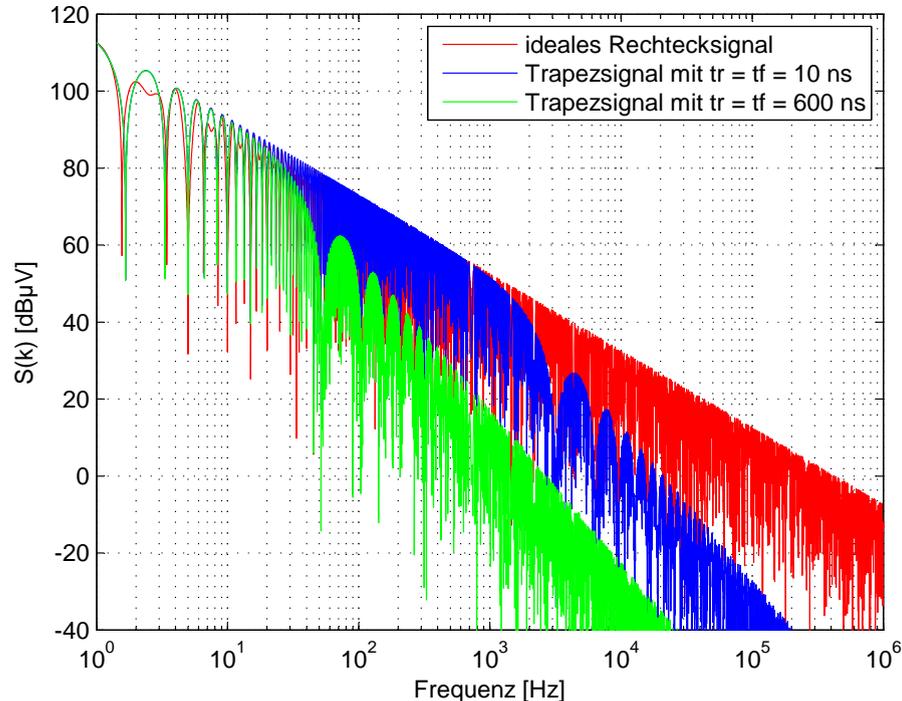


Abb. 2.7: Vergleich diverser Spektren

In [Nag97] wird beschrieben, wie es durch Approximation möglich ist eine einhüllende

Funktion zu generieren, was sich bei der Definition von der oberen Grenze der Amplitude im Spektrum als sehr nützlich erweisen kann.

Diese einhüllende Funktion kann in drei Bereiche für jeweils niedrige, mittlere und hohe Frequenzen geteilt und getrennt approximiert werden:

$$kc1 = \frac{2}{\sqrt{(\text{sinc}(\pi T_R) + \text{sinc}(\pi T_F))^2 \sin(\pi d)^2 + (\text{sinc}(\pi T_R) - \text{sinc}(\pi T_F))^2 \cos(\pi d)^2}} \quad (2.7)$$

$$S(1) = \frac{A}{\pi} \sqrt{(\text{sinc}(\pi T_R) + \text{sinc}(\pi T_F))^2 \sin(\pi d)^2 + (\text{sinc}(\pi T_R) - \text{sinc}(\pi T_F))^2 \cos(\pi d)^2} \quad (2.8)$$

Bereich	Einhüllende	dB/Dek	Eckfrequenzen
Fund Freq.	S(1)	0	kc1
Niedrige	$S(k)_{NF} = \frac{2A}{\pi k}$	-20	$kc2 = \frac{1}{\pi \alpha}, \alpha = \begin{cases} T_R & \text{if } T_R > T_F \\ T_F & \text{if } T_R < T_F \end{cases}$
Mittlere	$S(k)_{MF} = \frac{A}{\pi k} \left(1 + \frac{1}{\pi k \alpha}\right)$	-20 bis -40	$kc3 = \frac{1}{\pi \beta}, \beta = \begin{cases} T_F & \text{if } T_R > T_F \\ T_R & \text{if } T_R < T_F \end{cases}$
Hohe	$S(k)_{HF} = \frac{A}{\pi^2 k^2} \left(\frac{1}{T_R} + \frac{1}{T_F}\right)$	-40	

Tab. 2.1: Einhüllende Approximation für Frequenzspektren von Trapezsignalen

Abbildung (2.8) veranschaulicht das analytisch berechnete Spektrum einer 10 kHz Trapezkurve mit unterschiedlicher Anstiegs- und Abfallzeit. Wie zu sehen ist, approximiert die Einhüllende die Frequenzkurve sehr gut und bildet eine obere Schranke, was zum Abschätzen der EMV-Abstrahlung in Hinblick auf die Richtlinien sehr brauchbar ist.

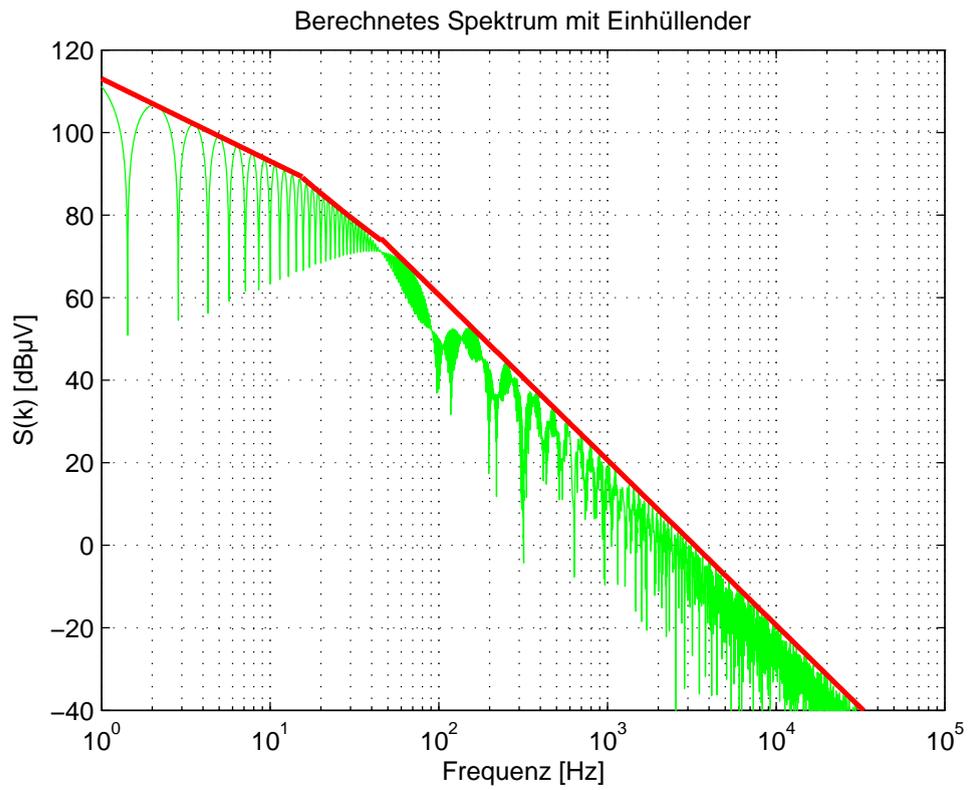
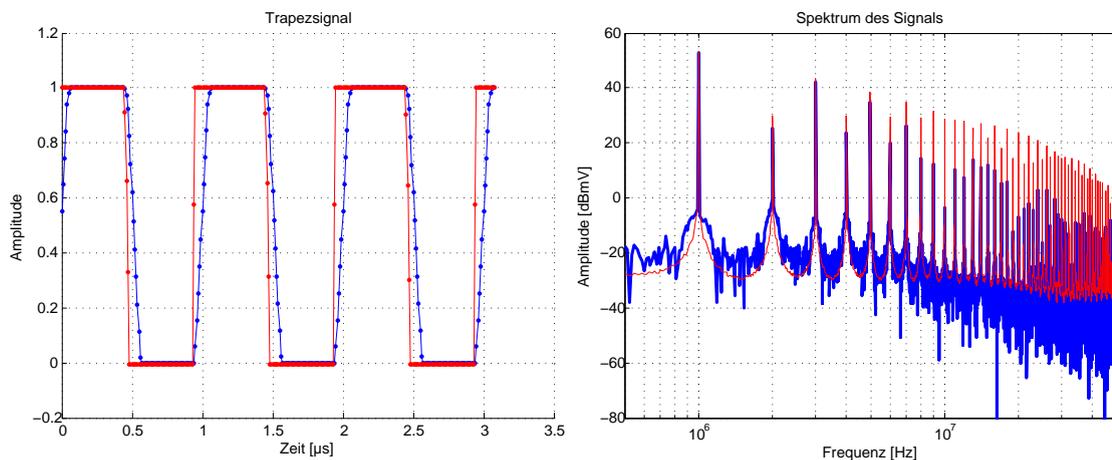


Abb. 2.8: Berechnetes Spektrum mit einhüllender Approximation

2.6 Signalform mit optimalem Spektrum

Aus den obigen Betrachtungen geht hervor, dass besagte Schaltflanken möglichst flach sein sollten, um Frequenzen im harmonischen Bereich leistungsmäßig nach Möglichkeit klein zu halten. Ausgegangen wurde dabei vom Amplitudenspektrum, wobei von der Rechteckkurve in Gleichung (2.4) zum Trapezkurven-Signal Gleichung (2.5) übergegangen wurde.

Betrachtet man hierzu die Fourierreihe für das Signal im Zeitbereich Gleichung (2.9), welche nach unendlicher Summation der Terme zum idealen Trapezsignal konvergiert, kann man darauf schließen, genau das nicht zu tun und das Trapezsignal nur durch eine endliche Summe zu approximieren, um die Signalleistung im Spektrum möglichst klein zu halten. Abbildung 2.9 zeigt eine Gegenüberstellung eines Signals mit gereg-



(a) Trapezsignale rot: geschaltet mit Invertern
blau: geregelt

(b) Spektren

Abb. 2.9: Gegenüberstellung von hart geschaltetem Signal(rot) und geregelter Flanke (blau) (a), (b)

ter Flanke und eines Signals mit steiler geschalteter Flanke. Vor allem im höheren Frequenzbereich macht sich der Unterschied visuell bemerkbar.

$$f(t) = \frac{4A}{\pi\delta} \left(\frac{1}{1^2} \sin\delta \sin t + \frac{1}{3^2} \sin 3\delta \sin 3t - \frac{1}{5^2} \sin 5\delta \sin 5t + \dots \right) \quad (2.9)$$

Gleichung (2.9) lässt sich auch in folgender Form angeben.

$$f(t) = \frac{4A}{\pi\delta} \sum_{n=1}^{\infty} \frac{1}{[2n-1]^2} \sin([2n-1]\delta) \cdot \sin([2n-1]t) \quad (2.10)$$

Beim Signalverlauf sind vor allem Unstetigkeitsstellen von Bedeutung (scharfe Ecken),

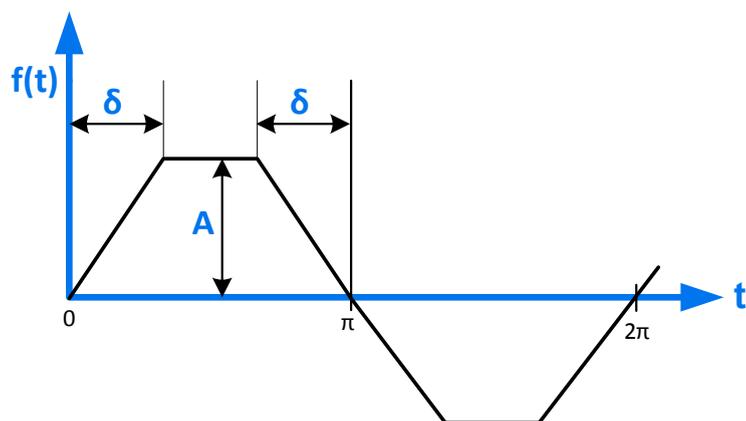
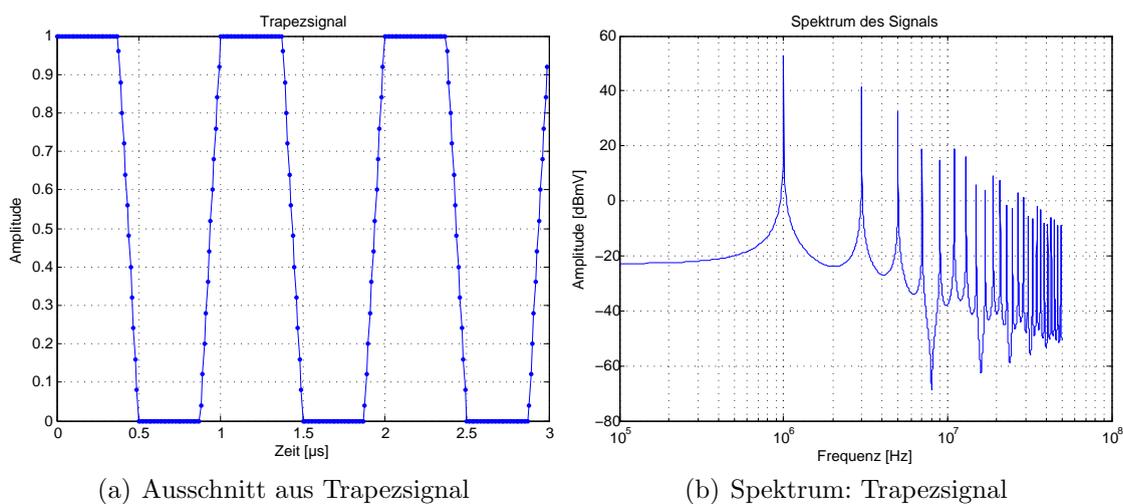


Abb. 2.10: Fourierreihe der Trapezkurve, [Bar99]

welche für eine beliebig genaue Approximation eine entsprechend hohe Anzahl der Summanden fordern. Um dies zu unterbinden, sollte die Signalform einen weichen Verlauf aufweisen.



(a) Ausschnitt aus Trapezsignal

(b) Spektrum: Trapezsignal

Abb. 2.11: (a) Zeitlicher Verlauf eines Trapezsignals, (b) dazugehöriges Spektrum

Aus Abbildung 2.11(b) erkennt man den Signalverlauf eines Trapezsignals. Das Spektrum enthält viel mehr Frequenzen im oberen Bereich als beispielsweise in Abbildung 2.12(b), wo neben der Grundschwingung nur drei ungerade Harmonische in der Summe von Gleichung (2.10) verwendet wurden.

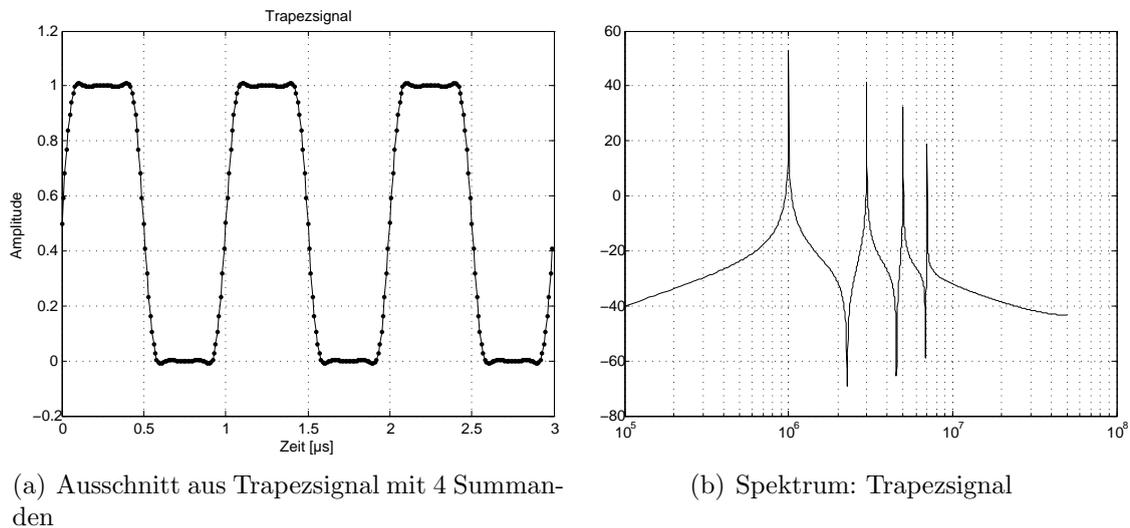


Abb. 2.12: (a) Zeitlicher Verlauf eines Trapezsignals mit Spektrum mit $n=4$ Summanden, (b) dazugehöriges Spektrum

Abbildung 2.13(b) zeigt einen Signalausschnitt, der durch Angabe einiger Referenzpunkte und anschließender Spline-Interpolation generiert wurde, um den Signalverlauf möglichst weich zu gestalten.

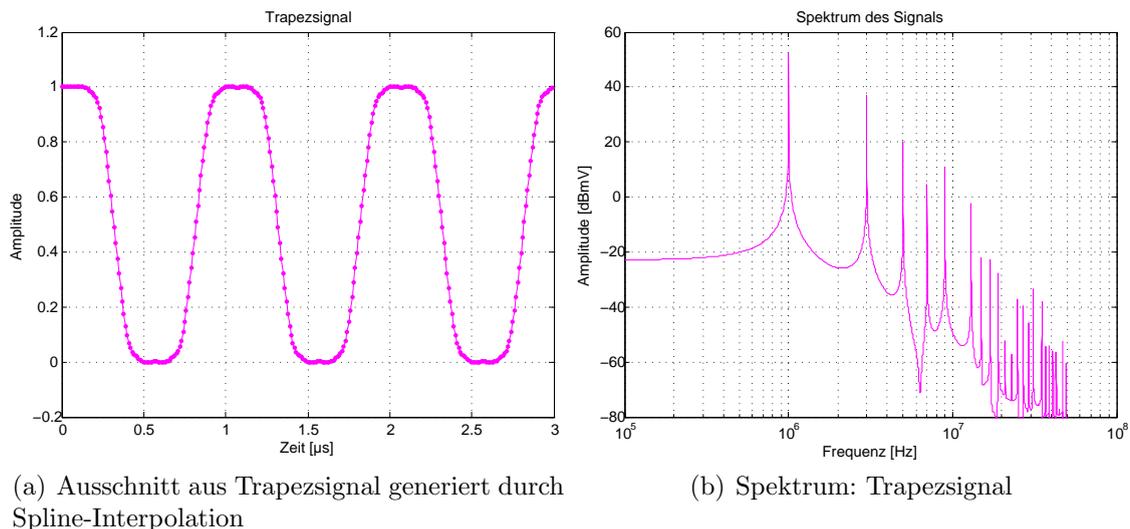


Abb. 2.13: (a) Zeitlicher Verlauf eines Trapezsignals aus einer Spline-Interpolation, (b) korrespondierendes Spektrum

Abbildung 2.14 zeigt die beiden Spektren überlagert, wobei erkennbar ist, dass das Signal konstruiert mit der Interpolation im Vergleich wesentlich geringere Anteile im oberen Frequenzbereich enthält. Um ein optimales Spektrum zu erreichen, sollte der Signalverlauf also frei von harten unstetigen Übergängen sein, besonders wenn die Flanke sehr schnell (Nanosekunden) verläuft. Für eine Fourierreihe gilt in diesem Zusammenhang: Ein unstetiger Bereich der Funktion, oder in einer ihrer Ableitungen fordert eine unendliche Summe an Termen für die Approximation [Kre06].

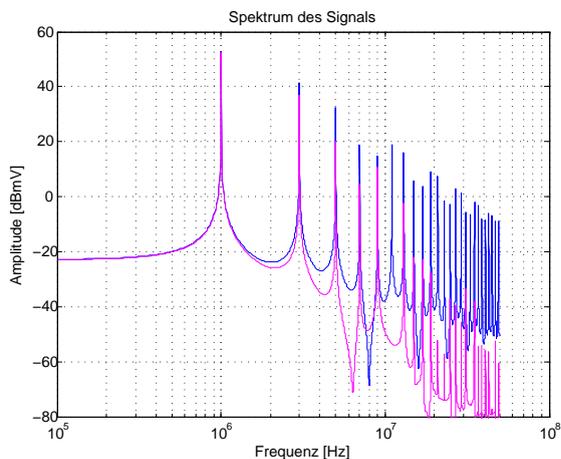
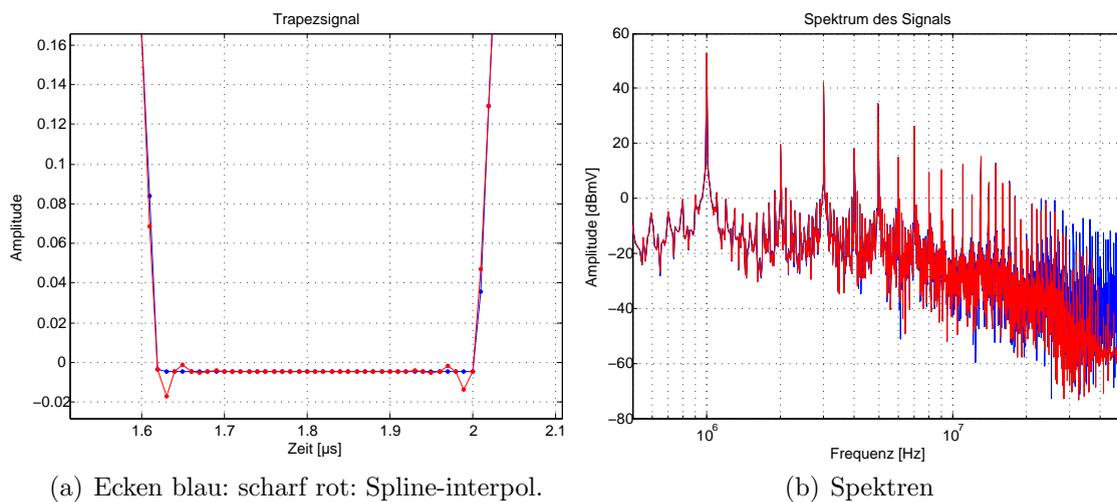


Abb. 2.14: Überlagerte Spektren aus 2.13(b) und 2.11(b)



(a) Ecken blau: scharf rot: Spline-interpol.

(b) Spektren

Abb. 2.15: Signalecken eines Trapezsignals (a), (b) überlagerte Spektren

Abbildung 2.15 zeigt zwei verschiedene Trapezsignale, wobei das blaue aus einer Simulation extrahiert wurde. Das rote Signal wurde aus dem blauen generiert: Zuerst wurde eine Unterabtastung um den Faktor zwei durchgeführt, danach wurden mit einer Spline-Interpolation die vorher verworfenen Punkte neu generiert. Der Unterschied ist vor allem bei den Signalecken sichtbar, welche nun wesentlich weniger hochfrequente Anteile enthalten als zuvor.

Aus den vorangegangenen Analysen kann man folgende Punkte schlussfolgern:

- Die Flanke sollte so langsam wie möglich verlaufen, um hochfrequente Anteile zu vermeiden.
- Die Signalform sollte frei von un stetigen Übergängen ("Ecken") sein, da ansonsten ebenfalls die hochfrequenten Anteile zunehmen.
- Die Signalform sollte einen möglichst linearen Anstieg mit "runden" Übergängen besitzen.

Abschließend sei hier noch anzumerken, dass die oben dargestellten Spektre nur für Signale mit symmetrischem Puls-Pausen-Verhältnis gültig sind. Betrachtet man Schalt-signale mit ungleichem Puls-Pausen-Verhältnis (bsp. Pulsweiten-Modulation ([PWM](#))), kommen auch gerade Spektrallinien hinzu!

tor kommt. Die maximal mögliche Spannung entlang dieser Driftzone hängt von deren Länge und Dotierung ab [Sch05].

Abbildung 3.2 zeigt einen detaillierteren Querschnitt eines lateralen DMOS-Transistors. Über der *Drain Extension* ersetzt Shallow Trench Isolation (STI) das früher verwendete Feldoxid (FOX), um die *Drain Extension* vom Gate zu isolieren [Wap07].

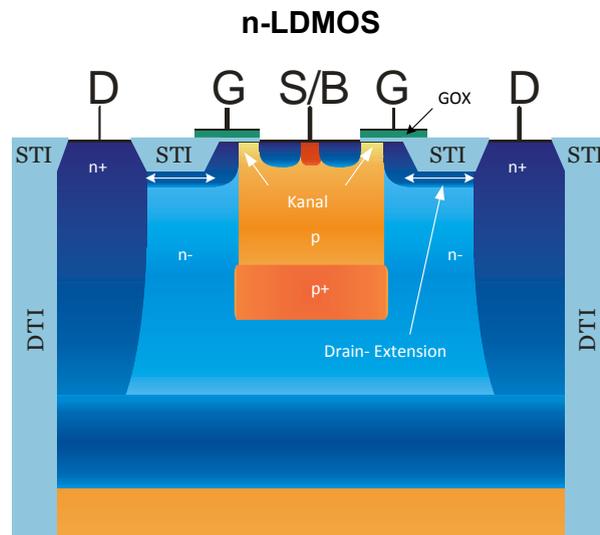


Abb. 3.2: Querschnitt eines lateralen n-Kanal DMOS-Transistors

3.1 Kapazitäten des MOS-Transistors

Um das Schaltverhalten von Leistungs-MOSFETs zu diskutieren, betrachtet man den zeitlichen Verlauf der wichtigsten Größen, die den Schaltvorgang beeinflussen. Ab einer Gate-Spannung größer der Schwellenspannung¹ (vgl. [Lut06]) beginnt ein Drain-Strom zu fließen.

Über dem Kanal wird durch die isolierende Oxidschicht, welche als Dielektrikum dient, und dem Gate-Kontakt eine der Gatefläche (WL) proportionale Kapazität realisiert.

$$C_{OX} = \frac{\epsilon_0 \epsilon_r}{t_{ox}} \cdot W \cdot L = C'_{OX} \cdot W \cdot L \quad (3.1)$$

Die gesamte Kapazität ergibt sich größtenteils aus der Gate-Source (C_{gs}) sowie der Gate-Drain (C_{gd}) Kapazität.

Abbildung 3.3 zeigt die parasitären Kapazitäten. Die Gate-Source Kapazität (C_{gs}) ist sehr abhängig vom Gateoxid sowie der Source Metallisierung. Die Gate-Drain Kapa-

¹engl. Threshold-Spannung

zität besteht größtenteils aus der Überlappungskapazität des Gate-Oxids mit der darunter liegenden Verarmungszone. Die Kapazität von Drain nach Source wird durch die dazwischen liegende Sperrschicht verursacht.

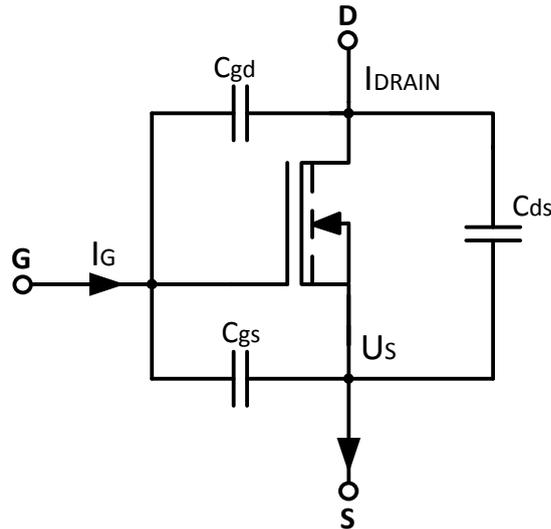


Abb. 3.3: Kapazitäten des MOS Transistors

In Datenblättern werden diese Kapazitäten in folgender Form beschrieben:

$$C_{iss} = C_{gs} + C_{gd} \quad (C_{ds} \text{ kurzgeschlossen}) \quad (3.2)$$

$$C_{rss} = C_{gd} \quad (3.3)$$

$$C_{oss} = C_{ds} + \frac{C_{gs} \cdot C_{gd}}{C_{gs} + C_{gd}} \quad (C_{gs} \text{ kurzgeschlossen}) \approx C_{ds} + C_{gd} \quad (3.4)$$

Die drei Gleichungen beschreiben die äquivalente Eingangskapazität C_{iss} , die Rückwirkungskapazität C_{rss} sowie die äquivalente Ausgangskapazität C_{oss} .

Die bereits erwähnten Kapazitäten C_{gs} und C_{gd} bilden größtenteils die Eingangskapazität des MOSFETs. Während der Schaltphase zeigt sich eine Abhängigkeit der Kapazitätswerte von der Drain-Source-Spannung. C_{gs} wird ausgehend vom ursprünglichen Wert im ausgeschaltetem Zustand während des Einschaltens anwachsen. Und C_{gd} wird im Verlaufe kleiner [Sti94].

3.2 Transiente Analyse des Schaltverhaltens

Für die folgenden Betrachtungen soll Abbildung 3.4 als visuelle Unterstützung dienen. Wird über einen Gate-Widerstand die äquivalente Gate-Kapazität mit zugeführtem Strom aufgeladen, beginnt die Gate-Spannung in der ersten Phase (1) bis zur Threshold-Spannung anzusteigen. Während dieser Zeit wirkt C_{gd} nur sehr schwach und kann gegenüber C_{gs} vernachlässigt werden. Ab dieser Spannung (2) beginnt ein Drain Strom

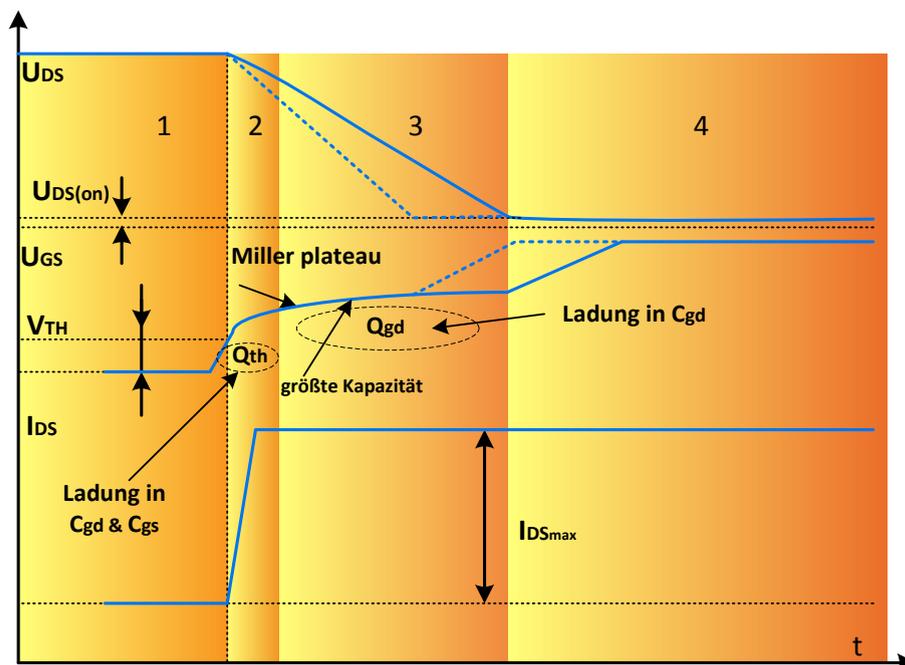


Abb. 3.4: Zeitlicher Verlauf des Schaltverhaltens

zu fließen, welcher bei weiterem Anwachsen der Gate-Source Spannung, in Abhängigkeit der Last, seinen Maximalwert erreicht. Währenddessen beginnt die Drain-Source-Spannung zu fallen.

In Phase (3) ist ein dramatisches Ansteigen der Gate-Drain Kapazität zu beobachten. Bedingt durch die Spannungsverstärkung wirkt die Gate-Drain Kapazität um ein Vielfaches größer am Eingang. Dieses Verhalten während der Umschaltphase wird Millerplateau genannt [Gal93]. Aufgrund des Millereffektes erscheint nun die Gate-Drain Kapazität am Eingang um die Verstärkung vergrößert, wodurch sich die Spannung am Gate im weiteren Verlauf nicht stark ändert. Sobald die Drain-Source-Spannung ihr Minimum erreicht (4), steigt die Gate-Source-Spannung weiter an, ohne jedoch I_{ds} oder U_{ds} weiter zu beeinflussen.

Generell wird der Übergang von U_{ds} während der Phasen (2) und (3) ablaufen. Durch

geeignete Maßnahmen (bsp. Rückkopplung) kann das Millerplateau verlängert werden, um die Anstiegsgeschwindigkeit zu verringern bzw. vorzugeben [Mit95].

3.3 Aktive Gate Regelung

Eine Möglichkeit, um das Schaltverhalten beeinflussen zu können, erhält man durch den Einsatz von Rückkopplungs-Netzwerken, die in Verbindung mit dem Gate-Widerstand zu einem Regelkreis verschaltet werden (Abbildung 3.5). Natürlich wird in den meis-

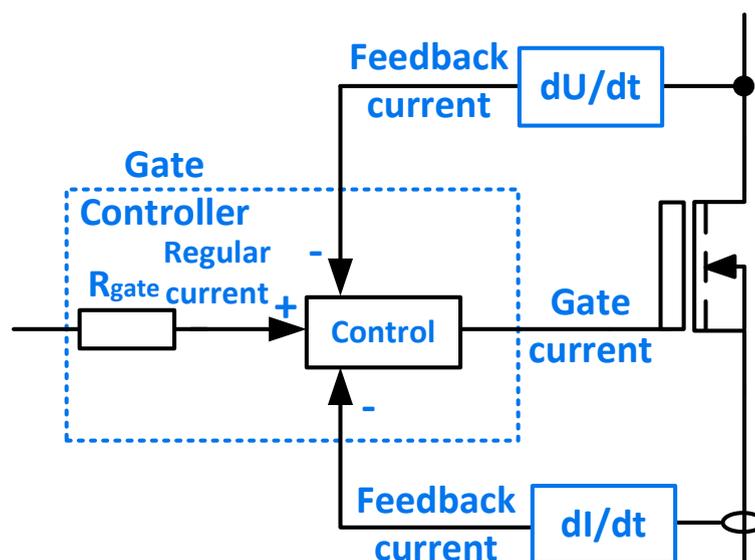


Abb. 3.5: Prinzip für aktive Gate Regelung

ten Fällen nur eine der beiden Größen für die Rückkopplung herangezogen, um einen Feedback-Strom zu generieren. Dieser Strom wird im Gate-Controller geeignet mit dem Gate-Regelstrom kombiniert, um das dU/dt bzw. dI/dt zu beeinflussen.

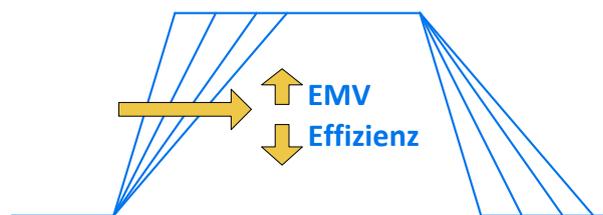


Abb. 3.6: Kompromiss zwischen verbesserten EMV-Eigenschaften und Verlustleistung

Dahingehende Maßnahmen zur Verbesserung der EMV-Eigenschaften stehen immer im Kompromiss mit der Effizienz im Hinblick auf Verlustleistung.

4 Fallbeispiel: Flankenregelung für LIN-Transmitter

In diesem Kapitel soll anhand der Applikation eines LIN Transmitter-Moduls die Implementierung einer Flankenregelung veranschaulicht werden.

4.1 Der LIN Bus

Der LIN Bus (Local Interconnect Network) ist eine kostengünstige Bus Topologie, die vorwiegend für eher unkritische Sensor-Aktuator-Anwendungen im automotiven Bereich Anfang der 1990er Jahre entwickelt wurde. Vor allem für Tür-, Sitz- oder Schiebedachapplikationen bietet der als Ein-Draht ausgeführte LIN Bus enorme Kostensparnisse im Vergleich zum CAN Bus. Die maximale Bitrate liegt bei 20 kbit/s, wobei 2,4 kbit/s, 9,6 kbit/s oder 19,2 kbit/s laut Spezifikation empfohlen werden.

Vom Konzept her ist der LIN Bus ein *Single Master System* mit mehreren *Slaves*. Im automobilen Bereich wird der LIN Bus als *Sub-Bus* für CAN-Netze eingesetzt, wobei ein sogenannter Master-Knoten als Schnittstelle zum übergeordneten CAN-Netz fungiert [ZS06].

4.1.1 EMV Verhalten

Das EMV Verhalten des LIN Bus' hängt sehr stark von der Form der Flanken ab. Die entscheidenden Parameter hierfür sind die *Slew rate* sowie $\frac{dI}{dt}$ und $\frac{dU}{dt}$ des Übertragungssignals. Es gilt prinzipiell die Flankenform so zu gestalten, dass die *Slew rate* im Hinblick auf Abstrahlung möglichst klein wird (kleines $\frac{dU}{dt}$), aber dennoch eine Periodendauer erreicht werden kann, welche die spezifizierte maximale Datenrate von 20 kbit/s erlaubt [LIN06].

4.2 Ansätze zur Flankenregelung

An dieser Stelle sollen Ansätze aufgezeigt werden, wie eine geregelte Schaltflanke mit integrierten Schaltungskonzepten realisiert werden kann. Um einem System ein bestimmtes Verhalten aufprägen zu können, muss man als ersten Schritt Informationen über das Verhalten eben dieses Systems in Form einer Beschreibung (Modell) extrahieren. Dieses Modell wird natürlich der tatsächlichen Strecke in vollem Verhalten nie genau entsprechen. Prinzipiell sollte so ein Modell möglichst einfach, aber nicht zu ungenau sein! Es sollte das Verhalten der tatsächlichen Strecke **im betrachteten Bereich** so gut wiedergeben, dass ein Regler entworfen werden kann, welcher der Strecke das gewünschte Verhalten aufzwingt.

4.3 Beschreibung der Strecke

Das Modell der Strecke besteht prinzipiell aus einem Schalttransistor, einem Lastwiderstand sowie der parasitären Induktivität der Leitung und den parasitären Kapazitäten nach VDD und GND (Abbildung 4.1).

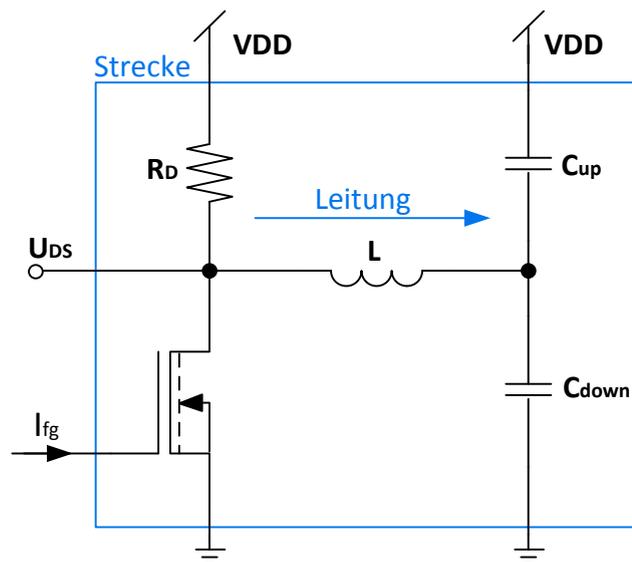


Abb. 4.1: Schematische Darstellung der Strecke

In Abbildung 4.2 ist das Kleinsignalersatzschaltbild der Strecke dargestellt. Für das Modell werden nur die relevanten Kapazitäten von Gate nach Source (C_{GS}) sowie von

Gate nach Drain (C_{GD}) berücksichtigt, da diese auch das Schaltverhalten mit beeinflussen. Weiters soll auch die Ordnung der später errechneten Übertragungsfunktion niedrig gehalten werden.

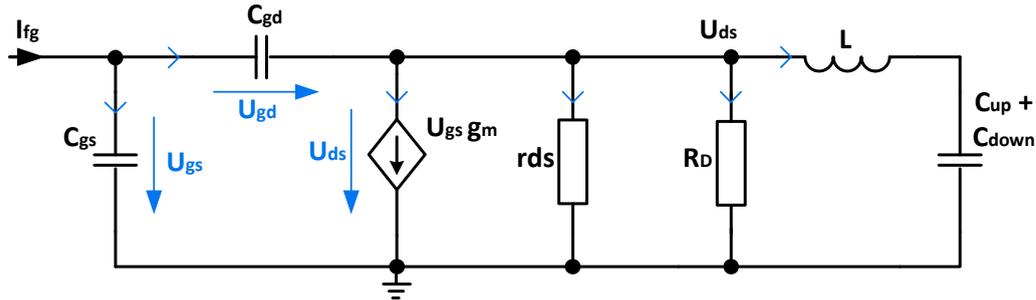


Abb. 4.2: Kleinsignalersatzschaltbild der Strecke

Ziel dieser Regelung ist es, die Drain-Source Spannung des Schalttransistors beim Umschalten so zu beeinflussen, dass dU_{DS}/dt konstant bleibt. Um das zu bewerkstelligen, muss der aktuelle Zustand der zu regelnden Größe (U_{DS}) erfasst werden, um über eine Rückkopplung einen Regelkreis aufbauen zu können.

Zunächst muss aber noch aus dem Kleinsignalersatzschaltbild (ESB) (Abbildung 4.2) die Übertragungsfunktion der Strecke bestimmt werden. Bemerkenswert an dieser Stelle ist auch, dass dieses Modell (ESB) das Verhalten des Transistors während der Umschaltphase beschreibt und nicht jenes während den beiden stabilen Schaltzuständen! Als Eingangsgröße nimmt man den Strom I_{fg} , was zu diesem Zeitpunkt vielleicht etwas seltsam erscheint, jedoch später erläutert wird. Da, wie schon gesagt, die Drain-Source-Spannung U_{DS} betrachtet wird, bildet diese die Ausgangsgröße der Strecke.

Nach Ansetzen der Knotengleichungen

$$-I_{fg} + U_{gs}sC_{gs} + (U_{gs} - U_{ds})sC_{gd} = 0 \quad (4.1)$$

$$-(U_{gs} - U_{ds})sC_{gd} + (U_{gs})g_m + \frac{U_{ds}}{\left(\frac{r_{ds}R_d}{r_{ds}+R_d}\right)} + \frac{U_{ds}}{\left(sL + \frac{1}{sC_{lin}}\right)} = 0 \quad (4.2)$$

und nach Elimination von U_g sowie einigen Umformungen erhält man für die Übertragungsfunktion folgenden Ausdruck:

$$G(s) = \frac{U_{ds}}{I_{fg}} = \frac{(s^3 L C_{gd} C_{lin} - s^2 L g_m C_{lin} + s C_{gd} - g_m) r_{ds} R_d}{s (s^3 C_{gd} \gamma + s^2 \delta + s \beta + C_{gd} g_m r_{ds} R_d + C_{gd} r_{ds} + C_{gd} R_d + C_{gs} r_{ds} + C_{gs} R_d)} \quad (4.3)$$

Wobei untere Terme wegen der besseren Lesbarkeit eingeführt wurden.

$$\beta = \left(C_{gd} \frac{\gamma}{LC_{lin}} + C_{gd} \frac{\gamma}{LC_{gs}} + \frac{\gamma}{L} \right) \quad (4.4)$$

$$\gamma = LC_{lin}r_{ds}C_{gs}R_d \quad (4.5)$$

$$\delta = \left(\frac{C_{gd}}{C_{gs}} g_m \gamma + C_{gd} \frac{\gamma}{R_d C_{gs}} + C_{gd} \frac{\gamma}{r_{ds} C_{gs}} + \frac{\gamma}{R_d} + \frac{\gamma}{r_{ds}} \right) \quad (4.6)$$

In (4.3) fällt, ohne genauere Analyse der Pole bzw. Nullstellen, sofort der Pol bei Null auf. Dieser liefert einen integrierenden Anteil, also ein I-Glied, das grundsätzlich strukturinstabil ist! Die Strecke enthält noch jeweils drei Null- und Polstellen und hat daher die Ordnung 4 und ist vom Typ IT_n , was einem I-Glied mit Verzögerungen entspricht.

Nun stellt sich die Frage nach einer geeigneten Methode zur Erfassung der Drain-Source-Spannung. Da diese Spannung mit bis zu 45 V verhältnismäßig hoch im Verhältnis zur üblichen Versorgungsspannung von 1,5V bzw. 5 V ist, schränkt dies die Messung etwas ein. Eine Möglichkeit wäre ein einfacher resistiver Teiler, wobei die zu messende Spannung geeignet skaliert wird. Diese Möglichkeit birgt allerdings durch die üblichen Probleme, wie statischen Stromverbrauch oder Fläche, eher mehr Nachteile. Vor allem aber müsste eine entsprechend hohe Auflösung auf der Niedervolt-Seite eingesetzt werden, da durch die Skalierung der Messbereich sehr eingeschränkt wird.

Um von der hohen Schaltspannung entkoppelt zu sein, erfasst man die Drain-Source-Spannung kapazitiv und kann, wie weiter unten gezeigt wird, die Messgröße somit direkt in einen Feedback-Strom I_{FB} umwandeln.

Durch diese Vorgehensweise wird auch deutlich, dass durch die Hinzugabe des Messkondensators das integrierende Verhalten der Strecke aufgehoben wird und man erhält ein System, welches als Eingangs- und Ausgangsgröße Ströme besitzt. Außerdem beseitigt man durch diesen Trick auch die Strukturinstabilität der Strecke und hat nun nur noch ein PT_n -Glied zu regeln.

4.4 Konzept zur Flankenregelung

Das Konzept zu dieser Regelung wird in Abbildung 4.3 veranschaulicht. Neben der bereits diskutierten Strecke kommt als Regler noch ein Block, der als *Gate Control* bezeichnet wird, hinzu, welcher einen Differenzstrom mit einem proportional Glied geeignet skaliert und schließlich der Strecke zuführt. Der Faktor χ dient zur Skalierung des Referenzstroms.

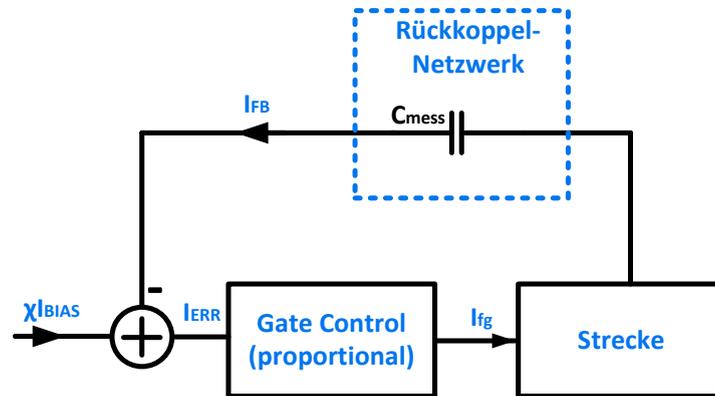


Abb. 4.3: Darstellung des Regelkreises

Abbildung 4.4 zeigt bereits ein erweitertes Bild des Systems mit drei Eingängen und einem Ausgang. Das Signal V_{TX} bezeichnet das digitale Eingangssignal, das zum Schalten benötigt wird. Das Signal $slope_mode$ stellt die gewünschte Flankensteilheit in Form von vier wählbaren Alternativen bereit. Als Ausgangssignal wird ein Strom angeführt, der das Gate-Potential regeln soll (I_{fg}). Den dritten Eingang bildet ein Strom, der vom Rückkoppelnetzwerk kommt und den Regelkreis schließt. Genau auf diesen Strom I_{FB} , der rückgeführt wird, soll nun weiter eingegangen werden. Die eigentliche Regelung

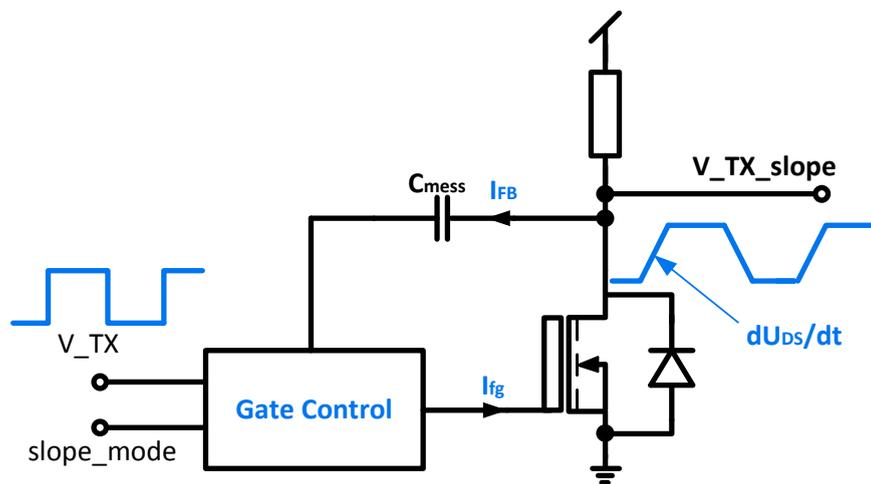


Abb. 4.4: Konzept zur Flankenregelung des LIN Bus'

findet nur während des Umschaltens statt, wenn also $\frac{dU_{DS}}{dt}$ ungleich Null wird. Ist dies der Fall, gilt für die Ladung Q_m im Kondensator C_{mess} :

$$Q_m = C_{mess} \cdot U_{DS} \quad (4.7)$$

Um nun noch die zeitliche Änderung dieser Ladung ins Spiel zu bringen, formt man Glei-

chung (4.7) um und führt anschließend die Ableitung nach der Zeit t ein [KMR06].

$$U_{DS}(t) = \frac{Q_m(t)}{C_{mess}} \quad (4.8)$$

$$\frac{dU_{DS}(t)}{dt} = \frac{dQ_m(t)}{dt} \cdot \frac{1}{C_{mess}} \quad (4.9)$$

$$I_{FB}(t) = \frac{dQ_m(t)}{dt} \quad (4.10)$$

$$\frac{dU_{DS}(t)}{dt} = I_{FB}(t) \cdot \frac{1}{C_{mess}} \quad (4.11)$$

So ergibt sich schließlich auch die Gleichung, mit der man für ein konstantes $\frac{dU_{DS}}{dt}$ und einer gegebenen Messkapazität C_{mess} den Ausgleichsstrom bestimmen kann:

$$I_{FB}(t) = C_{mess} \frac{dU_{DS}(t)}{dt} \quad (4.12)$$

Offensichtlich stellt sich dieser Strom in Abhängigkeit des Vorzeichens (steigende oder fallende Flanke) während des Umschaltens ein.

Als nächsten Schritt geht es an den Entwurf des in Abbildung 4.4 dargestellten Gate Control Blocks, dem eigentlichen Regler. Um einen geeigneten Regler für den Regelkreis zu finden wäre es naheliegend, ein Verfahren, wie beispielsweise das in [HD04] beschriebene Frequenzkennlinienverfahren (FKL), anzuwenden.

Jedoch stellt es sich als eher umständlich heraus, einen solchen Regler in integrierten Technologien bezüglich Pol- und Nullstellen im Frequenzbereich zeitkontinuierlich direkt zu dimensionieren.

Das Design wird ausgehend von der Struktur (Abbildung 4.3) aus analogen funktionalen Schaltungskomponenten zusammengesetzt und dimensioniert, obgleich werden Eigenschaften der Schaltung, welche die Stabilität betreffen, genau durch solche geeigneten Kriterien in Bezug auf Phasenreserve bzw. Amplitudenreserve sichergestellt, wie in [HD04] vorgestellt.

Ausgehend von den Informationen bezüglich verschiedener Betriebsmodi (verschieden einstellbare *Slew rates*, siehe Tabelle 4.1) beginnt man auf Konzept-Ebene mit dem Systementwurf. Das Ziel ist recht simpel formuliert: Ein konfigurierbares $\frac{dU_{DS}(t)}{dt}$.

Um den Schalttransistor einzuschalten, muss eine Spannung höher als die Schwellenspannung von Gate nach Source anliegen. Arbeitet man aber wie in diesem Beispiel mit Strömen als steuerbare Größen, muss man auf das hochohmige Gate nur eine bestimmte Zeit lang einen Strom einprägen, um die Spannung über besagte Schwellenspannung ansteigen zu lassen, so dass ein Drain-Source-Strom zu fließen beginnt. Dies soll in

AC Characteristics-LIN	Normal SM	Low SM	Fast SM	Flash Mode
Slew rate falling edge [V/ μ s]	-3 – -1	-1.5 – -0.5	-6 – -2	-24 – -8
Slew rate rising edge [V/ μ s]	1 – 3	0.5 – 1.5	2 – 6	8 – 24

Tab. 4.1: Betriebsmodi für LIN Transceiver

geregelter Form ablaufen.

$$U_g(t) = \frac{1}{C_{gg}} \int_{t_0}^{t_1} i(t) dt \quad (4.13)$$

$$C_{gg} = C_{gs} + C_{gd} \quad (4.14)$$

Dies kann mit Stromquellen bzw. Stromsenken erreicht werden. Betrachte man hierzu die in Abbildung 4.5 dargestellte Konfiguration der Stromquellen.

Ausgehend von einer Gate-Source Spannung von 0 V und $I_{FB} = 0$ (offener Kreis) lässt man den Parameter χ größer eins werden, wodurch mehr Strom in den hochohmigen Knoten U_G zufließt, als durch die untere Senke abfließen kann.

Als Konsequenz wird die Gate-Spannung so lange ansteigen, bis die obere Stromquelle aufgrund einer zu niedrigen Kanalspannung aus dem Sättigungsbereich herausfällt und ihren Stromfluss reduziert.

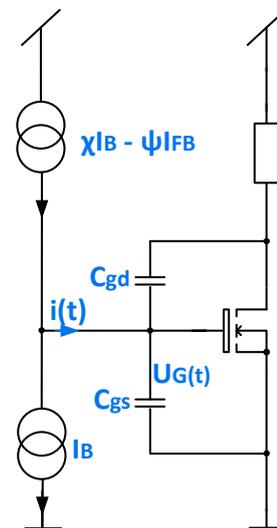


Abb. 4.5: Prinzip der Gate-Regelung

Betrachtet man nun vorhergehendes Szenario für den geschlossenen Regelkreis, wird die Gate-Spannung ansteigen, bis sie einen Wert erreicht hat, der geringfügig über der Schwellenspannung liegt.

Da der Transistor jetzt einen leitenden Kanal gebildet hat und ein Strom von Drain nach Source fließt, wird die Drain-Source Spannung sinken. Dadurch beginnt ein Feedback-Strom (4.12) durch die Kapazität C_{mess} zu fließen, welcher mit geeigneter Skalierung ψ dem zufließenden Strom der oberen Quelle entgegengewirkt.

Dieser Feedback-Strom wird natürlich nur solange fließen, solange sich die Drain-Source Spannung ändert ($\frac{dU_{DS}(t)}{dt} \neq 0$).

Wird das Verhältnis von Feedback-Strom zu Biasstrom geeignet gewählt, fließt während der fallenden Flanke von U_{DS} nahezu gleich viel Strom zu, wie auch durch die

Senke aufgenommen werden kann (in Wirklichkeit natürlich etwas mehr, andernfalls würde der Drain-Strom konstant bleiben!) und die Spannung am Gate bleibt ungefähr konstant.

Steigt die Gate-Spannung nun marginal an, fließt im nächsten Moment ein erhöhter Strom, welcher die Drain-Source-Spannung sinken lässt [IK04].

Ist die Schaltphase beendet und der Minimalwert von U_{DS} erreicht, versiegt auch der Feedback-Strom, was nun die Gate-Spannung weiter ansteigen lässt (Abbildung 4.6). In der Abbildung kann man auch deutlich das charakteristische Miller-Plateau erkennen [Gal93]. Des Weiteren kann auch der Drainstrom des Transistors betrachtet wer-

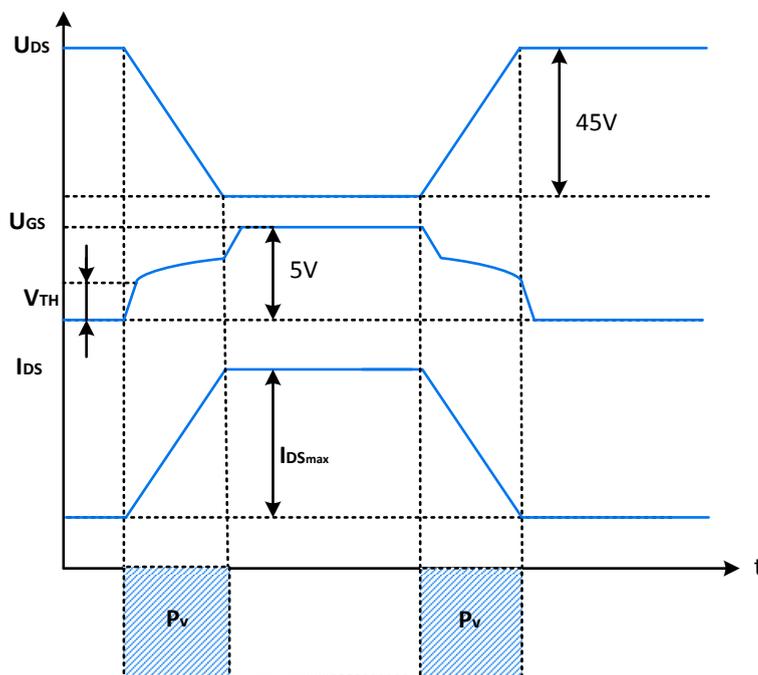


Abb. 4.6: zeitlicher Verlauf der Regelgrößen

den, welcher sich umgekehrt proportional zur Spannung ändert. Aus dem Grund, dass während dieser Schaltphasen weder Spannung noch Strom Null werden, kann man erkennen, dass die dadurch abfallende Verlustleistung proportional zur Schaltzeit ansteigt.

Hier ist bereits ein Kompromiss zwischen langer und kurzer Schaltzeit erkennbar. Für den Abschaltvorgang ist der Ablauf gleich, nur mit entgegengesetztem Vorzeichen des Feedback-Stroms und Vertauschen der aktiven und passiven Quellen. Abbildung 4.7 zeigt das Konzept für den Aufbau des Gate-Control Blocks in schematischer Form. Das Funktionsprinzip entspricht zuzüglich der Skalierung der Ströme dem oben erläuterten. Um nun ein P-Glied (bezüglich Strom) schaltungstechnisch realisieren zu können, bedient man sich einer recht einfachen Variante, nämlich Stromspiegeln (Abbildung 4.8). Ein Stromspiegel erzeugt aus einem Eingangsstrom einen proportionalen Ausgangsstrom.

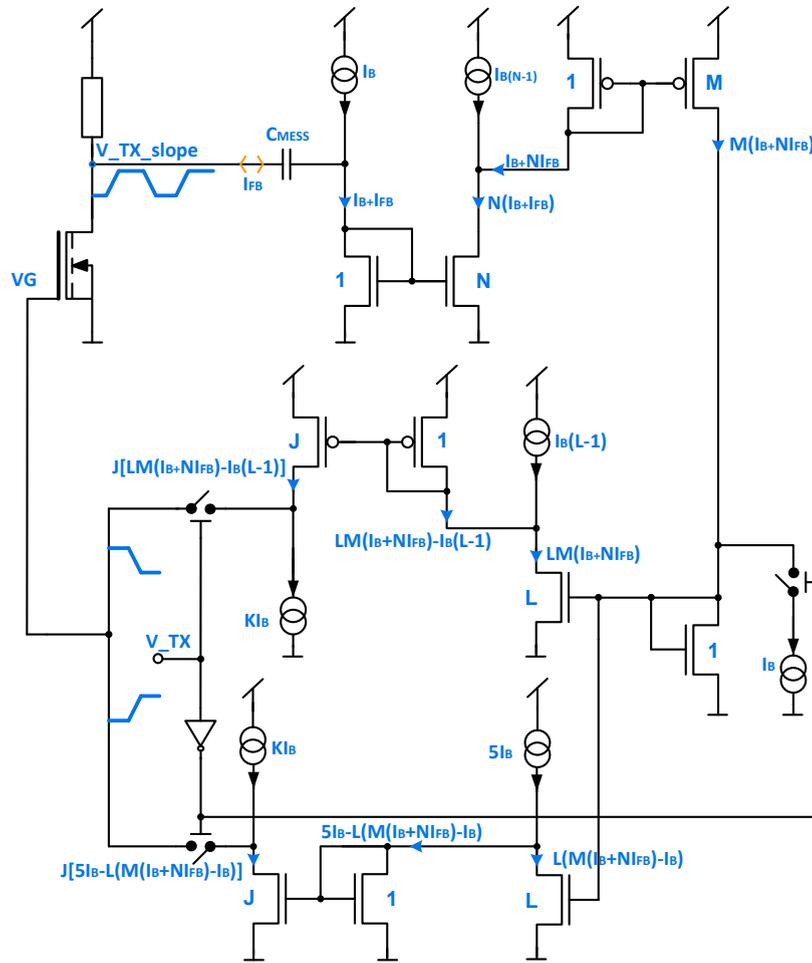


Abb. 4.7: Schaltungskonzept für LIN Gate Control

Um die Skalierung vornehmen zu können, wurden vier Stromspiegel verwendet, von denen die ersten beiden (bezeichnet mit N und M) für beide Flanken benutzt werden. Als Alternative dazu könnte man auch die Größe von C_{mess} variieren, was sich aber negativ auf die Fläche auswirkt.

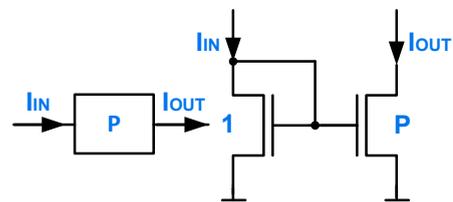


Abb. 4.8: Stromspiegel als P Glied

Betrachtet man den Strom für die steigende Flanke am Knoten V_G (durch Pfeile dargestellt) fließt in das Gate ein Strom von:

$$I_+ = J(LMI_B + LMNI_{FB} - (L - 1)I_B) \quad (4.15)$$

Der Strom durch die untere Senke wird nur mit K skaliert und ist unabhängig vom Feedback-Strom. Setzt man nun die Gleichung für den Knoten an und ersetzt für I_{FB}

Gleichung (4.12)

$$J(LMI_B + LMNI_{FB} - (L - 1)I_B) - KI_B = 0 \quad (4.16)$$

$$JLMI_B + JLMNI_{FB} - JLI_B + JI_B - KI_B = 0 \quad (4.17)$$

$$(JLM + J - JL - K)I_B = -JLMNI_{FB}, \quad (4.18)$$

erhält man eine Gleichung, mit der man die gewünschte *Slew rate* festlegen kann.

$$\frac{dU_{DS}}{dt} = I_B \frac{(K + JL - JLM - J)}{JLMNC_{mess}} \quad (4.19)$$

Für die steigende Flanke erhält man nach ähnlichem Ansatz:

$$KI_B - J[5I_B - L(M(I_B + NI_{FB}) - I_B)] = 0 \quad (4.20)$$

$$KI_B - 5JI_B + JLM I_B + JLMNI_{FB} - JLI_B = 0 \quad (4.21)$$

$$I_B(-5J - JL + K + JLM) = -I_{FB}(JLMN), \quad (4.22)$$

$$\frac{dU_{DS}}{dt} = I_B \frac{(5J + JL - JML - K)}{JLMNC_{mess}} \quad (4.23)$$

Mit den Gleichungen (4.19) bzw. (4.23) kann nun ein System implementiert werden, welches die in Tabelle 4.1 angegebenen Betriebs-Spezifikationen erfüllt.

Nun kann man den daraus entstehenden Regelkreis, ausgehend von Abbildung 4.3 mit den Gleichungen (4.18) und (4.22), auch detaillierter beschreiben. Die Stellgröße für das System wird mit dem Produkt $I_B\chi$ vorgegeben. Für die Implementierung wurden aufgrund dessen, dass die Stromspiegel im Spannungsbereich von 5 V liegen, die Faktoren J,L und K konstant gewählt und die verschiedenen Betriebsmodi wurden rein durch Variation von N, M und I_B eingestellt.

$$J = 1$$

$$L = 2$$

$$K = 2$$

$$(2M + 1 - 2 - 2)I_B + 2MNI_{FB} = 0 \quad (4.24)$$

$$I_B(2M - 3) + I_{FB}(2MN) = 0 \quad (4.25)$$

$$I_B\chi + I_{FB} = 0 \quad (4.26)$$

$$\text{mit } \chi = \frac{2M - 3}{2MN} \quad (4.27)$$

So kommt man zu Darstellung des Regelkreises:

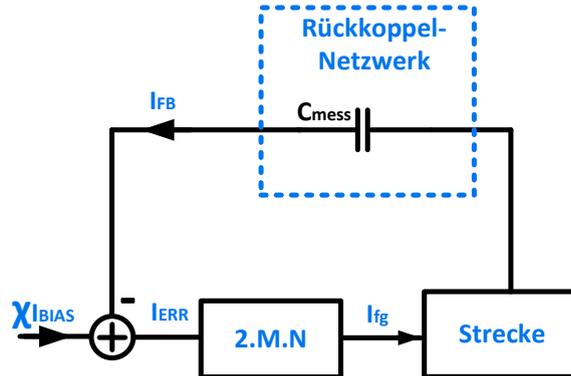


Abb. 4.9: Darstellung des Regelkreises

Für die steigende Flanke erhält man nach ähnlicher Überlegung:

$$I_B(-5 - 2 + 2 + 2M) + I_{FB}(2MN) = 0 \quad (4.28)$$

$$I_B(5 - 2M) + I_{FB}(2MN) = 0 \quad (4.29)$$

$$I_B\chi + I_{FB} = 0 \quad (4.30)$$

$$\text{mit } \chi = \frac{5 - 2M}{2MN} \quad (4.31)$$

Der Faktor χ ergibt sich also

$$\chi := \begin{cases} \frac{2M-3}{2MN} & \text{für fallende Flanke} \\ \frac{5-2M}{2MN} & \text{für steigende Flanke} \end{cases} \quad (4.32)$$

4.5 Implementierung für den Normal-Modus

Nachdem der Regelkreis jetzt beschrieben ist, kann man den nächsten Schritt in Richtung Implementierung vornehmen. Hier gilt es nun am Beispiel des Normal Modus (siehe Tabelle 4.1) die Unbekannten N , M , I_{FB} und I_B zu bestimmen.

Obgleich es zu Beginn etwas verwirrend erscheinen mag, dass alle vier Parameter bestimmt werden müssen, geht man von den bekannten Größen aus.

Aus der LIN Spezifikation erhält man den Wert für den R_{ON} Widerstand des Schalters, wodurch sich auch gleich die Weite dieses Transistors ergibt.

Als Nächstes ist bekannt, wie schnell man den Transistor einschalten muss, wonach sich auch der Strom errechnen lässt, der für diese Zeit auf das Gate gebracht werden muss. Aus der Simulation erhält man einen Wert für die Gate-Kapazität des DMOS-Transistors von 2,6 pF. Mit dem mittleren Wert von 3 V/ μ s im Fast-Mode kommt man so zu einem ersten Wert.

$$I_B = C \frac{dU}{dt} = 2,6 \text{ pF} \cdot 3 \text{ V}/\mu\text{s} = \underline{\underline{7,8 \mu\text{A}}}$$

Hinsichtlich benötigter Chip-Fläche sollten die Werte für N und M möglichst klein dimensioniert werden. Aus der Spezifikation ist ersichtlich, dass zwischen den Bereichen der Modi immer das konstante Verhältnis 2 liegt. Also sollte man für einen Modus dimensionieren und bei den anderen nur eine Größe teilen bzw. verdoppeln müssen.

Für den Normal-Modus ist der mittlere Wert für die Anstiegsgeschwindigkeit 2 V/ μ s. Also gilt es für dieses $\frac{dU_{DS}}{dt}$ aus Gleichung (4.19) N und M zu bestimmen.

Hier fällt als Nächstes auf, dass auch die Kapazität C_{mess} in der Formel enthalten ist und vorher bestimmt werden muss. Da der Feedback-Strom kleiner als der Biasstrom sein muss und auch wieder die Fläche eine Rolle spielt, wählt man einen Feedback-Strom von 1 μ A und erhält für C_{mess} :

$$C'_{mess} = I_{FB} \frac{1}{dU/dt} = 1 \mu\text{A} \cdot 1 \text{ V}/\mu\text{s} = \underline{\underline{1 \text{ pF}}}$$

$$C''_{mess} = I_{FB} \frac{1}{dU/dt} = 1 \mu\text{A} \cdot 3 \text{ V}/\mu\text{s} = \underline{\underline{333,33 \text{ fF}}}$$

für den *Slow-Mode* und den *Fast-Mode*. Da für den *Slow-Mode* ein anderer Parameter verändert werden wird, um Fläche zu sparen, wird C_{mess} mit **400 fF** gewählt.

Nun kann man mit Gleichung (4.19), C_{mess} und den bereits oben festgesetzten Werten

für J, L und K weiter rechnen. Durch Einsetzen kommt man auf Werte von M = 2 und N = 3, welche dann

$$\begin{aligned}\frac{dU_{DS}}{dt} &= I_B \frac{(K + JL - JLM - J)}{JLMNC_{mess}} = 10 \cdot 10^{-6} \frac{(2 + 1 \cdot 2 - 2 \cdot 2 - 1)}{1 \cdot 2 \cdot 2 \cdot 3 \cdot 400 \cdot 10^{-15}} \\ &= 10 \cdot 10^{-6} \cdot \frac{-1}{12 \cdot 400 \cdot 10^{-15}} = \underline{\underline{-2,083 \text{ V}/\mu\text{s}}}\end{aligned}$$

mit einem Wert von 10 μA für den Biasstrom -2,083 $\text{V}/\mu\text{s}$ ergeben. Für dieselben Werte für M und N ergibt sich für die steigende Flanke derselbe Wert mit geändertem Vorzeichen.

$$\begin{aligned}\frac{dU_{DS}}{dt} &= I_B \frac{(5J + JL - JML - K)}{JLMNC_{mess}} \\ &= 10 \cdot 10^{-6} \frac{(5 \cdot 1 + 1 \cdot 2 - 1 \cdot 2 \cdot 2 - 2)}{1 \cdot 2 \cdot 2 \cdot 3 \cdot 400 \cdot 10^{-15}} \\ &= 10 \cdot 10^{-6} \cdot \frac{1}{12 \cdot 400 \cdot 10^{-15}} = \underline{\underline{2,083 \text{ V}/\mu\text{s}}}\end{aligned}$$

Somit sind sämtliche Unbekannten für den *Normal-Mode* bestimmt. Für den *Low-Slope-Mode* wurde der Biasstrom für die gesamte Schaltung halbiert und für den *Fast-Slope-Mode* wurde N = 2 gewählt mit sonst gleichbleibenden Werten.

4.6 Schaltung des Gate Treibers

Die fertig implementierte Schaltung des *Gate Drivers* für den LIN Transmitter ist in Abbildung 4.10 dargestellt. Man erkennt die Feedback-Leitung, welche zum Eingang des ersten Spiegels bestehend aus N0 - N7 führt. Über die beiden Schalter TG3 und TG4 kann das Spiegelverhältnis von 1:1 bis 1:3 eingestellt werden.

Der nächste Spiegel (M7, M18, M22 - M25) ist mit P-MOS Transistoren realisiert, welche ebenfalls in die 1,5 Volt Spannungsklasse fallen.

Mit den Transistoren N8, N10, N11 und N13 wird eine geregelte Kaskode realisiert, welche über MN1 von der 5 Volt Versorgung abgetrennt wird. Besagte Kaskode dient dazu, den Biasstrom durch N8 vom 1,5 Volt-Bereich in den 5 Volt Bereich zu transferieren.

N9 und N12 bilden eine zu schaltbare Stromsenke, welche während der steigenden Schaltflanke benötigt wird. Der in Abbildung 4.7 dargestellte Spiegel, der mit L skaliert ist, wird hier in der Schaltung mit N14 - N19 und N24 realisiert, wobei MN2 und MN3 wieder für die Abtrennung dienen.

M28 und MP3 bilden die Quelle für den fünffach skalierten Biasstrom. Die Quellen und Senken für die fallende Flanke werden mit M31, M32, MP6, MP7, MN6, MN7, N22 und N23 gebildet. Den Spiegel für die steigende Flanke bilden M30, MP5, MN8, MN4, MN5, N20, N21.

Die Signale smode0/1 dienen zur Ansteuerung des Logikblocks, der wiederum die Schalterstellungen für die einzelnen Betriebszustände konfiguriert.

Auch ein Pegelwandler¹ wurde hinzugefügt, um die ankommenden Steuersignale der Logik von 1,5 V auf 5 V zu wandeln, damit die Schalter im 5 V Bereich auch mit der richtigen Spannung angesteuert werden können.

Um die Schaltung übersichtlich zu halten, wurden Teile für den *Power-Down-Mode* nicht eingezeichnet. Nach dem Regler soll nun die Implementierung des LIN-Schalters genauer behandelt werden. Abbildung 4.11 zeigt die Schaltung dazu.

Das zentrale Element bildet der Schalt-Transistor M0 selbst. Auch die oben behandelte Kapazität C_{mess} , welche das Rückkoppel-Netzwerk bildet, ist leicht zu finden. R5 ist in Serie zum Kondensator, um hochfrequente Störungen zu dämpfen.

Die Kapazität C1 dient in Kombination mit R1 zur Kompensation der Regelschleife, da das Gate den einzigen hochohmigen Knoten in der Schaltung bildet.

N1 dient als Schalter um das Gate, wenn nötig, auf Masse zu ziehen. Diode D1 beschränkt die Spannung am Gate. Der Widerstand R3, welcher auch über einen Pin nach außen geführt wird, führt zu einem weiteren Block für die Überstrom-Erkennung.

M1 und R4 bilden den Schutz gegen elektrostatische Entladungen (ESD), Hochspannungspulse werden gegen Masse abgeleitet. DIO1 ist ein sogenannter Rückstrom-Schutz², der eine Diode realisiert. Sollte die Spannung am Bus, speziell die Spannung am Drain von M0, unter das gemeinsame Masse-Potential absinken, würde die parasitäre Substrat-Diode in Durchlassrichtung betrieben, was einen Stromfluss zum Substrat zur Folge hätte. So wird durch DIO1 ein Stromfluss von Masse zum Bus verhindert. Ähnlich sperrt DIO2, sollte das Bus-Potential über die Batteriespannung ansteigen.

D2 und D3 sind anti-seriell verschaltet und dienen ebenfalls als ESD Schutz. Bedingt durch die Versorgungsspannung und dem für den Pin spezifizierten ESD-Prüfimpuls [Wap07], fällt die Stärke der Schutzdioden aus. Im normalen Betrieb beeinflusst dieses Gebilde die Schaltung nicht, erst bei höheren Spannungen leitet D2 hohe Spannungspulse nach Masse ab.

Zuletzt bleibt nur noch der Widerstand R7, der als interner Pull-Up Widerstand fungiert.

¹engl. Levelshifter

²engl. Reverse Current Diode

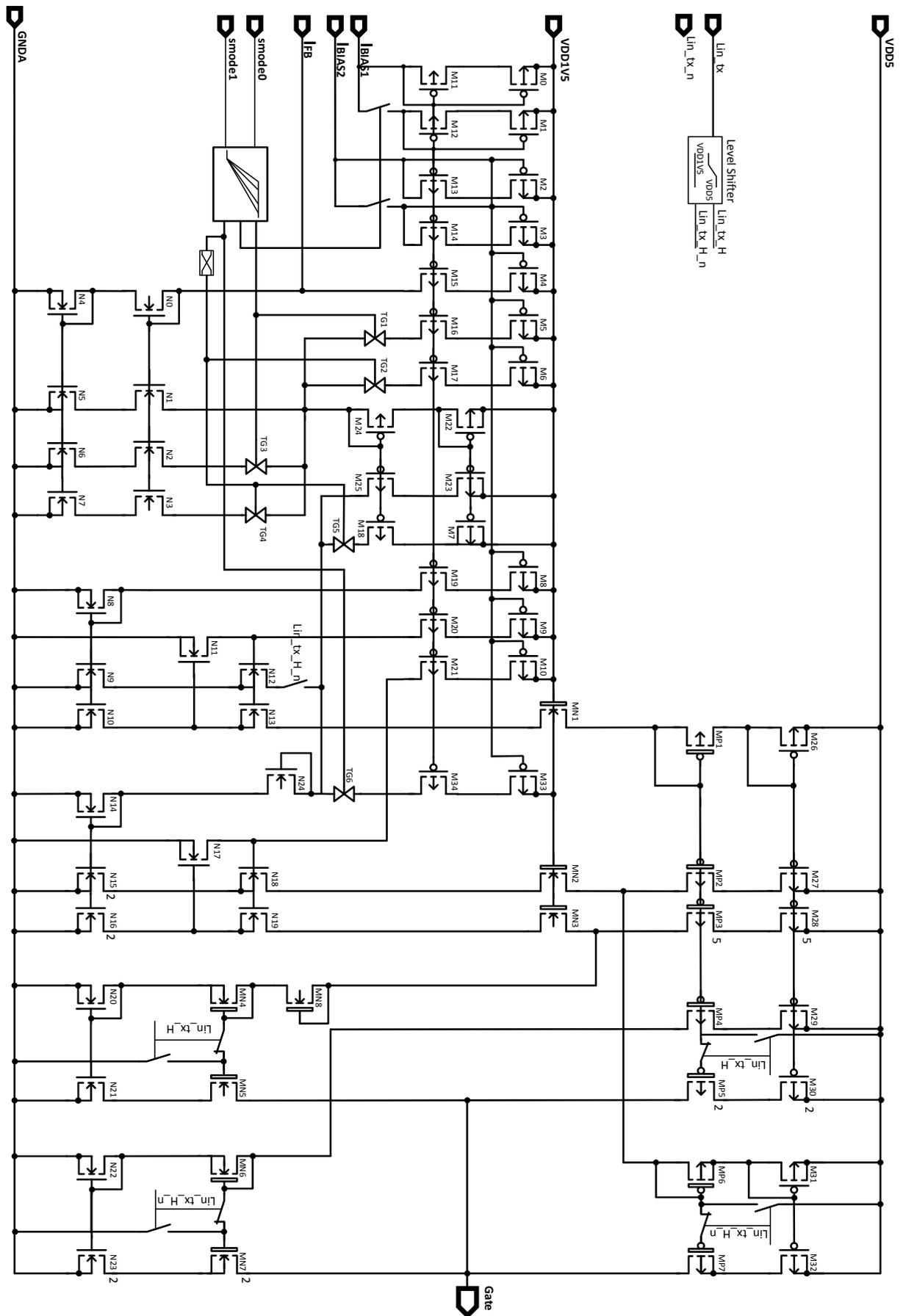


Abb. 4.10: Schaltung des LIN Reglers

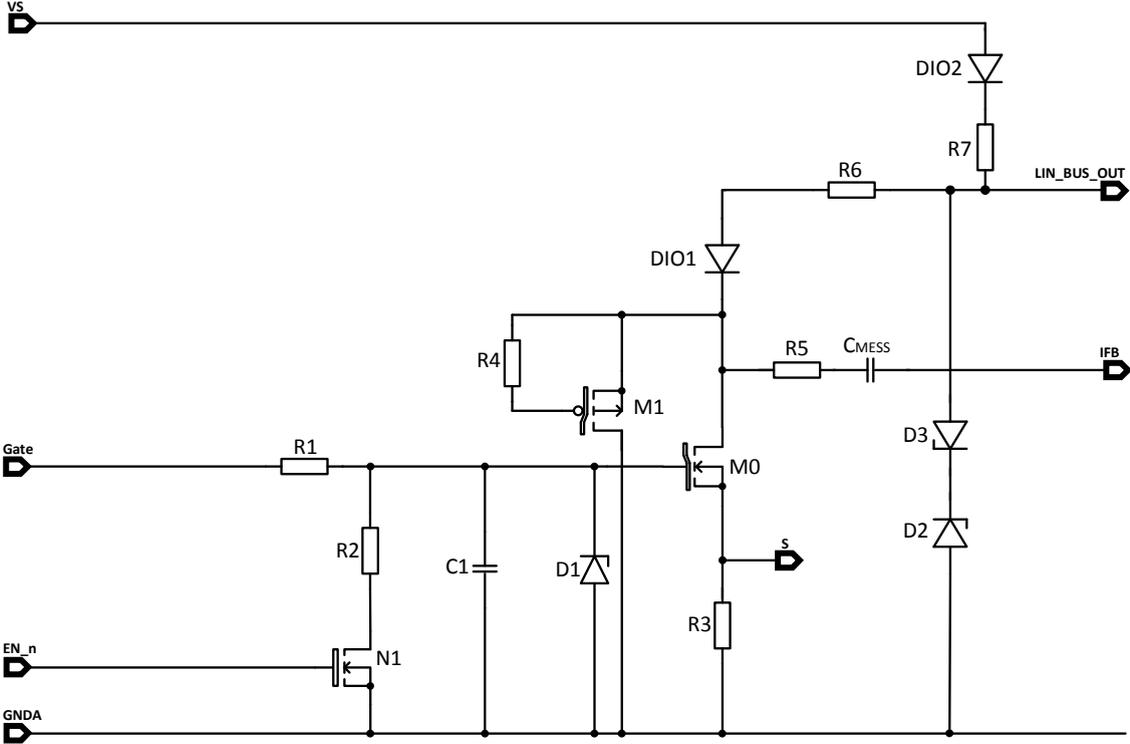


Abb. 4.11: Schaltung des LIN Schalters

4.7 Stabilität des Regelkreises

Die Stabilitätseigenschaften des Regelkreises wurden durch Simulation bestimmt. Hierzu wurde der Kreis für AC Betrachtungen mit Schaltern aufgetrennt und die Phasenreserve betrachtet. Mit einer Kompensationskapazität zwischen Gate und Source des Schalttransistors wurde der Kreis kompensiert.

Für die Simulation selber ist es wichtig, den Arbeitspunkt richtig zu wählen. Man muss also einen passenden Ausgangspunkt für die AC Analyse einstellen. Hierzu simuliert

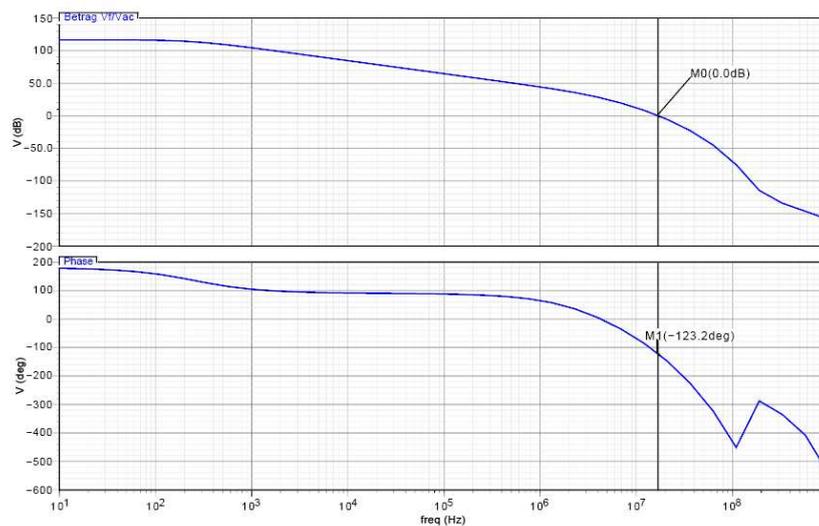


Abb. 4.12: Bode Diagramm der Rückkopplungsschleife

man transient bis zu einem Zeitpunkt während des Umschaltens und bricht dann ab, um die AC Analyse in diesem Punkt durchzuführen.

Es ist auch wichtig, mehrere verschiedene Bereiche zu betrachten, da sich die Gate-Kapazität während des Einschaltvorgangs ändert. Deswegen sind die Zeitpunkte am Beginn der steigenden Flanke bzw. am Ende der fallenden Flanke am kritischsten, da hier die Gate Kapazität am kleinsten ist und die Kompensation somit schwächer wirkt.

4.8 Simulationsergebnisse

Abbildung 4.13 zeigt den Verlauf der Gate-Source-Spannung (grün), der Drain-Source-Spannung (blau) und des Drain-Source-Stromes (rot). Die dargestellten Verläufe sind für den Normal-Modus simuliert worden.

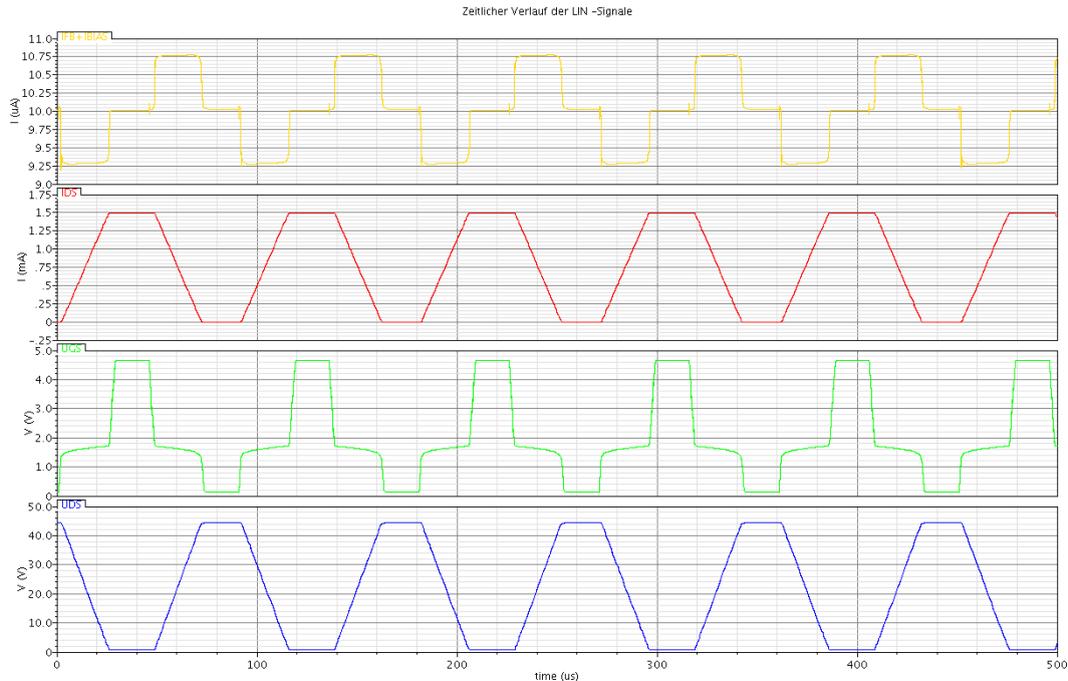


Abb. 4.13: Simulationsergebnisse im Normal-Modus

Auch der Eingangsstrom des ersten Stromspiegels, der sich aus dem Rückkoppelstrom und dem Biasstrom zusammensetzt, ist dargestellt. Man erkennt genau, dass sich während der fallenden Flanke von U_{DS} der Rückkoppelstrom vom Biasstrom subtrahiert und sich während der steigenden Flanke addiert.

In den Abbildungen 4.14, 4.15 und 4.16 sind die zeitlichen Verläufe des LIN-Transmitter-Signals für mehrere Temperaturen dargestellt.

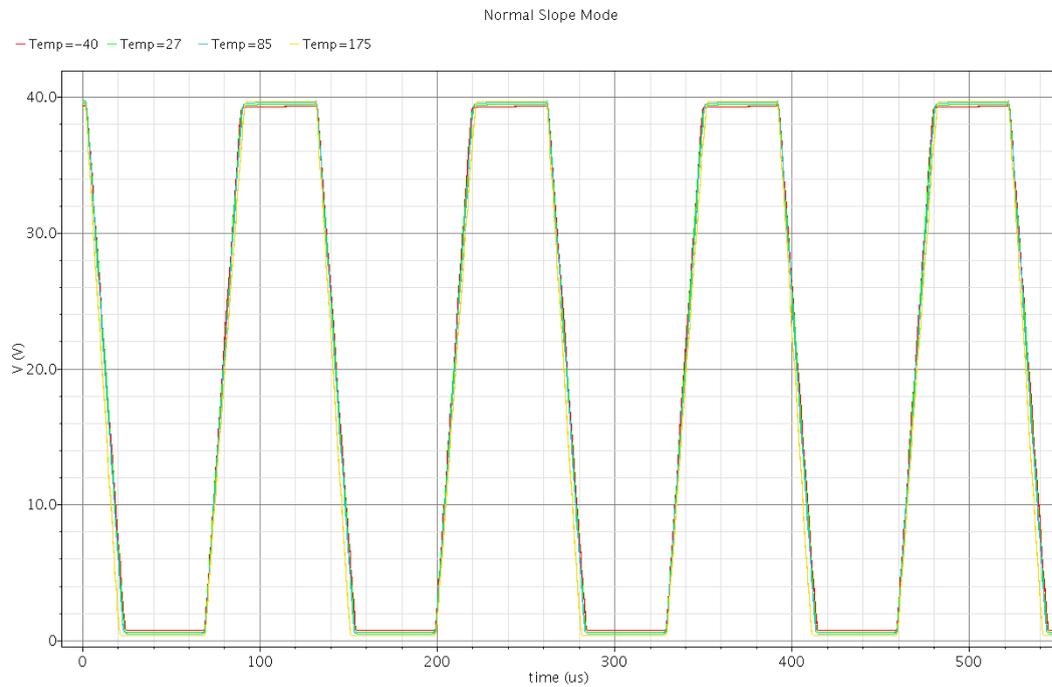


Abb. 4.14: Zeitlicher Verlauf des LIN-Bus Signales im *NORMAL-Slope-Mode* für -40° , 27° , 85° , und 175°C

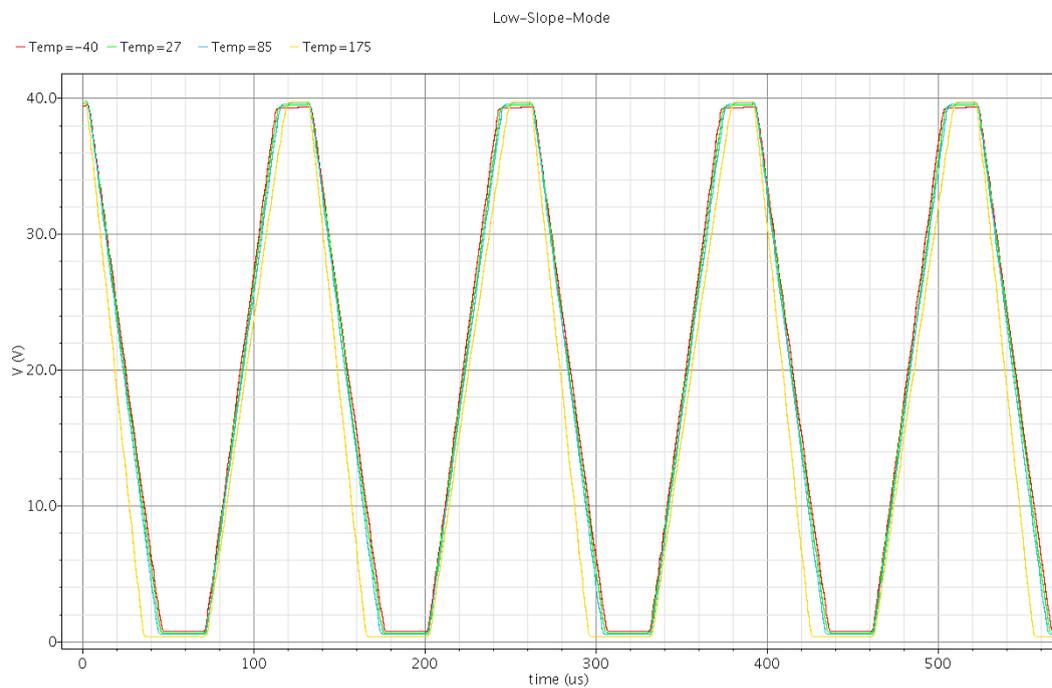


Abb. 4.15: Zeitlicher Verlauf des LIN-Bus Signales im *LOW-Slope-Mode* für -40° , 27° , 85° , und 175°C

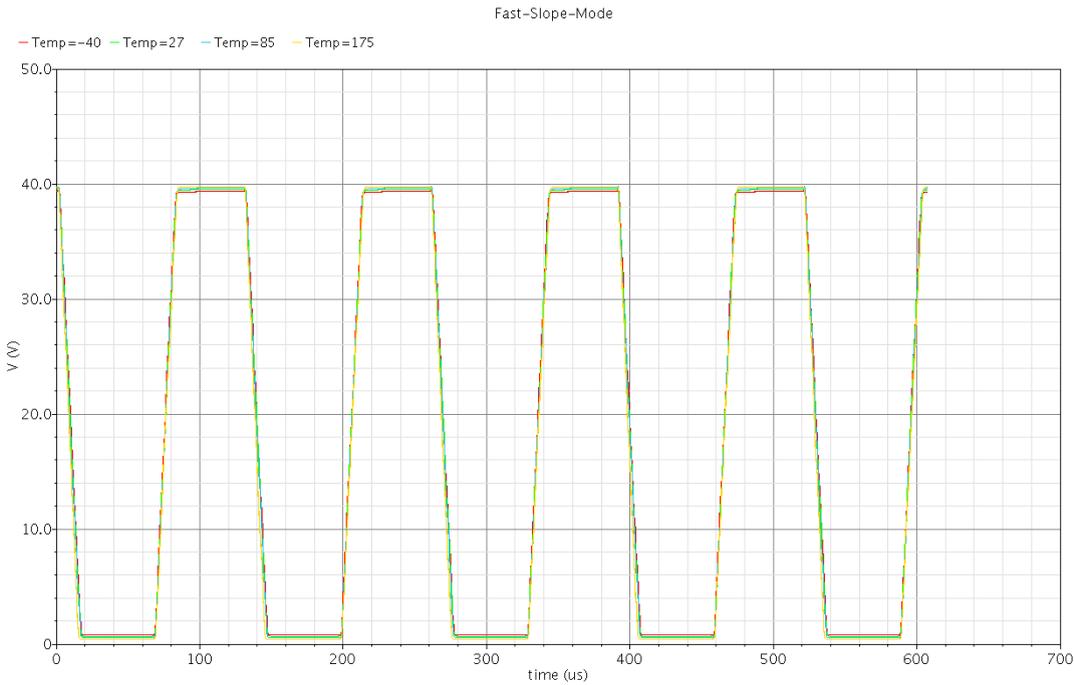


Abb. 4.16: Zeitlicher Verlauf des LIN-Bus Signales im *FAST-Slope-Mode* für -40° , 27° , 85° , und 175°C

4.9 Layout

In Abbildung 4.17 ist das Layout des LIN-Treibers dargestellt. Abbildung 4.18 zeigt die gesamte Top-Zelle des LIN-Transmitters. Im linken unteren Teil befindet sich der Gate-Treiber, welcher nur einen geringen Teil der gesamten Schaltung bildet.

Den größten Teil bilden die ESD Schutzkomponenten. Vor allem D2 und D3, die sich im rechten Teil des Top-Layouts befinden, nehmen den größten Anteil der benötigten Fläche in Anspruch. Der Grund dafür ist die hohe geforderte ESD-Festigkeit, da der Pin direkt mit der Busleitung, die das Steuergerät verlässt, verbunden ist.

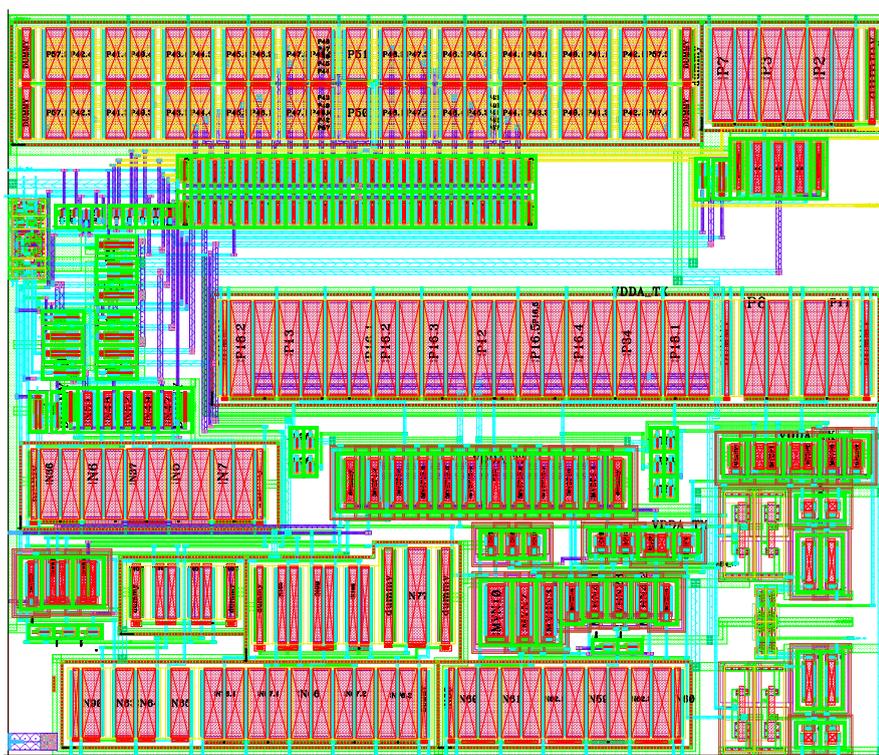


Abb. 4.17: Layout des LIN Gate-Driver Moduls

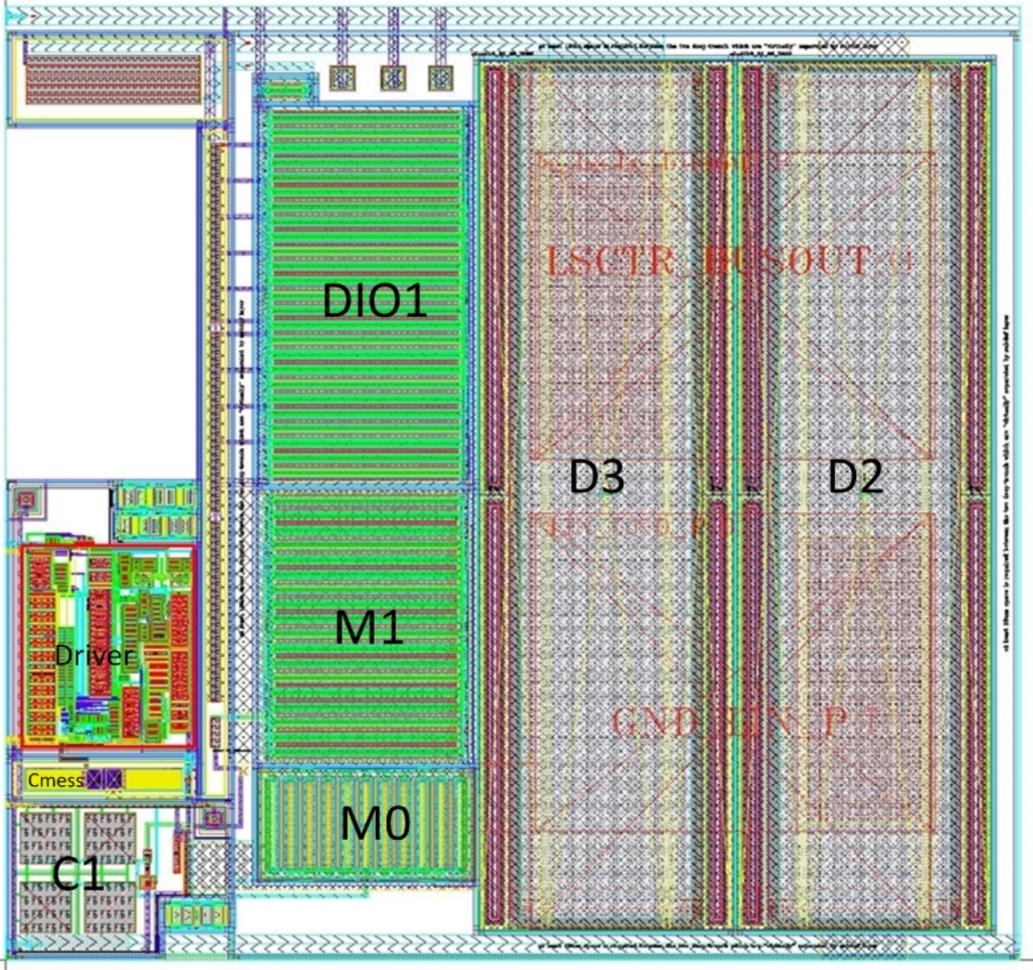


Abb. 4.18: Top-Layout LIN Modul

4.10 Abschließende Betrachtungen

Die Applikation des LIN Transmitters umfasst ein System, das darauf ausgerichtet ist, Informationsaustausch durchzuführen. Dies geschieht auf physikalischer Ebene mit Spannungspulsen, wobei die Leistung eher unmaßgeblich ist. Einer der weitaus wichtigsten Ziele ist es, so eine Implementierung hinsichtlich Temperatur, Prozessschwankungen und *Matching* über einen breiten Bereich funktionsfähig zu entwerfen.

Eine weitere Herausforderung stellt sich durch den sich ändernden Arbeitspunkt. Grundsätzlich geht man bei der Strecke von einem linearen zeitinvarianten System (LZI) aus. Da aber der Schalttransistor bezüglich der Drain-Source-Spannung einen Bereich von 0-40 Volt durchläuft, was auch Änderungen bezüglich parasitärer Parameter wie beispielsweise der Gate-Drain Kapazität mit sich bringt, kann man prinzipiell nicht mehr von einem zeitinvarianten System ausgehen.

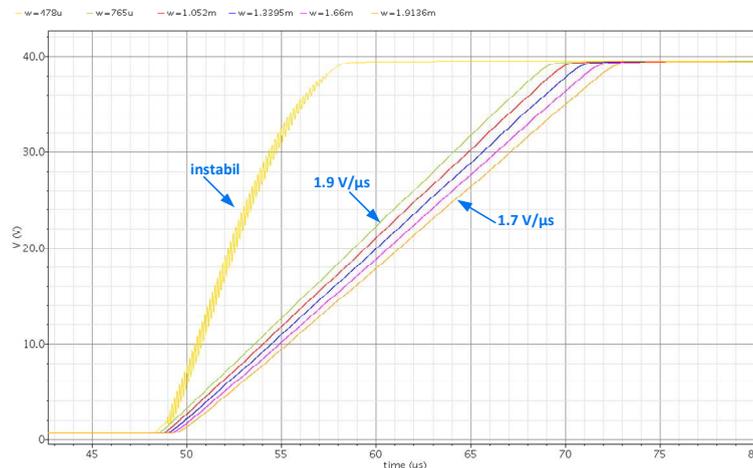


Abb. 4.19: LIN Flanke mit verschiedenen Weiten des Schalt-Transistors

Eine weitere Einschränkung in der analogen Implementierung ist vor allem, dass der Regler für genau einen Schalttransistor entworfen wird, wobei sehr viel Aufwand für die Realisierung mehrerer Betriebsmodi aufgewendet wurde. Abbildung 4.19 zeigt die LIN-Flanke, wobei die Weite des DMOS variiert. Man sieht, dass die *Slew rate* ebenfalls variiert, was vor allem auf die unterschiedliche Gate-Kapazität zurückzuführen ist. Vor allem bei kleinerer Weite, als die, für die die Schaltung dimensioniert wurde, wird das Stabilitätsverhalten (Phasenreserve) schlechter.

Vor allem aber in Hinblick auf Timing stellt sich die analoge Variante als unflexibel heraus, da zur Änderung der *Slew rate* auch die Schaltung geändert werden müsste. Da sich diese Hürden bei einer rein analogen Implementierung in den Weg eines Designers stellen, soll nun der nächste Schritt hin zu einer digitalen Implementierung gehen.

5 Digitale Flankenregelung

Nach der Problemstellung des [LIN Transmitters](#) soll in diesem Kapitel nun eine weitere Anwendung zur Flankenregelung untersucht werden. Anhand einer Applikation für Einspritzsysteme soll gezeigt werden, wie eine Flankenregelung digital implementiert wird. Vordergründig ist auch die Spezifikation der dazu benötigten Komponenten.

5.1 Leistungsregelung mit Pulsweiten-Modulation (PWM)

Um prinzipiell umgesetzte Leistung in einem Aktuator regeln zu können, geht man von einer benötigten Maximalleistung aus und regelt mit Hilfe eines veränderbaren Widerstandes die Leistung im Aktuator. Der Nachteil dieser sehr rudimentären Methode liegt in der Verlustleistung, die an besagtem veränderbaren Widerstand, welcher in der Regel durch einen [MOSFET](#) realisiert wird, abfällt. Eine Möglichkeit diese Verlustleistung zu umgehen bietet die Pulsweiten-Modulation. Das Prinzip ist einfach erklärt: Schaltet man die Stromzufuhr zu einem Aktuator für eine gewisse Zeit ein, um ihn danach für eine davon abhängige Pausenzeit wieder auszuschalten, kann man über das Verhältnis von aktiver Zeit zu Pausenzeit die zugeführte Leistung kontrollieren. Es wird also mit einer konstanten Frequenz, jedoch mit veränderlicher Pulsweite die zugeführte Leistung zwischen 0% und 100 % angepasst [[Krü04](#)].

Der elektronische Schalter wird digital angesteuert und befindet sich entweder im leitenden Zustand bzw. im Sperrzustand. Dadurch wird die Verlustleistung drastisch reduziert, jedoch mit einem erheblichen Nachteil: Diese *schnellen* Schaltflanken und hohen Taktraten bieten massives Potential für hochfrequente elektromagnetische Störungen, wie bereits in Kapitel 2 erläutert, welche sowohl in das Bordnetz zurück gespeist, aber auch in die Umgebung abgestrahlt werden können. Um diesen potentiellen Störungen beizukommen, werden Entstörfilter eingesetzt, welche unter Umständen¹ nicht mit integriert werden können, sondern extern (*Off-Chip*) angebracht werden [[Krü04](#)].

¹Bsp. externe veränderliche Last - entsprechend veränderliche Filter

Mit diesen externen Komponenten ist die Flankenkontrolle nur sehr ungenau möglich, da Bauteiltoleranzen (**MOSFET**, R, C) sehr stark einwirken. Dies fordert ein individuelles *Trimming* und Anpassung der Werte für unterschiedliche FETs.

Zusätzlich stellen diese Bauteile, vor allem sollte eine Integration nicht möglich sein, ein weiteres Risiko bezüglich Zuverlässigkeit dar.

Das Ziel dieses Fallbeispiels zur Regelung von Einspritzsystemen ist es, an Stelle dieser externen Filterkomponenten eine geregelte Schaltflanke zu realisieren, sodass die Notwendigkeit dieser externen Bauelemente damit nicht mehr gegeben ist.

5.2 PWM geregelte Systeme im KFZ

Nach der Betrachtung von **PWM** basierter Regelung sollen nun typische Anwendungsgebiete im KFZ erörtert werden. Ein Beispiel für ein solches Gebiet ist die Regelung von Gleichstrom-Motoren. Mit Applikationen wie **EPS** oder Active Front Steering (**AFS**), wird durch eine elektrische Servolenkung die Betätigung des Lenkrads unterstützt. Auch Kombinationen von elektrischer und hydraulischer Unterstützung (Überlagerung) werden in moderne KFZ eingebaut [Ise06].

Ein anderes Einsatzgebiet bilden elektronisch geregelte Einspritzsysteme, bei denen ebenfalls sehr hohe Leistungen zugeführt werden müssen, um ein gefordertes Drehmoment zu erreichen. Zum Zuführen des Kraftstoffes in die Brennkammer werden Magnetventile verwendet, die durch Zylinderspulen (*Solenoid*) ein magnetisches Feld erzeugen und durch die resultierende Kraftwirkung ein mechanisches Ventil öffnen. Diese Magnetventile werden durch elektronische Steuergeräte angesteuert.

5.3 Einspritzsysteme im Automobil

In modernen Kraftfahrzeugen steht nicht nur ein hoher Wirkungsgrad im Vordergrund sondern auch der Umstand, möglichst wenig Emissionen an die Umwelt abzugeben. Moderne Verbrennungsmotoren sind deshalb mit Systemen ausgerüstet, welche die Verbrennung des Kraftstoff-Luft-Gemisches möglichst vollständig gewährleisten. Dazu werden Einspritzsysteme benötigt, die es erlauben, den Zeitpunkt der Einspritzung sowie die benötigte Menge sehr genau zu dosieren. Systeme, wie beispielsweise Pumpe-Düse oder aber auch *Common-Rail*, arbeiten mit Einspritzdrücken von 200 bar bis über 2000 bar, um eine feine Zerstäubung des Kraftstoffs und in weiterer Folge eine vollständige Verbrennung mit wenig Rußemission zu ermöglichen. Solche Einspritzsysteme werden nicht zuletzt wegen der hohen Stückzahl, sowie den Anforderungen an Zuverlässigkeit durch hochintegrierte Elektronik gesteuert [Rei07].

5.4 Spezifikation der Flanke

Als Vorgabe für die Problemstellung sei die Anstiegszeit sowie der Bereich der Spannung gegeben. Das Ziel ist es, ein realistisches Konzept für die Regelung der Flanke zu erarbeiten, die für die Implementierung notwendigen Komponenten zu ermitteln und deren Parameter zu definieren.

	min	nom	max
Anstiegszeit	-	100 ns	-
Spannungsbereich	-	73 V	80 V
Slew rate	-	730 V/ μ s	-
Rs	-	2 Ω	-
Solenoid	-	20 mH	-

Tab. 5.1: Spezifikation der Flanke

5.5 Ausgangsstruktur High-Side Schalter

Abbildung 5.1 zeigt den Ausgangspunkt des Systems für die Einspritz-Regelung. Die Zylinderspule (*Solenoid*) wird mit einem *High-Side* Schalter HS_1 angesteuert. Der Schalter HS_1 selbst wird über einen Inverter angesteuert, wobei das Schaltsignal mit den dargestellten externen RC-Filtern bandbegrenzt wird. Diese Filter müssen je nach Schalttransistor adaptiert werden und stellen somit auch einen finanziellen Aufwand dar.

LS_1 ist während des Betriebs immer eingeschaltet und trägt zur Regelung nicht bei. Dieser *Low-Side* Schalter dient dazu, vor dem Einschalten die Source-Spannung zu definieren, vor allem aber der Systemsicherheit im Hinblick auf Systemausfälle, um den Stromfluss gegebenenfalls zu unterbrechen. Der eigentliche Regeltransistor ist HS_1.

Um ausreichend Leistung zuführen zu können, wird die Versorgungsspannung auf 73 Volt (V_{BOOST}) erhöht. Da während des Einschaltens von HS_1 das Gate immer mindestens ein um die Schwellenspannung höheres Potential als der Source-Knoten besitzen muss, um den Umschaltvorgang nicht zu unterbrechen, benötigt man eine eigene Versorgungsspannung für das Gate.

Da dies noch mehr Aufwand bedeutet, bedient man sich einer Lösung, die mit *Bootstrapping* arbeitet. Diese Methode arbeitet mit einer an HS_1 angepassten Kapazität und funktioniert folgendermaßen: Im ausgeschalteten Zustand fällt nahezu die gesamte Spannung am Schalter HS_1 ab. Das bedeutet das Source-Potential befindet

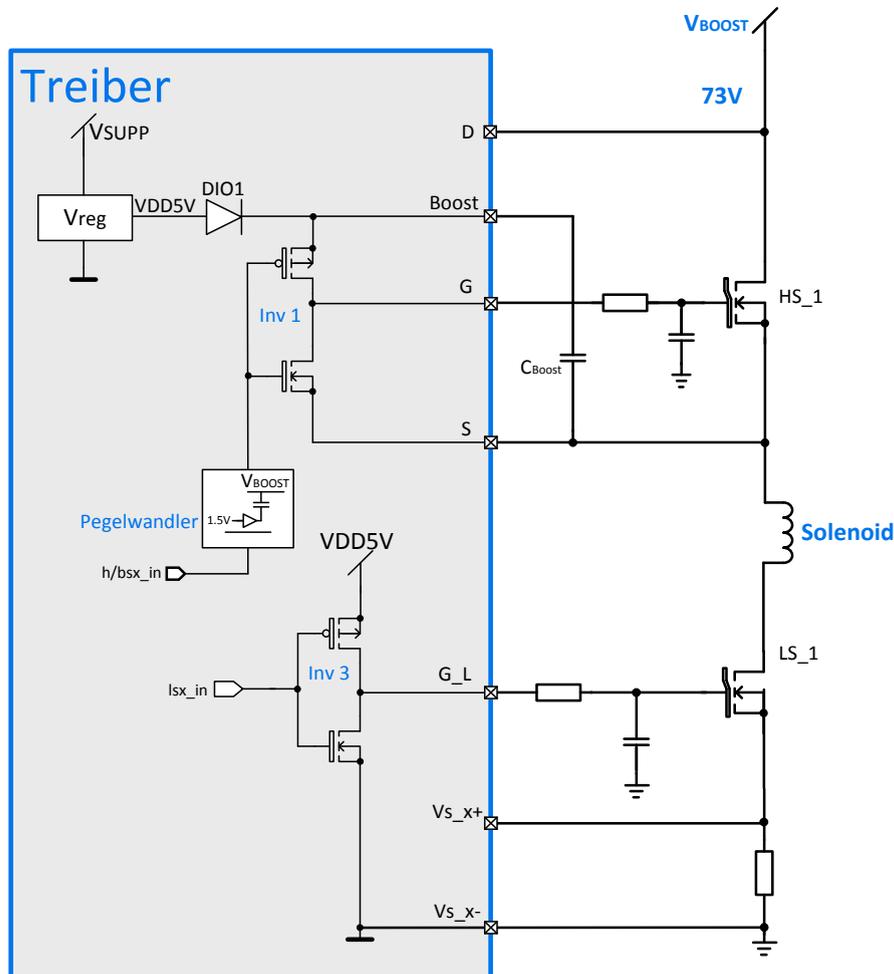


Abb. 5.1: High-Side Treiber Konzept

sich ungefähr bei 0 V. Die Boost-Kapazität wird nun auf die maximale Gate-Source Spannung aufgeladen. Initiiert man nun den Schaltvorgang durch Aufladen des Gates über Inv1, beginnt der Transistor zu leiten und die Source-Spannung in Folge zu steigen. Da C_{BOOST} noch immer auf ca. 5 Volt aufgeladen ist, steigt auch diese Spannung mit. Auf diese Art liegt der Punkt Boost immer um 5 V höher als das Source-Potential.

Ein erheblicher Nachteil dieser Methode ist, dass die Pulsweite bzw. das Puls-Pausen-Verhältnis nicht mehr 100 % betragen kann. Die Ursache liegt darin, dass aufgrund von Verlusten wie Leckströmen die Spannung an der Boost-Kapazität abnimmt.

Das zum Schalten benötigte PWM-Signal kommt über den internen Pin h/bsx_in auf den Eingang eines Pegelwandlers, welcher das Schaltsignal auf den zum Schalten notwendigen Pegel bringt, um den Inverter Inv 1 anzusteuern.

Generell wird das Gate des Transistors HS_1 durch Inverter angesteuert und "hart" eingeschalten. Ziel ist es, das vorliegende System so umzubauen, dass die Schaltflanke geregelt wird und so auf die externen Filter gänzlich verzichtet werden kann.

5.6 Konzepte für die Regelung

Abbildung 5.2 veranschaulicht die Ansteuerung des Schalt-Transistors durch geregelte Stromspiegel. Sofort fällt die bereits aus dem Kapitel 4 bekannte Kapazität C_{mess} auf, die auch in dieser Applikation den Rückkoppelpfad bildet. Das Bild zeigt den digitalen

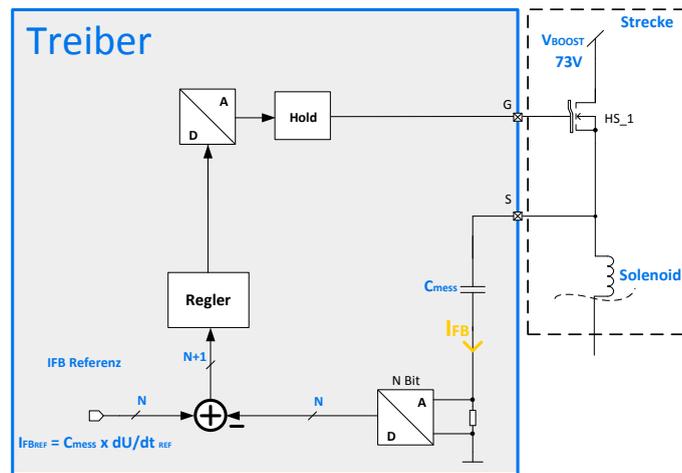


Abb. 5.2: Konzept mit *Slew rate Control*

Regelkreis, bestehend aus einem Analog/Digital Umsetzer (ADC), einem Regler, einem Digital/Analog Umsetzer (DAC) sowie der Strecke. Das Grundkonzept sieht so aus, dass während der Schaltphase ein Feedback-Strom vom Ausgang der Strecke (*Source*) mit einem Referenzsignal in digitaler Form verglichen wird und die Differenz dem Regler zugeführt wird, welcher die entsprechenden Ausgangskoeffizienten berechnet, um die Vorgabe durch die Referenz erfüllen zu können.

Obleich das Strukturbild sehr einfach aussieht, gibt es zahlreiche unbekannte Größen, die für die Implementierung noch bestimmt werden müssen. Ein wichtiger festzulegender Parameter ist die Abtastrate, mit der der Regelkreis arbeitet. Und vor allem die entstehenden Verzögerungszeiten des Kreises. Der ADC wird eine Zeit für die Wandlung benötigen, und auch der D/A Umsetzer wird eine Stellzeit benötigen.

Eine weitere Frage stellt sich nach der Anzahl der benötigten Bits sowie der zeitlichen Auflösung der Umsetzer.

Eine andere Möglichkeit, die Regelung durchzuführen, wird in Abbildung 5.3 dargestellt. Hier arbeitet ein adaptives Filter an der Korrektur des Regelfehlers. Für dieses Konzept wird allerdings ein weiterer Block zur Identifikation der Strecke benötigt, was bedeutet, dass bei bekannter Ordnung der Strecke deren Parameter (Gate-Kapazität, g_m, \dots) durch einen Algorithmus erkannt werden.

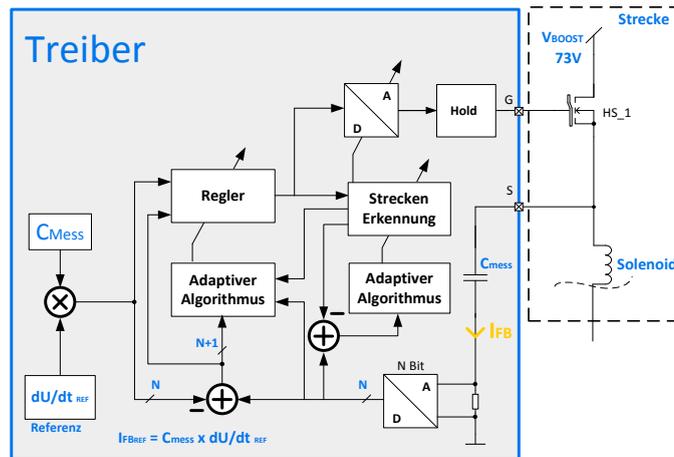


Abb. 5.3: Konzept für adaptive Regelung der Slew rate

Diese Lösung ist sehr aufwändig jedoch sehr vielversprechend, wollte man die externe Beschaltung ändern, ohne das System neu zu justieren. So müssten eventuell nur die Reglerkoeffizienten neu angepasst werden.

Abbildung 5.4 zeigt ein weiteres Konzept, das mit Zeitmessung arbeitet. Hier wird mit dem Komparator bei jeder steigenden Taktflanke überprüft, ob der Sollwert bereits erreicht wurde. Sollte dies nicht der Fall sein, wird durch den Regler entsprechend nachkorrigiert.

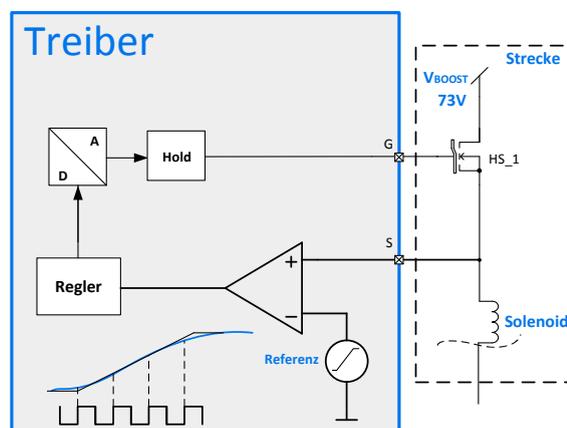


Abb. 5.4: Konzept für Regelung mit 1 Bit ADC

Dieses Konzept arbeitet ohne eine Kapazität für die Messung. Um dies zu realisieren, benötigt man schaltungstechnische Maßnahmen im Hochvolt-Bereich, da der Eingang

des Komparators die Spannung bis 80 V sieht. Dies kann man mit einer entsprechenden Skalierung durch Hochvoltabtrennung erreichen. Besonders nachteilig wird in diesem Zusammenhang die benötigte Fläche für die Aufbereitung des Eingangssignals, da man anstatt einer einfachen Kapazität mehrere hochvoltfähige Bauelemente benötigen wird.

5.7 Systemdefinition und Dimensionierung

An dieser Stelle soll mit der genaueren Definition der Komponenten, die für die Regelung benötigt werden, begonnen werden. Das System ist mit seinen grundlegenden Komponenten in Abbildung 5.5 dargestellt. Nach der Identifikation und Spezifikation der Komponenten wird im folgenden Schritt von jeder Komponente ein Modell angefertigt, das Gesamtsystem zusammengebaut und die Funktion kann mit Simulationen verifiziert werden. Entspricht die Funktion des Systems den Erwartungen, kann damit

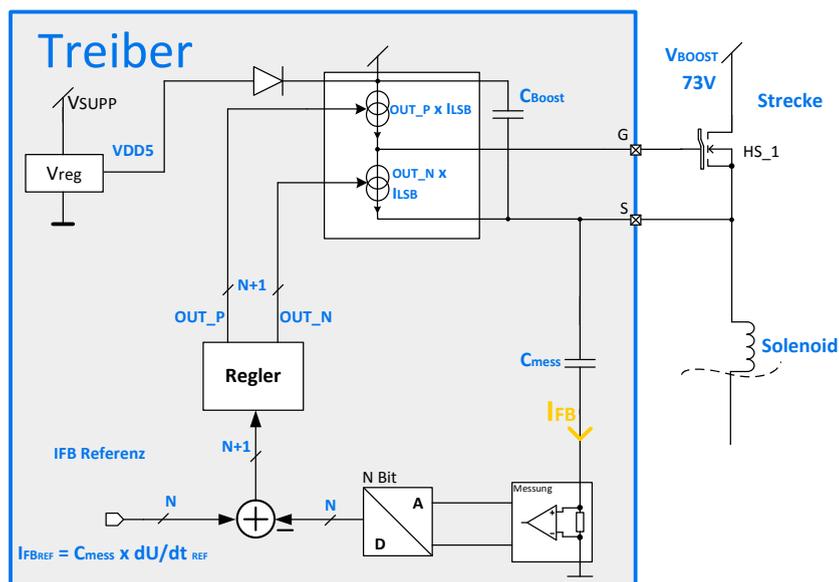


Abb. 5.5: Systemdefinition des Regelkreises

begonnen werden, die einzelnen Modelle, vor allem hinsichtlich zeitlicher Abhängigkeiten, genauer zu beschreiben.

Die Mess-Kapazität C_{mess} liefert während des Umschaltens einen Strom proportional der *Slew rate*. Mit dem dargestellten Messumformer wird dieser Strom rund um einen Sollwert in eine Spannung umgewandelt, welche im Anschluss dem ADC zugeführt und digitalisiert wird.

Über ein Summationsglied, welches aus einem Addierer besteht (mit integrierter Komplementbildung für die Subtraktion), wird das dadurch entstehende Fehlersignal dem Regler zugeführt. Dieses Glied kann auch in den Regler integriert werden.

Ein passender Regelalgorithmus steuert den DAC mit Stellwerten (immer noch digital) für die Stromquellen des DACs an. Mit diesen Stromquellen kann Ladung auf das Gate fließen bzw. abgeleitet werden, wodurch der Schaltvorgang letztendlich beeinflusst wird und sich so der Kreis schließt.

Nach dieser Identifikation der Komponenten geht es nun daran, die notwendigen Parameter für die einzelnen Komponenten zu bestimmen.

Der Wert für die Messkapazität richtet sich nach der Messeinrichtung und wird vorerst mit 1 pF angenommen. Soll nun eine Spannung von 0 V bis 73 V in einem minimalen Zeitabschnitt von 100 ns geschaltet werden, beträgt die *Slew rate* 730 V/ μ s und der Strom durch die Messkapazität beträgt laut Gleichung (4.12) 730 μ A. Die Referenzspannung des ADCs wird mit 1,23 Volt festgesetzt, da dies einem typischen Spannungswert einer *Bandgap* Referenz entspricht.

Setzt man nun diesen Strom auf die halbe Referenzspannung (0,6 V) und definiert einen auflösbaren Messbereich von $\delta_M = \pm 40\%$, erhält man Werte für den maximalen und den minimalen Strom durch die Kapazität:

$$I_{FB_{nom}} = 730 \mu A \quad (5.1)$$

$$I_{max} = I_{FB_{nom}} \cdot (1 + 0,4) = 1,022 mA \quad (5.2)$$

$$I_{min} = I_{FB_{nom}} \cdot (1 - 0,4) = 438 \mu A \quad (5.3)$$

Die Wahl des Messbereiches ist sehr applikationsspezifisch und soll in etwa im Bereich des möglichen Fehlers liegen. Die nächste essentielle Größe ist die Auflösung des ADCs. Hier stellt sich die Frage, um wieviel sich die *Slew rate* ändern darf, bevor der Regler durch den DAC eingreift, um den Wert zu korrigieren. Setzt man einen Wert von 5 Bit ein, ergibt sich eine Änderung des Feedback Stromes von

$$I_R = \frac{I_{max} - I_{min}}{2^{N_{ADC}}} = 18 \mu A = \Delta I_{FB}, \quad (5.4)$$

was für diese Anwendung bereits sehr genau ist. Der hier mit I_R bezeichnete Strom ist genau jener Strom, den der A/D Umsetzer pro Stufe auflösen kann. Setzt man nun Gleichung (5.4) in Gleichung (4.12) ein, erhält man eine minimal auflösbare Änderung der *Slew rate* von 18 V/ μ s. Dieser minimal auflösbare Wert ist sehr genau und kann durch Verwendung eines A/D Umsetzers mit geringerer Auflösung, was sich auch in geringerer Komplexität und Chipfläche äußert, erreicht werden.

Tab. 5.2: Auflösung bei verschiedenen Bits und Fehlerbereichen

Bit	I_{nom} [μA]	Bereich δ_M	I_{max} [μA]	I_{min} [μA]	I_R [μA]	maxFehler [$\text{V}/\mu\text{s}$]
3	730	0,2	876	584	41,71	20,86
4	730	0,2	876	584	19,47	9,73
5	730	0,2	876	584	9,42	4,71
6	730	0,2	876	584	4,63	2,32
7	730	0,2	876	584	2,30	1,15
8	730	0,2	876	584	1,15	0,57
3	730	0,3	949	511	62,57	31,29
4	730	0,3	949	511	29,20	14,60
5	730	0,3	949	511	14,13	7,06
6	730	0,3	949	511	6,95	3,48
7	730	0,3	949	511	3,45	1,72
8	730	0,3	949	511	1,72	0,86
3	730	0,4	1022	438	83,43	41,71
4	730	0,4	1022	438	38,93	19,47
5	730	0,4	1022	438	18,84	9,42
6	730	0,4	1022	438	9,27	4,63
7	730	0,4	1022	438	4,60	2,30
8	730	0,4	1022	438	2,29	1,15
3	730	0,5	1095	365	104,29	52,14
4	730	0,5	1095	365	48,67	24,33
5	730	0,5	1095	365	23,55	11,77
6	730	0,5	1095	365	11,59	5,79
7	730	0,5	1095	365	5,75	2,87
8	730	0,5	1095	365	2,86	1,43
3	730	0,6	1168	292	125,14	62,57
4	730	0,6	1168	292	58,40	29,20
5	730	0,6	1168	292	28,26	14,13
6	730	0,6	1168	292	13,90	6,95
7	730	0,6	1168	292	6,90	3,45
8	730	0,6	1168	292	3,44	1,72
3	730	0,7	1241	219	146,00	73,00
4	730	0,7	1241	219	68,13	34,07
5	730	0,7	1241	219	32,97	16,48
6	730	0,7	1241	219	16,22	8,11
7	730	0,7	1241	219	8,05	4,02
8	730	0,7	1241	219	4,01	2,00

Tabelle 5.2 zeigt den Fehlerstrom durch die Messkapazität pro Stufe, die der aktuelle Wert vom Sollwert entfernt ist. Bemerkenswert ist hier: Will man die Genauigkeit erhöhen (kleinerer I_R Strom), muss man entweder die Auflösung des A/D Umsetzers erhöhen oder den Betrachtungsbereich einschränken. Bei Erhöhung der Auflösung wird der A/D Umsetzer, abhängig von der Architektur, entweder in der Fläche oder auch der Wandelzeit beeinflusst. Schränkt man den Betrachtungsbereich ein, muss man prinzipiell schon vor der ersten Messung wissen, in welchem Bereich sich der Wert ungefähr befindet. Verringert man beispielsweise die ADC Auflösung um 1 Bit, muss man den Betrachtungsbereich halbieren, damit die Genauigkeit unverändert bleibt.

Um nun auf einen ersten Wert für die Auflösung zu kommen, definiert man die Genauigkeit in Prozent. Will man also die *Slew rate* von $730 \text{ V}/\mu\text{s}$ auf $\pm 2,5 \%$ genau detektieren (mit der oben definierten Messkapazität von 1 pF), benötigt man bei einem Beobachtungsbereich von $\pm 40\%$ einen Strom I_R von $730 \mu\text{A} \times 2 \times 0,025 = 36,5 \mu\text{A}$ pro Schritt. Ausgehend von einem maximalen Fehler von $1/2$ Least Significant Bit (LSB) benötigt man somit laut Tabelle 5.2 4 Bit. Aus den bisherigen Erläuterungen

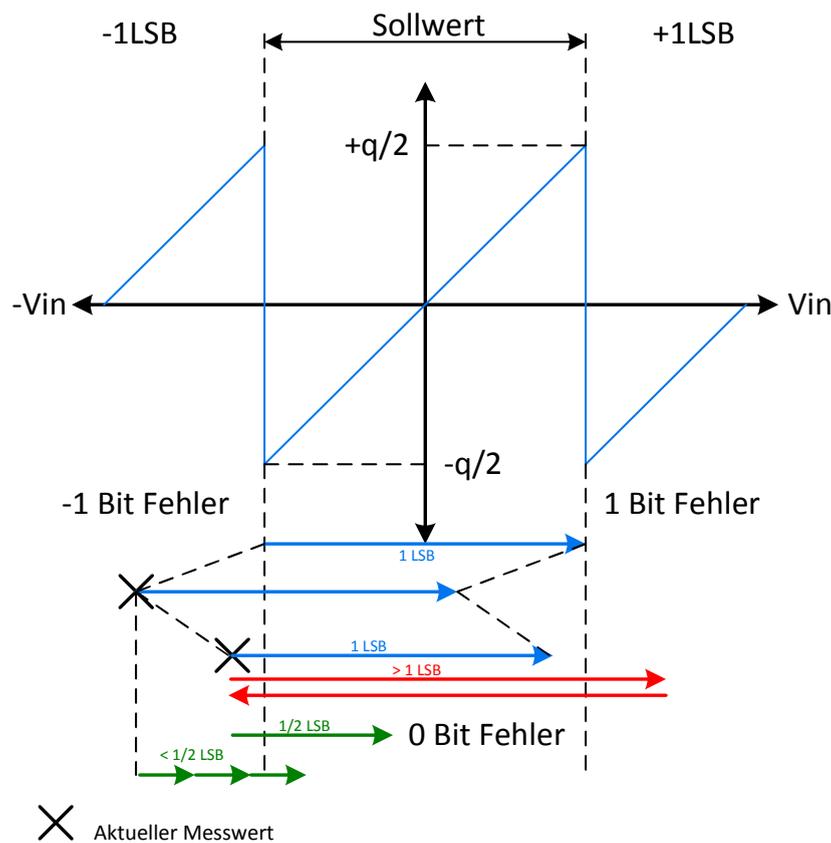


Abb. 5.6: Darstellung der minimalen Schrittweite zur Korrektur des Fehlers

ist hervorgegangen, in welchem Bereich Abweichungen erkannt werden können. Generell wird die Auflösung des Systems vom Kunden bestimmt, der festlegt, wie groß der

maximale Fehler sein darf. Nun soll berechnet werden, wie groß die entsprechende Stellgröße zur Korrektur sein muss, um etwaigen Abweichungen vom Sollwert entgegen zu wirken.

Hierfür ist vor allem das Verhalten der Strecke, welches im nächsten Abschnitt diskutiert wird, von Bedeutung.

Die Basis für die minimale Stellgröße ist die oben berechnete Auflösung des A/D Umsetzers. Prinzipiell ist es nur sinnvoll, den maximal detektierbaren Fehler zu korrigieren und nicht mehr. Dies soll vor allem Grenzyklus-Schwingungen vermeiden [Tre10].

$$\frac{1}{2^M} G_{DC} \leq \Delta V_{ADC} \quad (5.5)$$

Diesbezüglich zeigt Gleichung (5.5) die Bedingung, wobei M die Auflösung des Stellgliedes, G_{DC} die DC Verstärkung des Stellgliedes auf die Strecke und ΔV_{ADC} die Stufenspannung des A/D Umsetzers ist.

Aufgrund der endlichen Genauigkeit des ADCs beträgt der maximale Quantisierungsfehler $\pm 1/2$ LSB (*Least significant Bit*) [Kes05], das man vom Sollwert entfernt sein kann. Würde das geregelte System im Betrieb beispielsweise einen Regelfehler von -1 LSB aufweisen und der Regler würde mit einer zu groß gewählten Stellgröße diesen Fehler zu korrigieren versuchen, läge der Regelfehler nach der Korrektur oberhalb der Sollwertschranke bei $+1$ LSB, was im nächsten Schritt eine erneute Korrektur in die entgegengesetzte Richtung zur Folge hätte.

Abbildung 5.6 zeigt das Korrekturschema für Abweichungen vom Sollwert. Wählt man die minimale Stellgröße gleich 1 LSB, kommt man bei verschwindendem Regelfehler in den Sollwertbereich. Wählt man einen Wert der zuviel korrigiert, kommt man unter Umständen nie in den Zielbereich (rote Pfeile) und fällt die Wahl zu klein aus, kann man zwar mit mehreren Schritten in den Zielbereich gelangen, jedoch kann es dazu kommen, dass man bedingt durch die endliche Anzahl der Korrekturstufen (2^N Stufen), Abweichungen von mehreren LSB nicht mehr korrigieren kann.

5.8 Strecke des betrachteten Systems

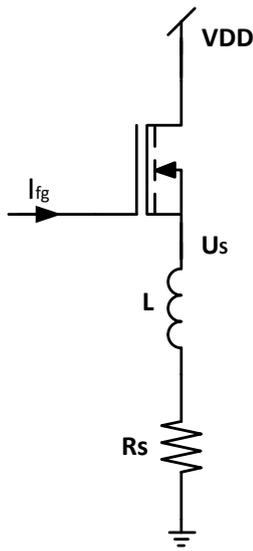


Abb. 5.7: Schematische Darstellung der Strecke

Für die Analyse der Strecke des betrachteten Systems sind nur die relevanten Komponenten zu betrachten. Diese sind der Transistor HS_1, die Zylinderspule und ein parasitärer Serienwiderstand (Abbildung 5.7). Wieder werden die relevanten parasitären Kapazitäten des Transistors in Betracht gezogen. Nun soll auf die Änderung der Source-Spannung aufgrund einer geänderten Gate-Spannung geschlossen werden. Man betrachte hierzu das dargestellte Ersatzschaltbild (Abbildung 5.8).

In diesem Ersatzschaltbild fällt sofort die vernachlässigte Gate-Drain Kapazität auf. Der Grund dafür ist, dass hierfür nur die Spannungsänderung am Source-Knoten, verursacht durch den Gate-Knoten, von Relevanz ist und aufgrund der gedachten Spannungsquelle U_{IN} eine Vereinfachung getroffen wird und das Gate ohne Verzögerung umgeladen werden kann. Um nun wieder eine Übertragungsfunktion zu erhalten, setzt

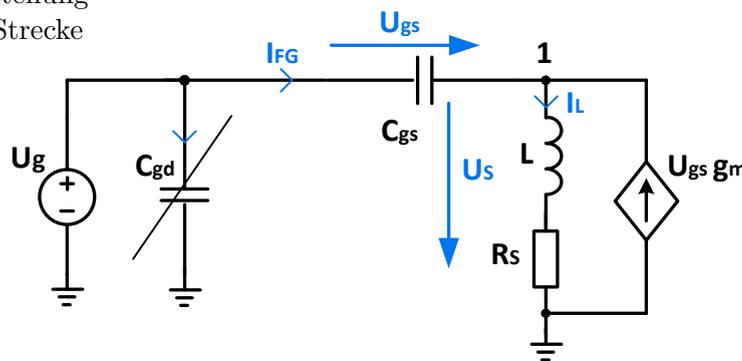


Abb. 5.8: Kleinsignal-ESB für vereinfachte Spannungsübertragung

man folgende Knotengleichungen an:

$$\begin{aligned}
 1: \quad U_{gs} C_{gs} s + g_m U_{gs} &= U_s \frac{1}{R_S + sL} \\
 U_{gs} (C_{gs} s + g_m) &= U_s \frac{1}{R_S + sL} \\
 U_{gs} &= U_s \frac{1}{(R_S + sL)(sC_{gs} + g_m)} \\
 U_g &= U_{gs} + U_s = U_s \left(\frac{1}{(R_S + sL)(sC_{gs} + g_m)} + 1 \right)
 \end{aligned}$$

Nach weiteren Schritten erhält man folgende Beziehung:

$$\frac{U_s}{U_g} = \frac{(R_S + sL)(sC_{gs} + g_m)}{1 + (R_S + sL)(sC_{gs} + g_m)} \quad (5.6)$$

Für diese Übertragungsfunktion erhält man eine Verstärkung von etwas weniger als Eins, was auch nicht verwunderlich ist, entspricht die dargestellte Konfiguration einer *Common-Drain*-Schaltung bzw. einem Source-Folger (Abbildung 5.9).

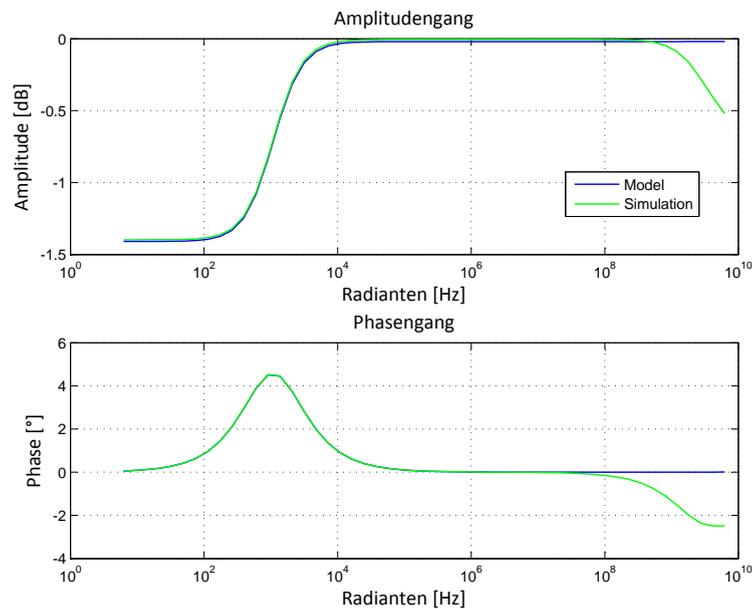


Abb. 5.9: Bode-Diagramm für das Übertragungsverhalten von U_g nach U_s

Die dominante Zeitkonstante des Systems kann mit L/R_S ermittelt werden und ist für die Stromregelung von Bedeutung.

Aus diesen Betrachtungen kann man also die Änderung der Source-Spannung aufgrund einer Änderung der Gate-Spannung als äquivalent annehmen.

Das Streckenmodell kann auch im Zeitbereich aufgestellt werden. Ziel ist ein Modell folgender Struktur mit den Grundgleichungen für die Systembeschreibung im Zustandsraum [HD04].

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (5.7)$$

$$y(t) = Cx(t) + Du(t) \quad (5.8)$$

Ausgehend vom [ESB](#) (Abbildung 5.8) geht man von den folgenden Gleichungen aus. Anzumerken ist hier, dass für $U_{gs}(t)$ einfach U_{gs} geschrieben wird, um die Übersicht zu

bewahren.

$$I_{FG} = C_{gs} \frac{dU_{gs}}{dt} \quad (5.9)$$

$$I_L = I_{FG} + U_{gs} \cdot g_m \quad (5.10)$$

$$U_s = L \frac{dI_L}{dt} + I_L \cdot R_S \quad (5.11)$$

Setzt man nun Gleichung (5.9) in Gleichung (5.10) und anschließend das Ergebnis in Gleichung (5.11) ein, erhält man folgenden Zusammenhang:

$$\begin{aligned} U_s &= L \frac{d \left(C_{gs} \frac{dU_{gs}}{dt} + U_{gs} \cdot g_m \right)}{dt} + \left(C_{gs} \frac{dU_{gs}}{dt} + U_{gs} \cdot g_m \right) \cdot R_S \\ U_s &= L \cdot C_{gs} \frac{d^2 U_{gs}}{dt^2} + L \cdot g_m \frac{dU_{gs}}{dt} + R_S \cdot C_{gs} \cdot \frac{dU_{gs}}{dt} + R_S \cdot g_m \cdot U_{gs} \\ U_s &= L \cdot C_{gs} \frac{d^2 U_{gs}}{dt^2} + (L \cdot g_m + R_S \cdot C_{gs}) \frac{dU_{gs}}{dt} + R_S \cdot g_m \cdot U_{gs} \end{aligned}$$

Mit dem Zusammenhang für die Gate-Sourcespannung

$$U_{gs}(t) = U_g(t) - U_s(t) \quad (5.12)$$

kommt man zu folgender Relation:

$$\begin{aligned} U_g - L \cdot C_{gs} \frac{d^2 U_{gs}}{dt^2} &= (L \cdot g_m + R_S \cdot C_{gs}) \frac{dU_{gs}}{dt} + R_S \cdot g_m \cdot U_{gs} + U_g - U_s \\ \frac{d^2 U_{gs}}{dt^2} &= \frac{-(L \cdot g_m + R_S \cdot C_{gs})}{L \cdot C_{gs}} \frac{dU_{gs}}{dt} - \frac{R_S \cdot g_m + 1}{L \cdot C_{gs}} \cdot U_{gs} + \frac{U_g}{L \cdot C_{gs}} \end{aligned}$$

Nun führt man den Zustandsvektor $x(t)$ ein:

$$x(t) = \begin{bmatrix} U_{gs}(t) \\ \dot{U}_{gs}(t) \end{bmatrix} \quad (5.13)$$

Mit der Schreibweise für $\dot{U}_{gs}(t)$

$$\dot{U}_{gs}(t) = \frac{dU_{gs}(t)}{dt} \quad (5.14)$$

kommt man schließlich zur Systembeschreibung im Zustandsraum :

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -\frac{R_S \cdot g_m + 1}{L \cdot C_{gs}} & -\frac{L \cdot g_m + R_S \cdot C_{gs}}{L \cdot C_{gs}} \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ \frac{1}{L \cdot C_{gs}} \end{bmatrix} U_g(t) \quad (5.15)$$

5.8.1 Modell mit 3 Zustandsvariablen

Das obige Modell 2. Ordnung beschreibt die Beziehung der Spannungen zwischen Gate- und Source-Knoten. Wie jedoch die Gate-Spannung zustande kommt, wird dort vorausgesetzt. Will man ein Modell mit Stromeingang haben, erweitert man Abbildung 5.8 um eine Gate-Kapazität C_g , welche auch die Gate-Drain Kapazität enthält.

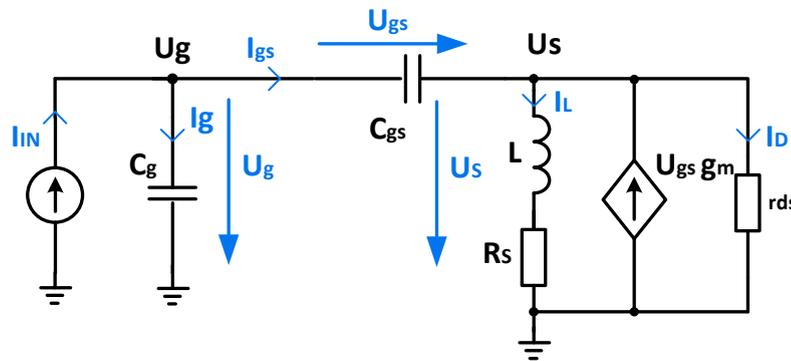


Abb. 5.10: Kleinsignal-ESB für Modell mit Stromeingang

Der Zustandsvektor, der die Zustandsvariablen enthält, sieht nun folgendermaßen aus.

$$x(t) = \begin{bmatrix} U_g(t) \\ U_s(t) \\ I_L(t) \end{bmatrix} \quad (5.16)$$

Dazu ist noch zu sagen, dass die Wahl der Zustandsvariablen, wie in [HD04] beschrieben, nicht eindeutig ist und man die Größen mit vorrangigem Interesse wählt. Nun geht man wieder von den Ansatzgleichungen aus:

$$I_{in} = I_g + I_{gs} \quad (5.17)$$

$$I_{gs} + U_{gs} \cdot g_m = I_L + U_s \cdot \frac{1}{r_{ds}} \quad (5.18)$$

$$U_s = L \frac{dI_L}{dt} + I_L \cdot R_S \quad (5.19)$$

Gleichung (5.19) entspricht genau Gleichung (5.11). Nun setzt man folgende Relation an:

$$I_{in} = I_{gs} + I_g \quad (5.20)$$

$$= C_{gs} \frac{dU_{gs}}{dt} + C_g \frac{dU_g}{dt} \quad (5.21)$$

$$= C_{gs} \frac{dU_g}{dt} - C_{gs} \frac{dU_s}{dt} + C_g \frac{dU_g}{dt} \quad (5.22)$$

$$\Rightarrow \frac{dU_g}{dt} = \frac{I_{in} + C_{gs} \frac{dU_s}{dt}}{C_{gs} + C_g} \quad (5.23)$$

Nun formt man Gleichung (5.18) nach I_{gs} um und setzt den resultierenden Term in Gleichung (5.17) ein.

$$I_L = I_{in} - C_g \frac{dU_g}{dt} + U_g \cdot g_m - U_s \cdot g_m - U_s \cdot \frac{1}{r_{ds}} \quad (5.24)$$

$$\frac{dU_g}{dt} = -\frac{1}{C_g} I_L + \frac{g_m}{C_g} U_g - \frac{g_m + \frac{1}{r_{ds}}}{C_g} U_s + \frac{1}{C_g} I_{in} \quad (5.25)$$

Für die Bestimmung der Relation für die Source-Spannung setzt man Gleichung (5.23) in Gleichung (5.25) ein.

$$\frac{I_{in} + C_{gs} \frac{dU_s}{dt}}{C_{gs} + C_g} = -\frac{1}{C_g} I_L + \frac{g_m}{C_g} U_g - \frac{g_m + \frac{1}{r_{ds}}}{C_g} U_s + \frac{1}{C_g} I_{in} \quad (5.26)$$

$$\frac{dU_s}{dt} = \acute{C}_g g_m U_g - \acute{C}_g \left(g_m + \frac{1}{r_{ds}} \right) U_s - \acute{C}_g I_L + \frac{1}{C_g} I_{in} \quad (5.27)$$

Mit der Bezeichnung

$$\acute{C}_g = \frac{C_{gs} + C_g}{C_{gs} \cdot C_g} \quad (5.28)$$

erhält man aus den Gleichungen (5.25), (5.27) und der bleibenden Relation (5.11) schließlich die Zustandsraumbeschreibung.

$$\dot{x}(t) = \begin{bmatrix} \frac{g_m}{C_g} & -\frac{g_m + \frac{1}{r_{ds}}}{C_g} & -\frac{1}{C_g} \\ \acute{C}_g g_m & -\acute{C}_g \left(g_m + \frac{1}{r_{ds}} \right) & -\acute{C}_g \\ 0 & \frac{1}{L} & -\frac{R_S}{L} \end{bmatrix} x(t) + \begin{bmatrix} \frac{1}{C_g} \\ \frac{1}{C_g} \\ 0 \end{bmatrix} I_{in}(t) \quad (5.29)$$

Mit einer geeigneten Wahl des Ausgangsvektors mit

$$C = [0 \quad 1 \quad 0] x(t) \quad (5.30)$$

erhält man ein LZI mit I_{in} als Eingangsgröße und U_s als Ausgangsgröße.

Nach dem Einsetzen der Werte für einen Arbeitspunkt und der Berechnung der Übertragungsfunktion kann das Bode-Diagramm gezeichnet werden.

$$G(s) = \frac{U_s}{I_{in}} = \frac{r_{ds}(s^2LC_{gs} + sLg_m + sC_{gs}R_s + g_mR_s)}{s[r_{ds}C_{gd}(s^2LC_{gs} + sLg_m + sC_{gs}R_s + g_mR_s) + (C_{gd} + C_{gs})(sL + r_{ds} + R_s)]} \quad (5.31)$$

Bemerkenswert ist wiederum der Pol bei $s = 0$ und das dadurch integrierende Verhalten, welches zur Strukturinstabilität führt.

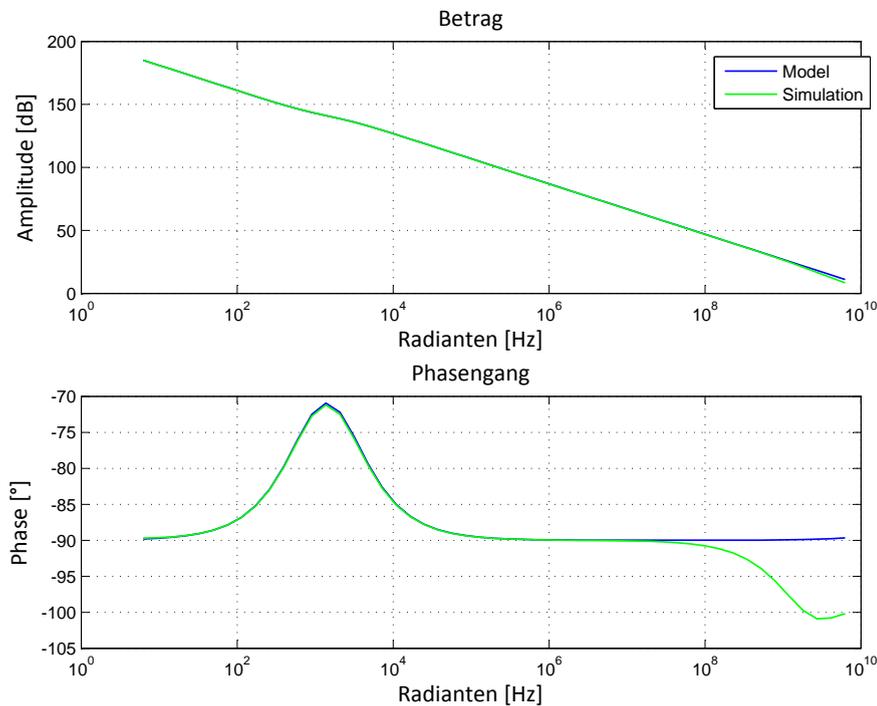


Abb. 5.11: Bode-Diagramm des Modells sowie einer Simulation im Vergleich

Abbildung 5.11 zeigt das aufgestellte Modell im Vergleich mit einer AC Simulation der Strecke. Man sieht, das Modell approximiert die Strecke bis auf Einflüsse bei hohen Frequenzen sehr gut.

Das oben dargestellte Bode-Diagramm zeigt, wie bereits erwähnt, die Übertragungsfunktion von I_{in} nach U_s . Da jedoch die Ausgangsgröße U_s nicht direkt verwendet werden kann, da der verwendete Messumformer C_{mess} die zeitliche Ableitung, nämlich

den Rückkoppelstrom liefert, muss um das Übertragungsverhalten von I_{in} nach I_{FB} zu erhalten, obige Übertragungsfunktion mit dem Term

$$Y_{mess}(s) = C_{mess} \cdot s \quad (5.32)$$

multipliziert werden. So fällt der Amplitudengang dann nicht mehr ab, sondern bleibt ungefähr konstant, was einem Proportionalglied mit Verzögerung entspricht. Abbildung 5.12 zeigt das entsprechende Bode-Diagramm. Der in Abbildung 5.12 dargestellte

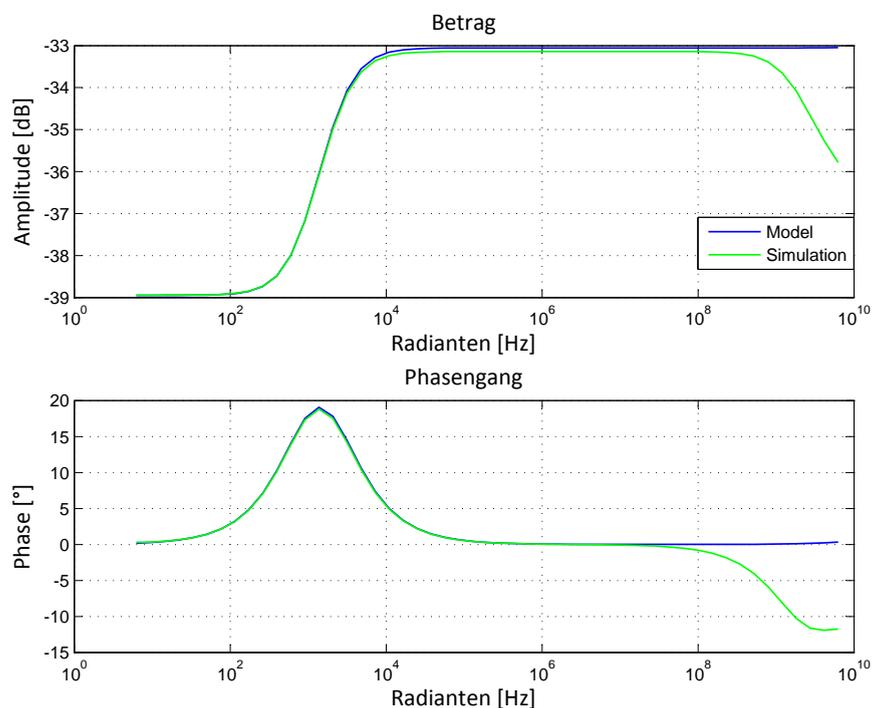


Abb. 5.12: Bode-Diagramm des Übertragungsverhaltens der Ströme

Proportionalwert der Amplitude resultiert direkt aus der Beziehung (5.33) mit dem Verhältnis aus Messkapazität zu Gate-Kapazität.

Beachtenswert ist vor allem, dass durch besagte Multiplikation die Strecke sprungfähig wird. Durch die Kürzung von s im Nenner erhält man ein Polynom 2. Grades mit ausschließlich positiven Koeffizienten. Dies ist in diesem Fall eine hinreichende Bedingung für ein Hurwitz-Polynom, wodurch die Strecke, unabhängig von den gewählten Koeffizienten (C_{gd} , C_{gs} , g_m , R_s , L), welche sicher immer positiv sind, asymptotisch stabil ist [HD04].

Dies ist insofern wichtig, da es somit keine instabilen Pole gibt, die durch einen Regler kompensiert werden müssen.

5.8.2 Berechnung des Stellwertes

Mit Gleichung (5.4) und (4.12) wurde bereits die minimal auflösbare Änderung hergeleitet. Transferiert man nun noch die Änderung der Source-Spannung oder $\frac{dU_s}{dt}$ mit der oben getroffenen Vereinfachung für das *Sourcefolger*-Modell zurück auf die Gate-Spannung und multipliziert laut (4.12) mit der Gate-Kapazität, erhält man den notwendigen LSB-Strom, um die Änderung korrigieren zu können.

$$I_{LSB} = \frac{C_{Gate} (I_{max} - I_{min})}{C_{mess} 2^{N_{ADC}}} = \frac{C_{Gate}}{C_{mess}} I_R = \frac{1}{K_F} I_R \quad (5.33)$$

C_{Gate} ist in diesem Zusammenhang die Gate-Kapazität, die während des Umschaltvorgangs wirkt. Aus Gleichung (5.33) ist auch ersichtlich, dass der wichtigste Parameter, um einen Leistungs-Transistor in obiger Konfiguration während des Schaltvorgangs beeinflussen zu können, die Gate-Kapazität ist.

Um nun auch Relation (5.5) einzuhalten, muss umgelegt auf diese Anwendung folgende Ungleichung erfüllt sein, um Grenzzyklus-Schwingungen zu vermeiden.

$$I_{LSB} K_F \leq I_R \quad (5.34)$$

Wobei K_F der Proportionalfaktor der Strecke ist, welcher aus Abbildung 5.12 für die betrachtete Frequenz mit -33 dB abgelesen werden kann. Dies entspricht genau dem umgekehrten Verhältnis der Kapazitäten aus (5.33). Um Ungleichung (5.34) zu erfüllen, kann in Gleichung (5.33) ein Quotienten hinzugefügt werden.

$$I_{LSB} \leq \frac{1}{K_F} I_R \frac{2^{N_{ADC}}}{2^{N_{DAC}}} \quad (5.35)$$

Hier bezeichnet N_{ADC} die Auflösung des A/D Umsetzers und N_{DAC} die Auflösung des Strom-DACs. Die Ungleichung (5.35) ist offensichtlich genau dann erfüllt, wenn die Auflösung des DACs größer gewählt wird als jene des ADCs.

Um solche Grenzzyklus-Schwingungen in linearen digitalen Regelkreisen zu vermeiden, müssen zwei Bedingungen eingehalten werden:

$$\text{Auflösung(DAC)} > \text{Auflösung(ADC)} \quad (5.36)$$

Für das geregelte System ist es von Wichtigkeit, dass unter allen Umständen ein DAC-Pegel existiert, der den Ausgang der Strecke mit Sicherheit in den Null-Fehler-Bereich bringt. Dies kann sichergestellt werden, indem die Auflösung des DACs größer gewählt wird als jene des A/D Umsetzers.

$$1 \geq K_I > 0 \quad (5.37)$$

Diese Bedingung bedeutet, dass sich bei Ungenauigkeiten im Vorwärtszweig der Regelschleife, sollte kein integraler Anteil vorhanden sein, ein bleibender Regelfehler einstellt, welcher ebenfalls zu besagten Schwingungen führen kann.

Zum ersten Punkt zeigt Abbildung 5.13 das Ausgangssignal eines digitalen Regelkreises. Bei dem oberen Diagramm wurde die erste Bedingung verletzt. Das Signal pendelt im eingeschwungenen Zeitbereich zwischen zwei Bereichen des A/D Umsetzers. Im unteren Diagramm wurde die Auflösung des DACs höher gewählt als die des ADCs. So schwingt das Signal ebenfalls, jedoch mit einer wesentlich geringeren Abweichung, welche es nicht dazu bringt, den Null-Fehler-Bereich des ADCs zu verlassen [Pet03].

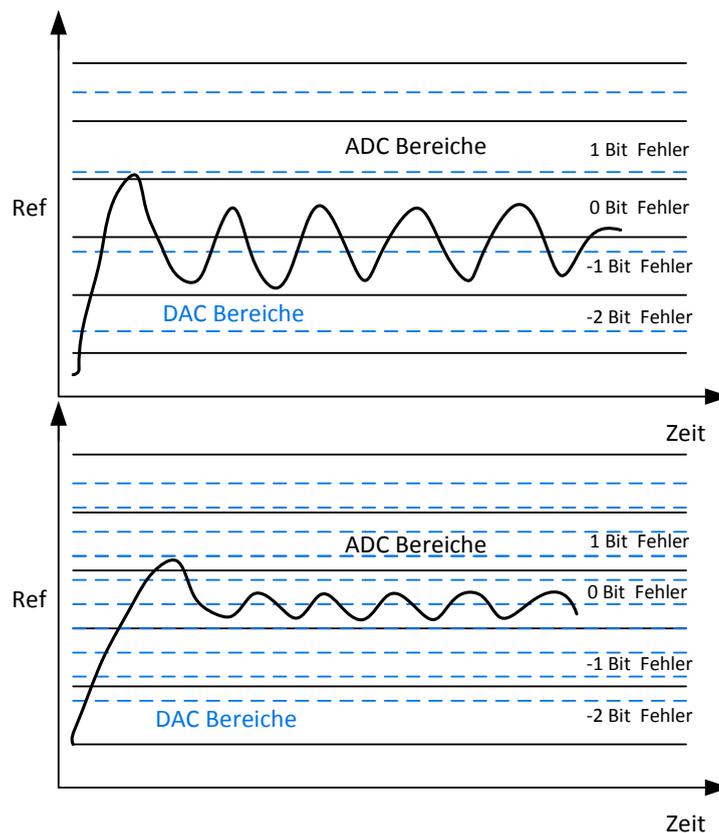


Abb. 5.13: Gegenüberstellung der Auflösungen von ADC und DAC in Hinblick auf Grenzyklus-Schwingungen

Aufgrund der sich ändernden Gate Kapazität, bezogen auf den Arbeitspunkt des Transistors, ergeben sich auch einige Probleme in Bezug auf die Stellgröße. Da sich die Kapazität während des Einschaltvorganges ändert und somit auch ein Koeffizient der Übertragungsfunktion verändert wird, stellt das ein zeitvariantes System dar.

5.8.3 Modell der Zylinderspule

Aus elektronischer Sicht besteht das Magnetventil aus einer Spule, deren Induktivität sich aus folgender Gleichung ergibt.

$$L_{max} = \frac{\mu N^2 A_L}{l} \quad (5.38)$$

Der genaue Wert der Induktivität hängt von der Position des Tauchkolbens ab und wird folgendermaßen modelliert [DD03]:

$$L(x) = \frac{L_{max}}{1 + K_L x} \quad (5.39)$$

$\mu = \mu_0 \mu_r$	Permeabilität
N	Windungszahl
A_L	Querschnittsfläche einer Windung
l	Länge der Spule
x	Position des Tauchkerns
K_L	Induktionskoeffizient

Da nun die Größen an den Schnittstellen zwischen den analogen und den digitalen Komponenten spezifiziert sind, soll als nächster Schritt der digitale Regler bzw. die Art der Regelung genauer betrachtet werden.

5.9 Regelalgorithmus

Dem aus der Definition des Systems bekannten Regelkreis liegt ein Regelalgorithmus zugrunde, der Abweichungen vom Sollwert aber auch Störungen kompensieren soll.

Ausgehend von der rein analogen Systembetrachtung, wie sie beispielsweise in Kapitel 4 verwendet wurde, sollen nun die digitalen Möglichkeiten einer Regelung erörtert werden.

Die Wahl der Abtastzeit T_A des Systems wird in der vorliegenden Applikation in einem Bereich, der sehr viel kleiner als die kritische Zeitkonstante der Strecke ist, liegen, da die Taktfrequenz durch die Dauer der Flanken nach unten hin begrenzt ist. Die Regelung kann als quasikontinuierlich betrachtet werden. Ist dies der Fall, kann man den Regelkreis als analog ansehen und auch den Entwurf mit analogen Verfahren durchführen [RZ08]. Die einzige Einschränkung, die man beachten muss, ist, dass für Berechnungen,

wie beispielsweise der Regeldifferenz oder der Stellkoeffizienten, genug Taktzyklen zur Verfügung stehen, um die Funktionalität zu gewährleisten.

Eine direkte Regelung bedeutet ein quasi-kontinuierliches System ähnlich der analogen Variante im vorherigen Kapitel. Der aktuelle Regelfehler wird also auf Basis des vorherigen Steigungswertes der Flanke berechnet und ausgegeben Abbildung 5.14(a).

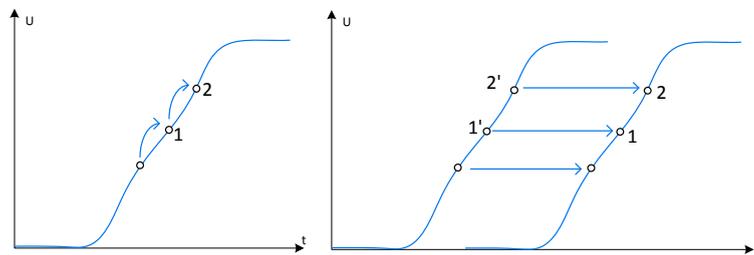


Abb. 5.14: Basis der Regelung, (a) direkt, (b) nachlaufend

In Abbildung 5.14(b) wird der Korrekturwert für den aktuellen Zeitpunkt aus demselben Zeitpunkt der vorherigen Flanke gebildet. Der Unterschied zwischen den beiden Methoden liegt in der Systembetrachtung. Wie oben bereits erwähnt, handelt es sich um ein zeitvariantes System bezüglich der parasitären MOS Kapazitäten bzw. auch der Parameter r_{ds} und g_m . Das offensichtliche Problem für die Methode (a) besteht also darin, dass der Korrekturwert für den Zeitpunkt 2 aufgrund der Kapazität im Punkt 1 berechnet wird, also wird sich auch die Strecke im Punkt 2 aufgrund der Einwirkung anders verhalten als im Punkt 1.

Methode (b) behebt dieses Problem durch Speicherung der Werte der vorherigen Flanke und so wird der Korrekturwert für den richtig gemessenen Ausgangszeitpunkt ermittelt. Mit diesem Ausgangszeitpunkt kann man wieder von einem LZI-System ausgehen.

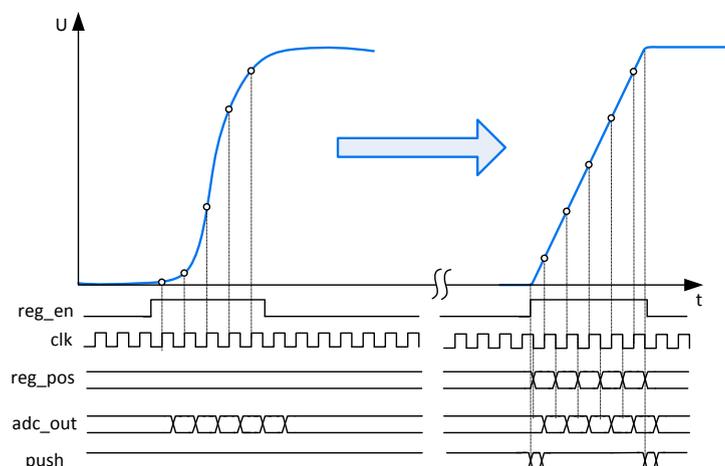


Abb. 5.15: Ablauf der nachlaufenden Regelung

Abbildung 5.15 zeigt den zeitlichen Ablauf der nachlaufenden Regelung, für die steigende Flanke. Im unteren Teil sind einige digitale Signale beschrieben, die für die Implementierung benötigt werden. Das Signal `reg_en` aktiviert den Regler. Am Beginn der Flanke wird mit dem Signal `push` ein höherer Strom auf das Gate geleitet, um die Schwellenspannung zu erreichen und längere Verzögerungszeiten zu vermeiden. Während der Flanke werden zuerst die Registerwerte in `reg_pos` an den DAC ausgegeben, und einen halben Takt später mit dem ADC gewandelt und in `adc_out` geschrieben. Auch im letzten Teil der Flanke sorgt das Register `push` für einen erhöhten Stromwert, um möglichst schnell aus dem Millerplateau heraus zu kommen.

Ein digitaler Regelalgorithmus berechnet zwischen den Flanken mit dem Eingangswert `adc_out` die Ausgangswerte für `reg_pos`.

Da, wie oben gezeigt, die Strecke als P-Glied vorliegt, scheint es einfach, eine P-Regleinrichtung zum Regeln der Flanke zu verwenden, allerdings wird bei solch einem Regler nur eine Ausgangsgröße gebildet, wenn eine nicht verschwindende Regeldifferenz vorliegt. So würde der Reglerausgang bei optimaler Flanke null werden, was auch die Flanke wieder verändern würde. Um auch den stationären Regelfehler zu beseitigen, fällt die Wahl auf einen Proportional-Integralen (PI) Regler, bei dem nacheinander folgende Schritte ausgeführt werden:

1. Wandlung des Eingangstromes I_{FB} während der Flanke für sämtliche betrachtete Zeitpunkte
2. Berechnung der Regeldifferenz
3. Berechnung der Koeffizienten für den Gate-Treiber durch Multiplikation und Summation

Diese nachführende Regelung bietet auch einen weiteren Vorteil für das zeitliche Verhalten. Bei der direkten Methode in Abbildung 5.14(a) müsste nach dem Einlesen des ersten Wertes zu Beginn der Flanke bereits der gesamte obere Ablauf mit der Berechnung der Regeldifferenz beginnend erfolgen, um den nachfolgenden Ausgangswert vor dem nächsten Einlesen ausgeben zu können. Dies würde die Anforderungen an die Abtastrate erhöhen, da zwischen zwei Messpunkten die komplette Berechnung sowie die Ausgabe erfolgen muss. Bei der Methode (b) wird während der Flanke nur gemessen sowie die neuen Werte ausgegeben. Die Berechnung dieser Werte findet während der stationären Pegel statt. So richtet sich die Abtastrate nur nach der Anzahl der Messpunkte. Als nächsten Schritt betrachtet man nun das Übertragungsverhalten der einzelnen Komponenten. Die Relation für den Messumformer ist mit Gleichung (4.12) bereits bekannt. Die nächste Komponente bildet der A/D Umsetzer, welcher aus systemtheoretischer Sicht als Kombination eines Verstärkungsfaktors mit einer Totzeit

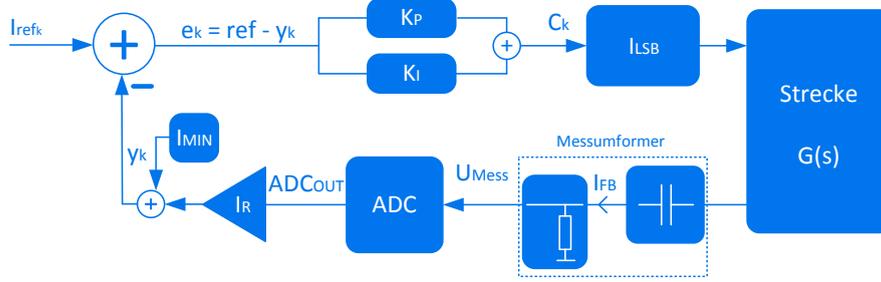


Abb. 5.16: Mathematische Beschreibung des Standardregelkreises

(Wandelzeit) beschrieben werden kann [Tre10].

$$ADC_{out} = \frac{1}{\Delta V_{adc}} e^{-s\Delta t_{wandel}} \quad (5.40)$$

$$\Delta V_{adc} = \frac{V_{ref}}{2^{N_{ADC}}} \quad (5.41)$$

Mit N_{ADC} als Auflösung des A/D Umsetzers. Der benötigte Messwiderstand, um den Strom in eine Messspannung umzuwandeln ist gegeben durch:

$$R_{mess} = \frac{V_{ref}}{I_{max} - I_{min}} \quad (5.42)$$

Befindet sich der Eingangswert zwischen den definierten Schranken (5.2) und (5.3) ergibt sich der Ausgangswert aus folgenden Ansätzen:

$$V_{mess} = I_{FB} \cdot \frac{V_{ref}}{I_{max} - I_{min}} - \frac{I_{max} V_{ref}}{I_{max} - I_{min}} + V_{ref} \quad (5.43)$$

$$= V_{ref} \left(\frac{I_{FB}}{I_{max} - I_{min}} - \frac{I_{max}}{I_{max} - I_{min}} + 1 \right) \quad (5.44)$$

Die Messeinrichtung liefert dem ADC eine Spannung durch R_{mess} proportional I_{FB} um von I_{min} bis I_{max} aufzulösen:

$$ADC_{out} = \frac{1}{\Delta V_{adc}} \cdot V_{ref} \left(\frac{I_{FB}}{I_{max} - I_{min}} - \frac{I_{max}}{I_{max} - I_{min}} + 1 \right) \quad (5.45)$$

$$ADC_{out} = \frac{2^{N_{ADC}}}{V_{ref}} V_{ref} \left(\frac{I_{FB}}{I_{max} - I_{min}} - \frac{I_{max}}{I_{max} - I_{min}} + 1 \right) \quad (5.46)$$

$$= 2^{N_{ADC}} \left(\frac{I_{FB}}{I_{max} - I_{min}} - \frac{I_{max} - I_{max} + I_{min}}{I_{max} - I_{min}} \right) \quad (5.47)$$

$$= \frac{I_{FB}}{I_R} - \frac{I_{min}}{I_R} \quad (5.48)$$

$$ADC_{out} = \frac{I - I_{min}}{I_R} = \left[\underbrace{\frac{1}{I_R}}_{K_{ADC}} I - \frac{I_{min}}{I_R} \right] \quad (5.49)$$

Mit K_{ADC} wird der Verstärkungsfaktor des Analog/Digital Umsetzers bezeichnet. Um nun den digitalisierten Wert I_{FB_k} für I_{FB} zu erhalten, müsste man Gleichung (5.49) mit I_R multiplizieren.

$$I_{FB_k} = ADC_{out} \cdot I_R + I_{min} \quad (5.50)$$

Nachfolgend geschieht die Berechnung des Regelfehlers:

$$e(t) = ref - y(t) \quad (5.51)$$

Für die nachfolgenden Ausführungen soll davon ausgegangen werden, dass aufgrund der nachlaufenden Regelung von Messpunkt zu Messpunkt von einem linearen zeitinvarianten System ausgegangen werden kann.

Der zeitkontinuierliche PI-Regelalgorithmus kann mit folgender Relation beschrieben werden [RZ08] :

$$c(t) = \underbrace{K_p e(t)}_{y_P(t)} + K_I \underbrace{\int e(t) dt}_{y_I(t)} \quad (5.52)$$

Für die Diskretisierung von Gleichung (5.52) werden der Ausgangswert $y(t)$ sowie der Regelfehler $e(t)$ ausschließlich zu diskreten Zeitabständen der Abtastperiode T_A betrachtet. Das bedeutet für t wird $k \cdot T_A$ eingesetzt, wobei k ganzzahlig ist und für die Zeitspanne dt wird T_A eingesetzt. Aus Gleichung (5.51) wird der zeitdiskrete Regelfehler e_k .

$$e_k = ref - y_k \quad (5.53)$$

Abbildung 5.17 veranschaulicht die diskrete Integration durch Approximieren mit der sogenannten Rechtecknäherung. Dies geschieht durch Aufsummieren der Elementarflächen, beginnend mit der linken Intervallgrenze. Für Gleichung (5.52) ergibt sich mit Gleichung (5.53):

$$c_k = K_p \cdot e_k + K_I \cdot T_A \sum_{i=0}^{k-1} e_i \quad (5.54)$$

Der Regelalgorithmus (5.54) wird in der Literatur [RZ08] auch *Stellungsalgorithmus* genannt und ist in vorliegender Form für die direkte Implementierung nur für Systeme mit großen Zeitkonstanten geeignet, da die Summe für jeden Ausgangswert neu berechnet werden muss.

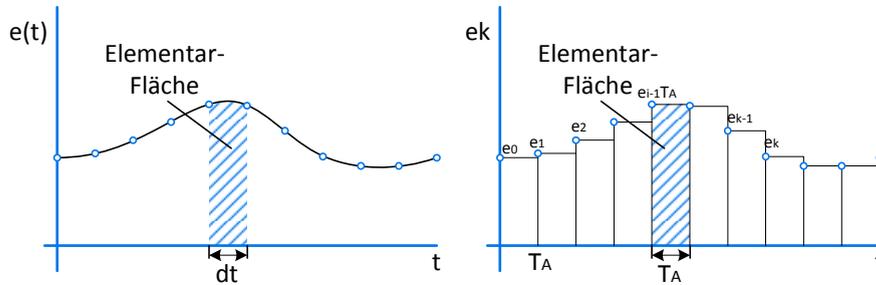


Abb. 5.17: Approximation der Integration mit der Rechteckregel

Für die vorliegende Anwendung ist der *Geschwindigkeitsalgorithmus* besser geeignet, welcher nachfolgend rekursiv aus (5.54) berechnet wird. Ziel ist es, die aktuelle Ausgangsgröße mit Hilfe der Differenz zum vorangegangenen Wert zu berechnen.

$$c_k = c_{k-1} + \Delta c_k \quad (5.55)$$

Gleichung (5.54) kann mit Einführung von T_N , der sogenannten Nachstellzeit [RZ08], folgendermaßen geschrieben werden:

$$c_k = K_p \cdot e_k + \frac{K_p}{T_N} \cdot T_A \sum_{i=0}^{k-1} e_i \quad (5.56)$$

Setzt man nun für k , $(k-1)$ ein und subtrahiert nachfolgend Gleichung (5.57) von Gleichung (5.56), erhält man genau den für (5.55) notwendigen Differenzterm:

$$c_{k-1} = K_p \cdot e_{k-1} + \frac{K_p}{T_N} \cdot T_A \sum_{i=0}^{k-2} e_i \quad (5.57)$$

$$\Delta c_k = c_k - c_{k-1} = K_p \cdot (e_k - e_{k-1}) + K_p \frac{T_A}{T_N} \left(\underbrace{\sum_{i=0}^{k-1} e_i - \sum_{i=0}^{k-2} e_i}_{e_{k-1}} \right) \quad (5.58)$$

Jetzt bleibt nur noch Gleichung (5.58) in Gleichung (5.55) einzusetzen, um den *Geschwindigkeitsalgorithmus* zu erhalten:

$$c_k = c_{k-1} + K_p (e_k - e_{k-1}) + K_p \frac{T_A}{T_N} e_{k-1} \quad (5.59)$$

Die Berechnung der Stellgröße ist eine Multiplikation mit I_{LSB} .

$$u_k = c_k \cdot I_{LSB} \quad (5.60)$$

Um nun noch aus obiger Differenzgleichung die Übertragungsfunktion zu erhalten, setzt man die Gleichung wie folgt an:

$$\begin{aligned}
 c(z) &= c(z)z^{-1} + K_p(e(z) - e(z)z^{-1}) + K_p\frac{T_A}{T_N}e(z)z^{-1} \\
 c(z) &= c(z)z^{-1} + K_p e(z) - K_p e(z)z^{-1} + K_p\frac{T_A}{T_N}e(z)z^{-1} \\
 c(z)(1 - z^{-1}) &= e(z)\left(K_p - K_p z^{-1} + K_p\frac{T_A}{T_N}z^{-1}\right) \\
 \Rightarrow \frac{c(z)}{e(z)} &= \frac{K_p - K_p z^{-1} + K_p\frac{T_A}{T_N}z^{-1}}{1 - z^{-1}} \\
 G_d(z) &= K_p \frac{z - 1 + \frac{T_A}{T_N}}{z - 1} = K_p + \frac{K_p T_A}{(z - 1)T_N} \tag{5.61}
 \end{aligned}$$

5.10 Das Stellglied

Das Stellglied in einem Regelkreis hat die Aufgabe, den Ausgangswert des Reglers entweder direkt oder nach Umformungen, der Strecke zuzuführen. Bei einem digitalen Regelkreis ist dieser Ausgangswert eine Zahl bzw. eine Bitkombination, die in eine analoge Größe umgewandelt werden muss, bevor sie der Strecke zugeführt werden kann. Wie oben bereits erläutert wurde, wird die Spannung zwischen Gate und Source mit Stromquellen gesteuert. Diese Stromquellen werden mit Schaltern zum Gate hin verbunden und so kann der Strom durch Summation mehrerer Stromquellen konfiguriert werden.

Der Ablauf der Regelung läuft in groben Schritten so ab: Es wird die Flankensteilheit gemessen, ein Fehler berechnet und aufgrund des Korrekturfaktors, den der Regelalgorithmus liefert, ein Strom zur Korrektur auf das Gate gespeist. Eine essentielle Voraussetzung, welche in dieser Beschreibung wahrscheinlich nicht sofort erkannt wird, ist, dass bereits eine Flanke am Ausgang vorhanden sein muss, um die Regelung auch in Gang zu bringen. Das bedeutet, die Strecke muss zuerst in einen Übergangszustand gebracht werden, in welchem auch die oben beschriebene Modelle gültig sind, bevor die Regelung überhaupt arbeiten kann.

Hierzu sollen nun die parasitären Kapazitäten des Transistors genauer betrachtet werden. Abbildung 5.18 zeigt typische Verläufe dieser Kapazitäten in Abhängigkeit von der Gate-Source Spannung. Die Gate-Kapazität nimmt ab einer positiven Gate-Source Spannung zu, wobei die Gate-Drain Kapazität ab der Schwellenspannung abnimmt.

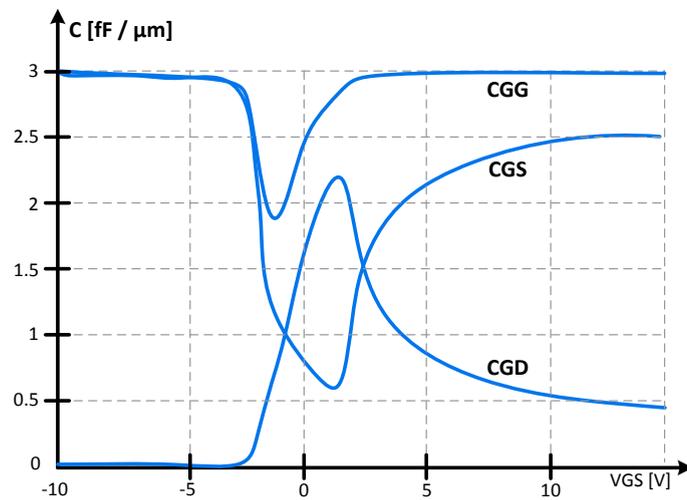


Abb. 5.18: Kapazitäten des DMOS Transistors [Sti94]

Um den DMOS-Transistor in den Schaltzustand zu bringen, muss zunächst eine bestimmte Ladung auf das Gate fließen, um die Einsatzspannung zu erreichen. Danach beginnt die Phase, in der die Regelung aktiv ist, wobei zusätzlich zu einem konstanten Strom der Regelstrom dazu beiträgt, den gewünschten Zielwert zu erreichen.

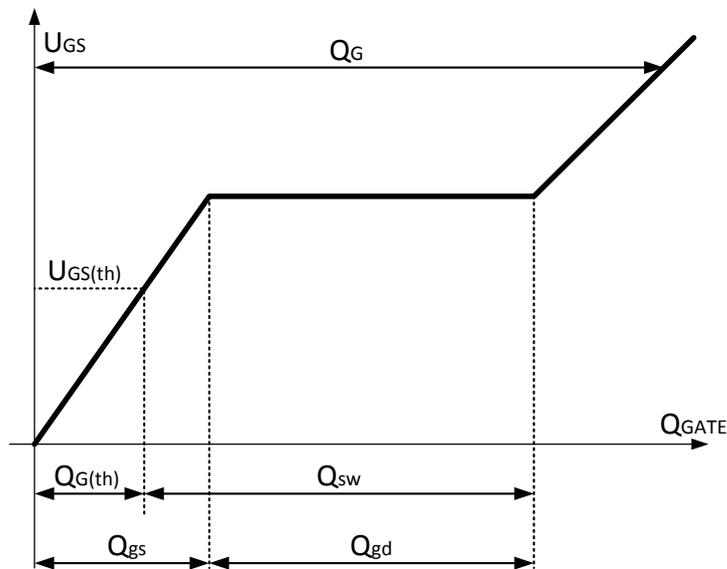


Abb. 5.19: Typisches Gate-Ladungs-Diagramm für externe MOSFETs

Abbildung 5.19 zeigt ein Gate-Ladungs-Diagramm [Inf08], welches in Datenblättern für diskrete MOSFETs zur Angabe diverser Ladungsmengen verwendet wird. Den Strom zum Erreichen der Schwellenspannung kann man aus der Ladung $Q_G(th)$ und der geforderten Zeit bestimmen. In ähnlicher Weise kann der Strom zum Entladen des Gates vor dem Ausschaltvorgang bestimmt werden.

Diese Ladung soll in weiterer Folge mit $Q_G^-(th)$ bezeichnet werden und ist mit Übereinstimmung von Abbildung 5.19 folgend definiert:

$$Q_G^-(th) = Q_G - Q_{sw} - Q_G(th) \quad (5.62)$$

Sobald also das Signal zum Umschalten von der Logik aktiv wird, fließt für eine gewisse Zeit der Strom I_{push} , um die Gate-Kapazität vorzuladen und die Einsatzspannung zu erreichen. Würde man lediglich den Transferstrom Strom I_{slope} dazu verwenden, ergäben sich längere Verzögerungszeiten, die im Rahmen spezifizierter Schaltzeiten nicht akzeptabel sind. Danach wird zusätzlich zu dem konstanten Strom I_{slope} , welcher beinahe nötig ist den Schaltvorgang in der vorgegebenen Zeit durchzuführen, der Korrekturstrom, bestimmt durch die Ausgangskoeffizienten des Reglers, addiert und die Flanke so geregelt. Abbildung 5.20 zeigt den schematischen Aufbau des DACs, welcher schal-

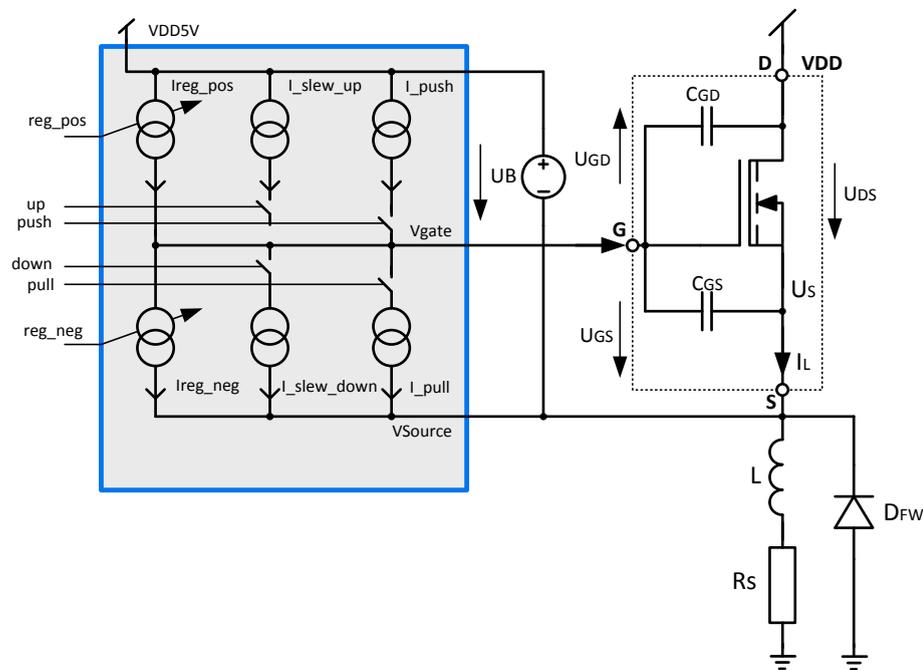


Abb. 5.20: Digital/Analog Strom Wandler (Gate-Treiber)

tungstechnisch natürlich etwas ausgereifter ausfallen wird, da die Gate-Source Spannung wegen der hohen Schaltspannung zusätzlich einen Offsetanteil enthalten muss, um den DMOS-Transistor ansteuern zu können.

Man muss also vorab Informationen über die Geometrie des Leistungstransistors, speziell über die Dimension der Eingangskapazität, besitzen, um das System für die Regelung auszurichten.

Um den Schaltvorgang einigermaßen genau durchführen zu können, werden folgende

Ströme auf das Gate gespeist.

$$I_{Push} = \frac{Q_G(th)}{T_A} \quad (5.63)$$

$$I_{Pull} = \frac{Q_G^-(th)}{T_A} \quad (5.64)$$

$$I_{slope} = C_{Gate} \frac{dU}{dt} \kappa \quad (5.65)$$

Dabei dienen I_{Push} und I_{Pull} dazu, kurz vor der jeweiligen Schaltflanke, einen erhöhten Strom zu liefern, um den Schaltvorgang ohne längere Verzögerungszeiten zu initiieren. κ ist eine Konstante zwischen 0 und 1 und bestimmt den Anteil an Konstantstrom während der Regelphase, welcher nicht verändert wird. Aus Simulationen ergibt sich ein guter Wert zwischen 0,75 und 0,9. T_A ist die Diskretisierungszeit des Systems, also jene Zeit, die ein Korrekturwert am Gate anliegt. So muss am Beginn des Schaltvorgangs für genau eine Taktperiode der Strom I_{Push} auf das Gate fließen, um die Einsatzspannung des Schalttransistors zu erreichen. Dasselbe gilt für den Beginn der Ausschaltphase: Hier muss I_{Pull} für eine Taktperiode abfließen, um die Nähe der Einsatzspannung zu erreichen.

Für eine spätere Implementierung wird es bezüglich dieser Ströme notwendig sein, mit analogen Maßnahmen dafür zu sorgen, dass die Ströme rechtzeitig abgeschaltet werden, um ungewollte Spannungsspitzen, die sich wiederum negativ auf das [EMV](#)-Verhalten auswirken, zu vermeiden.

5.11 Dimensionierung der Messkapazität

Die Größe der Messkapazität C_{mess} wurde weiter oben mit einem runden Wert angenommen. Doch für eine Spezifikation sollte dieser Wert auf eine Applikation dimensioniert werden, um unnötige hohe Verluste zu vermeiden. Nach der Relation (4.12) verhält sich der resultierende Verschiebungsstrom proportional zum Kapazitätswert. Nach der verwendeten Konfiguration muss dieser Strom in eine Messspannung umgewandelt werden, was am einfachsten mit einem Widerstand funktioniert. Allem Anschein nach sollte die Kapazität also so klein wie möglich gewählt werden, um möglichst wenig Leistung zu verbrauchen.

Eine untere Grenze für die Wahl der Kapazität bildet die Rauschanforderung an den A/D Umsetzer. Je kleiner die Stufenbreite ist, desto kleiner muss auch die Rauschspannung sein und dies hängt von der gewählten Genauigkeit und der verwendeten

Referenzspannung des A/D Umsetzers ab.

Für das verwendete Modell wurde eine Referenzspannung von 1,23 Volt verwendet und die höchste Auflösung beträgt 6 Bit. Der nötige Messwiderstand richtet sich nach dem mit Gleichung (5.2) und Gleichung (5.3) definierten Messbereich:

$$R_{mess} = \frac{V_{ref}}{I_{max} - I_{min}} \quad (5.66)$$

Nach [JM97] beträgt die Rauschleistungsdichte an einem Widerstand R_{mess} :

$$V_R^2(f) = 4kTR_{mess} \quad \left[\frac{V^2}{Hz} \right] \quad (5.67)$$

k Boltzmann-Konstante
T Temperatur in °Kelvin

Mit den aus Tabelle 5.2 gewählten Werten für $I_{MAX}=1,022$ mA und $I_{MIN}=438$ μ A, welche für den Messbereich 0,4 dimensioniert sind, errechnet sich der nötige Messwiderstand wie folgt:

$$R_{mess} = \frac{1,23V}{1,022 mA - 438 \mu A} = 2054,79 \Omega$$

Mit (5.67) beträgt die spektrale Rauschleistungsdichte gleich:

$$V_R^2(f) = 4 \cdot 1,23 \cdot 10^{-23} \cdot 300^\circ \cdot 2054,79 = 3,4027 \cdot 10^{-17} \frac{V^2}{Hz}$$

Mit einer Bandbreite von $B = 80$ MHz (durch die Abtastrate gegeben) kommt man so zu einer äquivalenten Rauschspannung:

$$V_{R(rms)} = \sqrt{3,4027 \cdot 10^{-17} \cdot 80 \cdot 10^6} = 52 \mu V$$

Mit einem 6 Bit A/D Umsetzer beträgt die Stufenbreite $\frac{1,23V}{2^6} = 19,2$ mV, womit man auf einen Signal-/Rauschabstand von

$$S/N = 20 \log \left(\frac{19,2 mV}{52 \mu V} \right) = \underline{\underline{51,33 dB}}$$

kommt.

Ist also der Messbereich, die Auflösung des A/D Umsetzers und der nötige Rauschabstand bekannt, kann der Kapazitätswert bestimmt werden. Des Weiteren ist zu beachten, je geringer der Messbereich gewählt wird, umso größer muss auch die Messkapazität dimensioniert werden, also richtet sich diese bei variablem Messbereich nach dem

Kleinsten.

$$V_{R(rms)} = \left(\frac{1}{10}\right)^{\frac{S/N}{20}} \cdot V_{ST} \quad (5.68)$$

$$R_{mess} = \frac{V_{R(rms)}^2}{4kTB} \quad (5.69)$$

Es gilt mit (5.66)

$$\begin{aligned} R_{mess} &= \frac{V_{REF}}{I_{MAX} - I_{MIN}} = \frac{V_{REF}}{I_{FBnom} \cdot (1 + \delta_M) - I_{FBnom} \cdot (1 - \delta_M)} \\ &= \frac{V_{REF}}{I_{FBnom} \cdot (2\delta_M)} \\ &= \frac{V_{REF}}{C_{mess} \frac{dU}{dt} \cdot (2\delta_M)}. \end{aligned}$$

Und nach Gleichsetzen der beiden Gleichungen für R_{mess} , erhält man für C_{mess}

$$C_{mess} = \frac{4kTBV_{REF}}{V_{R(rms)}^2 \frac{dU}{dt} 2\delta_M}. \quad (5.70)$$

5.12 Auflösung und Wahl der Abtastzeit

Die Wahl der Diskretisierungszeit für ein zeitdiskretes System hängt normalerweise eng mit den Zeitkonstanten der Regelstrecke zusammen. Die Abtastzeit kann prinzipiell wie eine Totzeit interpretiert werden und kann, sollte sie zu groß gewählt werden, auch zur Instabilität des Systems führen. Im vorliegenden Fall orientiert man sich nicht an den Zeitkonstanten der Strecke, sondern am gewünschten Verhalten. Das bedeutet, die untere Grenze der Abtastzeit hängt davon ab, wie oft während der Umschaltphase korrigierend eingegriffen werden muss.

Abbildung 5.21 zeigt die Auflösung des Systems, bezogen auf den Zeit- und Wertebereich. Hier herrscht nun ein Kompromiss dieser beiden Komponenten. Es wird einerseits nicht zielführend sein, die Flanke ein einziges Mal mit einer Genauigkeit von 16 Bit abzutasten und gegensätzlich wird es mit hundert Abtastwerten, welche mit einer Genauigkeit von einem Bit gemacht wurden, schwer sein, eine zufriedenstellende Korrektur vorzunehmen.

Die Abtastrate (Sampling rate) richtet sich im vorliegenden Fall nach der Anzahl der Punkte, die pro Flanke berücksichtigt werden sollen.

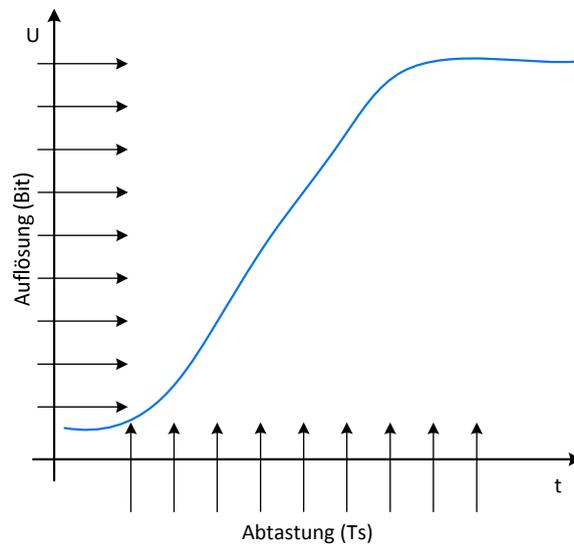


Abb. 5.21: Auflösung der Flanke

Ausgehend von den bekannten 100 ns aus Tabelle 5.1 bedeutet das für zehn Punkte, dass man eine Abtastrate von 100 MHz benötigen würde (100 Mega-*samples*). Aufgrund von ersten modellbasierten Simulationen kann auf eine Abtastrate von 40 - 80 Mega-*samples* geschlossen werden.

Für die Bestimmung der Auflösung in Bit kann die Tabelle 5.2 herangezogen werden. Spezifiziert man den Betrachtungsbereich und definiert man die maximale Abweichung unter der Annahme eines verschwindenden Regelfehlers, kann man die Auflösung in Bit direkt bestimmen.

Als Beispiel soll dieser Vorgang an dieser Stelle erläutert werden: Für einen Beobachtungsbereich von $\pm 30\%$ und einer Abweichung von maximal $13,5 \text{ V}/\mu\text{s}$ ($1/2 \times 27 \text{ V}/\mu\text{s}$) ergibt sich laut Tabelle 5.2 ein 4 Bit **ADC**.

Abbildung 5.22 veranschaulicht die für den digitalen Regler notwendigen Komponenten. Hier ist besonders zu beachten, dass für deren Implementierung ein hoher Preis gezahlt werden muss.

- Die Messung, welche vor der Wandlung geschieht, um die Ausgangsgröße geeignet umzuformen, benötigt schaltungstechnischen Aufwand. Vor allem muss diese Messung viel schneller erfolgen als die Diskretisierungszeit, um ein eingeschwungenes Messsignal für den **ADC** Eingang zu liefern.
- Ein digitaler Proportional-Integral-Derivativer (**PID**) Regelalgorithmus besteht im Allgemeinen aus der Form:

$$c[n] = c[n - 1] + a \cdot e[n] + b \cdot e[n - 1] + c \cdot e[n - 2]$$

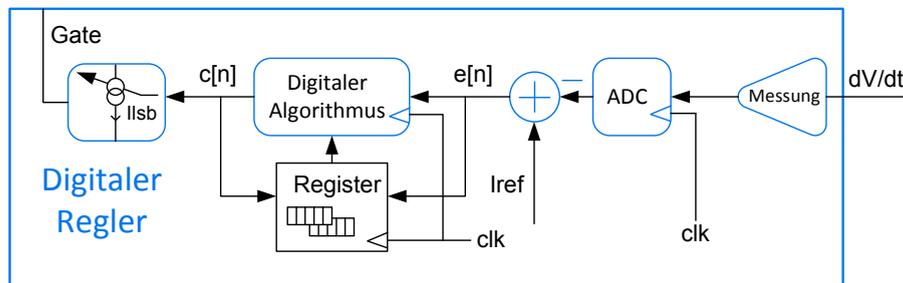


Abb. 5.22: Beschränkungen durch den digitalen Regler

Für eine Implementierung müssen drei hochauflösende Multiplikationen sowie drei Additionen innerhalb von Bruchteilen der Diskretisierungszeit gemacht werden!

- Der **ADC** bildet vor allem in den meisten Fällen den "Flaschenhals" des digital geregelten Systems, da die Einschwingzeit für diverse Komponenten (Komparatoren) ebenfalls in Bruchteilen der Diskretisierungszeit liegen muss. Dies resultiert schnell im Leistungsverbrauch von dutzenden mW [Tre10]!

Ausgehend von diesen Begrenzungen kann man schließen, dass eine digitale Regeleinrichtung in den meisten Fällen nicht als Ersatz für die analoge Variante herangezogen werden sollte, sondern man sollte eine digitale Implementierung nur dann vorziehen, wenn notwendige Operationen mit analogen Komponenten nicht durchgeführt werden können! Dies zielt im vorliegenden Fall auf die Konfigurierbarkeit und Anpassung an verschiedene Systeme ab.

Aus diesen Erkenntnissen geht hervor, dass für eine Implementierung die nachführende Regelung interessant ist, da die Berechnungen der Ausgangswerte während der Flanke geschieht und so die Komplexität des Reglers im Hinblick auf Rechenoperationen gering gehalten werden kann. Abbildung 5.23 zeigt eine Übersicht diverser **ADC** Umsetzungsverfahren in Abhängigkeit von der Auflösung und der Abtastrate [Kut08]. Daraus ist zu entnehmen, dass für die vorliegenden Anforderungen von 4 Bit und 80 *Megasamples* am besten ein Flash **ADC** verwendet werden sollte.

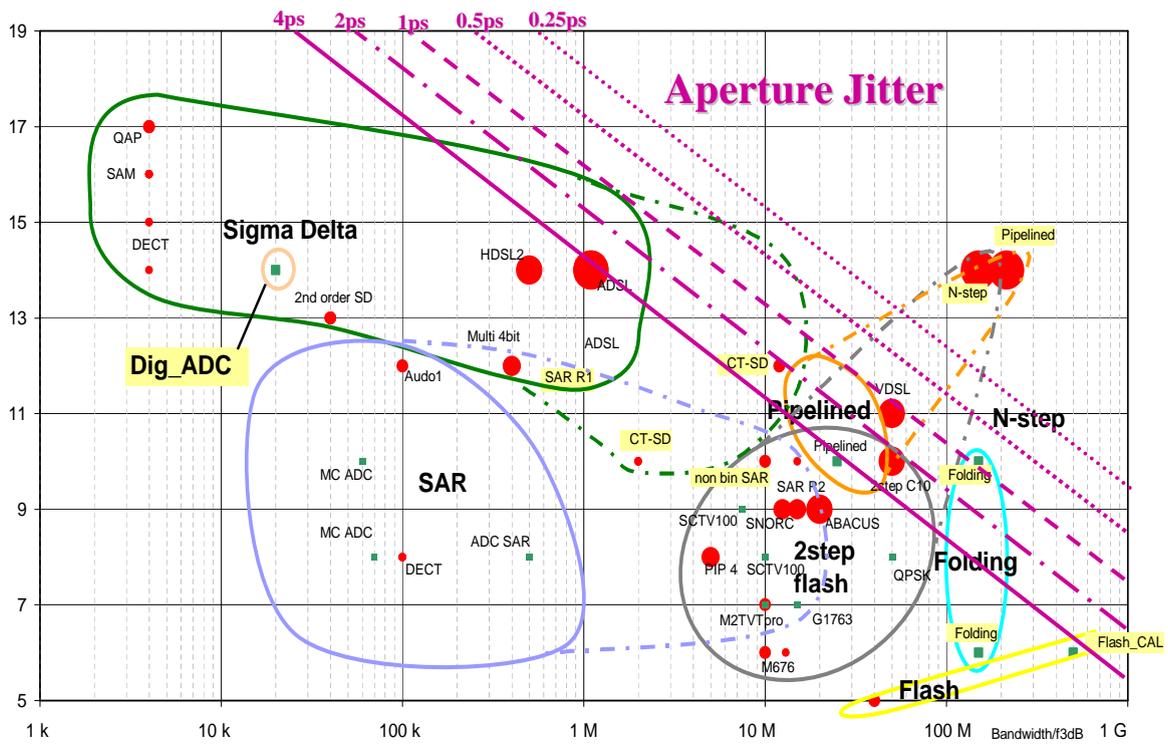


Abb. 5.23: AD/DA-Umsetzer Auflösung und Abtastrate für diverse Architektur [Kut08]

5.13 Stabilitätsbetrachtungen

Um sicherzustellen, dass das geregelte System auch stabil ist, muss ausgehend vom Komponentenmodell in Abbildung 5.16 ein System abstrahiert werden, welches nur aus den mathematisch interessanten Komponenten während der Regelphase besteht.

Hierzu wird der Ausgangsvektor des Modells aus Gleichung (5.30) etwas verändert. Anstelle der 1 wird die Messkapazität eingesetzt:

$$C = [0 \quad C_{mess} \quad 0] x(t) \quad (5.71)$$

Durch diese Veränderung erhält man als Ausgangswert im dynamischen Fall ($\frac{dU}{dt} \neq 0$) den Rückwirkungsstrom der Messkapazität.

Mit den Übertragungsfunktionen aus Gleichung (5.61) und Gleichung (5.29) hat man das eigentliche System bereits beschrieben. Während der Regelphase sind nun nur noch der A/D Umsetzer sowie der Strom-DAC aktiv, welche beide mit einem Verstärkungsfaktor beschrieben werden können.

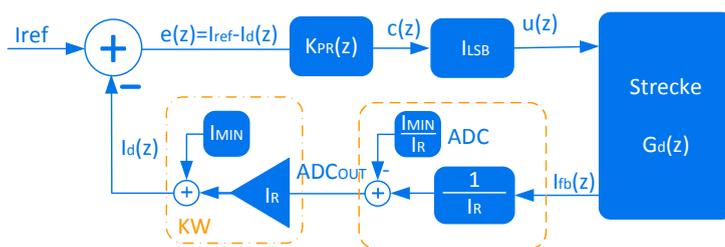


Abb. 5.24: Struktur des Standardregelkreises mit Umformung durch den A/D Umsetzer

Abbildung 5.24 zeigt, wie der durch den A/D Umsetzer einwirkende Verstärkungsfaktor durch Änderung der Referenzgröße in den Vorwärtszweig verschoben und dort durch Multiplikation mit $\frac{1}{I_R}$ kompensiert werden kann. Der mit KW gekennzeichnete Block kann durch folgende Umformung für den Algorithmus vermieden werden. Der Regelfehler ergibt sich bei obiger Konfiguration, wie in Abbildung 5.24 dargestellt.

$$e(z) = I_{ref} - I_d(z) \quad (5.72)$$

$I_d(z)$ entspricht hier genau $I_{fb}(z)$. Wollte man nun die Multiplikation sowie die Addition, die in dem Block KW durchgeführt werden, einsparen, müsste man den Referenzstrom den selben Einflüssen des A/D Umsetzers unterziehen.

$$e(z) = \left[\frac{I_{ref}}{I_R} - \frac{I_{min}}{I_R} \right] - \left[\frac{I_{fb}}{I_R} - \frac{I_{min}}{I_R} \right] = \frac{I_{ref}}{I_R} - \frac{I_{min}}{I_R} - \frac{I_{fb}}{I_R} + \frac{I_{min}}{I_R} = (I_{ref} - I_{fb}) \frac{1}{I_R} \quad (5.73)$$

Auf diese Art wird der Faktor $\frac{1}{I_R}$ direkt in den Proportionalfaktor des Reglers integriert. Nun kann man, wie in Abbildung 5.26, die Schleife "aufbrechen" und Untersuchungen

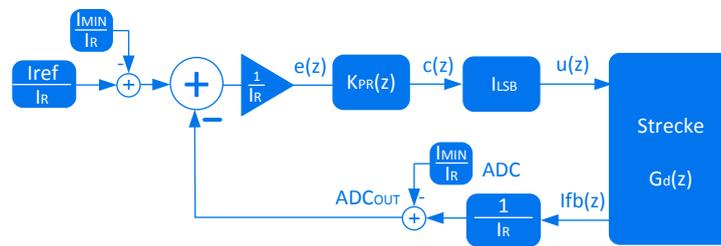


Abb. 5.25: Struktur nach Umformung der Größen

bezüglich der Stabilität im Hinblick auf die Implementierung durchführen. Des Weiteren ist an dieser Stelle zu bemerken, dass die Verzögerungszeit des A/D Umsetzers (Wandelzeit) bei der nachlaufenden Regelung für Stabilitätsbetrachtungen keinen Einfluss hat, da zwischen Wandlung und Setzen der Stellgröße während des stationären Zustandes die Regelschleife nicht aktiv ist! Dies wird weiter unten in Kapitel 5.15.2 gezeigt.

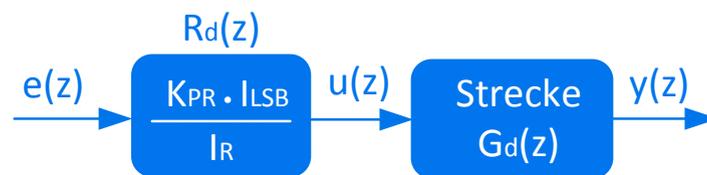


Abb. 5.26: Struktur für die Untersuchung der offene Schleife

5.14 Vorschlag für die Implementierung des Reglers

Die Implementierung des Reglers soll hier anhand diverser Möglichkeiten behandelt werden.

Den Ablauf der einzelnen Rechenoperationen für die PI-Struktur zeigt Abbildung 5.27, angelehnt an das Blockdiagramm für eine PID Struktur [Pro01].

Da die Berechnungen nicht während der Flanke gemacht werden, gibt es eine Reihe von Ansätzen für die Implementierung, um die Komplexität möglichst niedrig zu halten.

Eine dieser Möglichkeiten wäre, Multiplikationen mit großen Faktoren durch Bit- Schie-

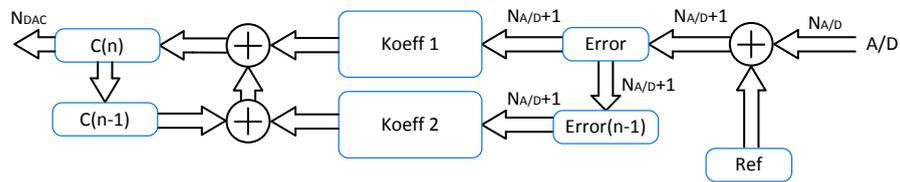


Abb. 5.27: Block Diagramm der Reglereinheit

be Operationen zu ersetzen. Hierzu betrachte man folgenden Regelalgorithmus:

$$d[n] = d[n - 1] + 49,2(e[n] - 1,3e[n - 1] + 0,81e[n - 2])$$

Die vorliegende Gleichung enthält einen großen Faktor vor der Klammer. Um die Anzahl der notwendigen Multiplikationen zu verringern, kann obige Gleichung wie folgt umgeschrieben werden.

$$d[n] = d[n - 1] + 64(0,769e[n] - e[n - 1] + 0,63e[n - 2])$$

Der dadurch erzielte Gewinn wird deutlich, wenn man den Multiplikator vor der Klammer betrachtet. Anstatt der tatsächlichen Multiplikation mit der Zweierpotenz, müssen die Werte in der Klammer einfach durch eine Bit-Schiebe-Operation um 6 Stellen verschoben werden, was den Aufwand für die Multiplikation drastisch reduziert [Pro01].

Die notwendigen Anforderungen an eine Implementierung sind größtenteils dadurch gegeben, dass während des stationären Zustands zwar durchaus genügend Zeit für die Berechnung zur Verfügung steht, allerdings müssen auch mehrere Stellwerte berechnet werden.

Ausgehend von diesem Sachverhalt liegt eine Implementierung nahe, die die Berechnungen für einen Wert sehr schnell machen kann, um danach auf einen anderen Kanal umzuschalten. Dieses Konzept wird als *Parallel PID based multiple-channel design* in [ZKLV05] beschrieben.

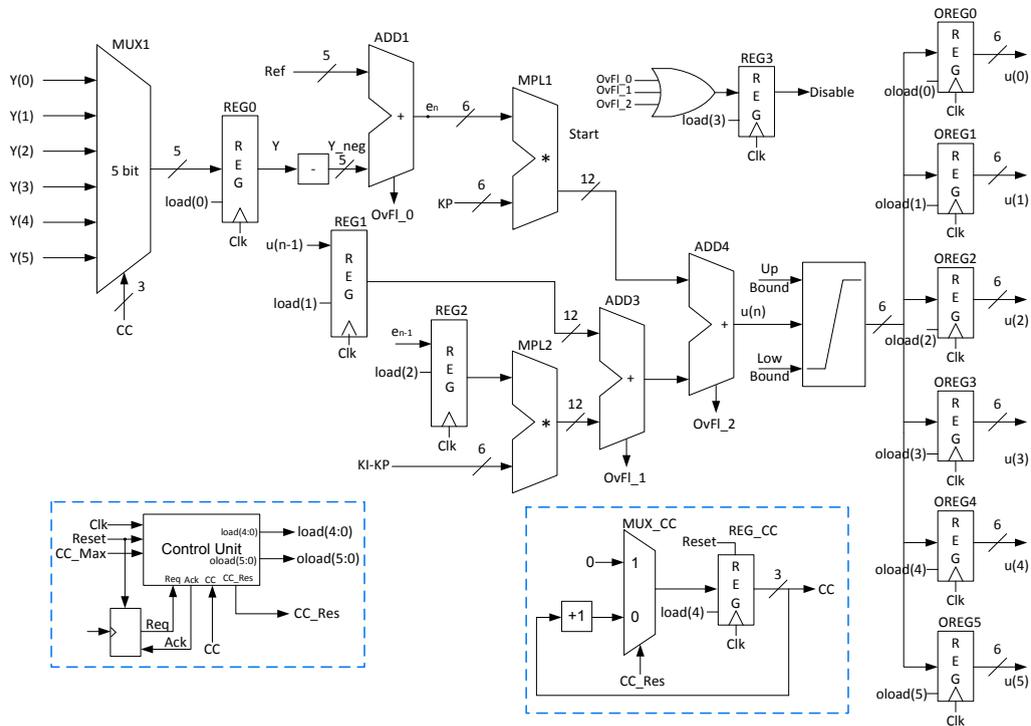


Abb. 5.28: Konzeptbild für die PI-Regler Implementierung

Abbildung 5.28 zeigt das vorgeschlagene Konzept für die digitale PI-Reglerstruktur. Die Berechnung des Ausgangswertes erfolgt sehr parallelisiert. So kann der Ausgangswert für jeden Messpunkt in einem Takt nacheinander berechnet werden.

Der Multiplexer MUX1 dient zum Kontextwechsel zwischen den einzelnen Regelkreisen und bildet den Eingang. Durch bitweise Invertierung und Addition mit 1 wird die Negation vorgenommen. In Folge werden drei Addierer und zwei Multiplexer zur Berechnung verwendet. Am Ausgang sorgt eine Beschränkung dafür, dass die Ausgangswerte im definierten Bereich bleiben.

Die Werte für $u(n-1)$ bzw. $e(n-1)$ müssen vor jeder Berechnung separat in die Register REG1 bzw. REG2 geladen werden. Das bedeutet auch, nach jeder Berechnung vor dem Kontextwechsel müssen diese Werte in entsprechenden Registern gesichert werden.

5.15.2 Dimensionierung für einen Leistungs-MOSFET

In diesem Beispiel soll nun anhand der oben erläuterten Zusammenhänge eine Dimensionierung für eine vorgegebene Konfiguration durchgeführt werden. Diese Dimensionierung gilt nur für einen Arbeitspunkt und muss für jeden Arbeitspunkt, je nach Anzahl der berücksichtigten Punkte, separat durchgeführt werden.

Den Ausgangspunkt bilden jene Größen, die bekannt sein müssen:

	Wert
Spannungsbereich	0 - 73 V
Slew rate	730 V/ μ s
Rs	2 Ω
Abtastrate	70 MHz
Solenoid	2 mH
gm des MOSFET	4,66 S
Cg des MOSFET	47 pF
Cgs des MOSFET	23,5 pF
V _{TH} des MOSFET	1,36 V
Q _G (th)	1,062 nC
Q _G ⁻ (th)	1,358 nC
Messbereich δ_M	$\pm 40\%$ I _{NOM}
S/N für A/D Umsetzer	58 dB
Auflösung A/D Umsetzer	5 Bit
Auflösung D/A Umsetzer	6 Bit
Referenzspannung A/D	1,23 V

Tab. 5.3: Bekannte Größen für die Dimensionierung

Man beginnt mit der Bestimmung der Rauschspannung.

$$V_{R(rms)} = \left(\frac{1}{10}\right)^{\frac{58}{20}} \cdot \Delta V_{ADC} = \left(\frac{1}{10}\right)^{\frac{58}{20}} \cdot \frac{1,23}{2^5} = \underline{\underline{48,39 \mu V}}$$

Als Nächstes kann bereits die Messkapazität bestimmt werden.

$$C_{mess} = \frac{4kTBV_{REF}}{V_{R(rms)}^2 \frac{dU}{dt} 2\delta_M} = \frac{4 \cdot 1,38 \cdot 10^{-23} \cdot 300 \cdot 70 \cdot 10^6 \cdot 1,23}{(48,39 \cdot 10^{-6})^2 \cdot 730 \cdot 10^6 \cdot 2 \cdot 0,4} = \underline{\underline{1,043 pF \approx 1 pF}}$$

Nun kann als Nächstes der nominelle Rückwirkungsstrom I_{Nom} sowie die Stromgrenzen

I_{MAX} und I_{MIN} bestimmt werden:

$$\begin{aligned} I_{Nom} &= C_{mess} \cdot \frac{dU}{dt} = \underline{\underline{730 \mu A}} \\ I_{MAX} &= I_{Nom}(1 + \delta_M) = \underline{\underline{1,022 \mu A}} \\ I_{MIN} &= I_{Nom}(1 - \delta_M) = \underline{\underline{438 \mu A}} \end{aligned}$$

Mit Gleichung 5.66 kann nun der Messwiderstand bestimmt werden:

$$R_{mess} = \frac{V_{REF}}{I_{MAX} - I_{MIN}} = \underline{\underline{2106,16 \Omega}}$$

Für den Reaktionsstrom ergibt sich:

$$I_R = \frac{I_{max} - I_{min}}{2^5} = \underline{\underline{18,25 \mu A}}$$

Und nun die Berechnung für die Initialströme:

$$\begin{aligned} I_{slope} &= C_{Gate} \frac{dU}{dt} \kappa = 47 \cdot 10^{-12} \cdot 730 \cdot 10^6 \cdot 0,9 = \underline{\underline{30,8 mA}} \\ I_{Push} &= \frac{Q_G(th)}{T_A} = \frac{1,061510^{-9}}{14,285 \cdot 10^{-9}} = \underline{\underline{74,3 mA}} \\ I_{Pull} &= \frac{Q_G^-(th)}{T_A} = \frac{1,35810^{-9}}{14,285 \cdot 10^{-9}} = \underline{\underline{95,1 mA}} \end{aligned}$$

Da bei dem verwendeten Modell auch der Strom I_{slope} bereits vor dem Erreichen der Schwellenspannung fließt, muss dieser von den Strömen I_{Push} und I_{Pull} subtrahiert werden.

Schließlich noch die Berechnung des LSB Stroms:

$$I_{LSB} = \frac{C_{Gate}}{C_{mess}} I_R \frac{2^{N_{ADC}}}{2^{N_{DAC}}} = \frac{47 \cdot 10^{-12}}{1 \cdot 10^{-12}} 18,25 \cdot 10^{-6} \frac{2^5}{2^6} = \underline{\underline{429 \mu A}}$$

Da nun alle Größen der Strecke und der Wandler bekannt sind, kann mit der Dimensionierung des Reglers begonnen werden. Die Struktur des Reglers von Gleichung (5.61) wird hier noch einmal angeführt:

$$R_d(z) = K_p \frac{z - 1 + \frac{T_A}{T_N}}{z - 1} = K_p + \frac{K_p T_A}{(z - 1) T_N} \quad (5.74)$$

Hier müssen die Parameter T_N und K_p noch bestimmt werden.

Für die Bestimmung dieser Parameter eignen sich das Frequenzkennlinienverfahren für

Abtastsysteme [GHS91] oder das Wurzelortskurvenverfahren für zeitdiskrete Systeme [DB06] sehr gut. Vor allem Ersteres soll im folgenden Abschnitt genauer beschrieben werden.

Dimensionierung der Regelerparameter nach dem Frequenzkennlinienverfahren

Dieses Verfahren ist in der Vorgehensweise sehr ähnlich dem FKL-Verfahren für zeitkontinuierliche Systeme. Da man es aber hier mit z-Übtragungsfunktionen zu tun hat, muss man erst eine bilineare Transformation durchführen, um bezüglich der Frequenz eine gebrochen-rationale Funktion zu erhalten, mit der man dann das Verfahren durchführen kann.

$$G^\#(q) := G(z) \Big|_{z = \frac{1+q\frac{T_A}{T_s}}{1-q\frac{T_A}{T_s}}} \quad (5.75)$$

Für das Verfahren werden folgende zwei Faustformeln aus [GHS91] verwendet, wobei M_p die Überschwingweite und ϕ die Phasenreserve in Grad bezeichnen.

$$\Omega_{ct_r} \approx 1,2 \quad (5.76)$$

$$\phi [^\circ] + 100 (M_p - 1) \approx 70 \quad (5.77)$$

Ausgangspunkt bildet also die Regelstrecke aus (5.31), welche durch Multiplikation mit (5.32) den integrierenden Anteil verliert und als Proportionalglied mit Verzögerung wirkt.

$$P(s) = \frac{0,021013(s + 1,983 \cdot 10^{11})(s + 1000)}{(s + 1,997 \cdot 10^{11})(s + 1159)}$$

$$P^\#(q) = \frac{0,021165(q + 1,38 \cdot 10^8)(q + 1000)}{(q + 1,4 \cdot 10^8)(q + 1159)} \quad (5.78)$$

Abbildung 5.30 zeigt das Bode-Diagramm für die q-Übtragungsfunktion $\frac{P_q}{q}$, welche also bereits den integrierenden Anteil des Reglers enthält. Um nun im nächsten Schritt mit (5.76) eine Transitfrequenz errechnen zu können, benötigt man zuerst eine Anstiegszeit des geregelten Systems. Hierfür muss man das System der nachlaufenden Regelung etwas verändert betrachten.

Man nimmt an, dass die Arbeitspunkte der nachlaufenden Regelung, nicht nachlaufend durchlaufen werden, sondern, dass sämtliche stationäre Zustände gar nicht auftreten. Das geregelte System wird also so betrachtet, als würde ein Arbeitspunkt gemessen und nach der Messung (eine Taktperiode später) sofort nachgeregelt werden. Dies geschieht

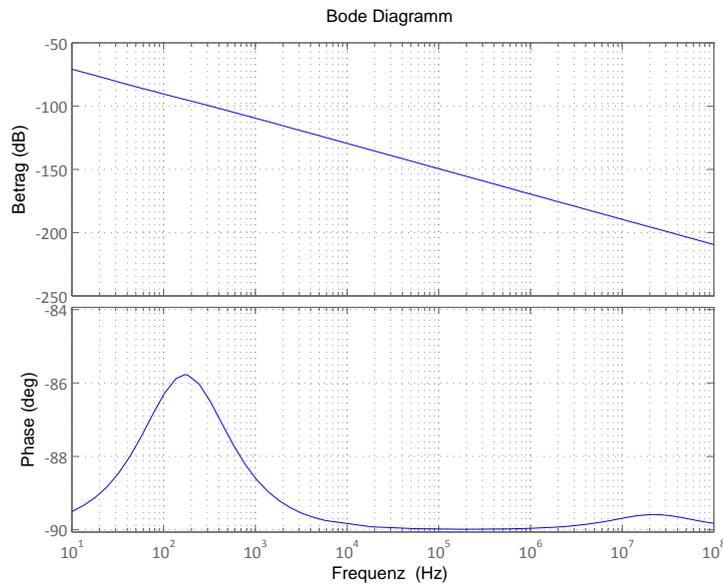


Abb. 5.30: Bode-Diagramm $P(q)/q$

dann solange, bis der Referenzwert erreicht ist und der Regelfehler verschwindet. Nun kann man genau die Anzahl an Taktperioden, bis der Regelfehler verschwindet, als Anstiegszeit des Reglers interpretieren.

Auf diese Art wird es auch möglich, die Anzahl der Flankenübergänge einzustellen, welche durchlaufen wird, bevor die Regelung die vorgegebene Flanke liefert.

Obige Erläuterung gilt für steigende und fallende Flanke getrennt. Da jedoch die Berechnung der Stellgröße alternierend abläuft, muss die Anstiegszeit verdoppelt werden. Als Vorgabe wird der Wert 14 gewählt.

$$t_r = 2 \cdot 14 \cdot T_A = \underline{\underline{400 \text{ ns}}}$$

$$\Omega_C = \frac{1,2}{t_r} = \underline{\underline{3,6 \text{ MHz}}}$$

Aus Abbildung 5.30 entnimmt man, dass für eine Transitfrequenz von 3,6 MHz der Betrag um 180,59 dB angehoben werden muss.

So ergibt sich für den Regler:

$$R^\#(q) = \frac{1,07 \cdot 10^9}{q}. \tag{5.79}$$

Für die Phasenreserve ergibt sich ein Wert von 90° aufgrund des integrierenden Anteils des Reglers. Da ein mögliches Überschwingen für die Flankenregelung eher vermieden werden soll, da dies mögliche "Knicke" in der Einregelphase nach sich ziehen würde, was

sich wiederum im Hinblick auf Abstrahlung negativ auswirkt, kann die Phasenreserve so bleiben. Nach der Rücktransformation erhält man für den Regler

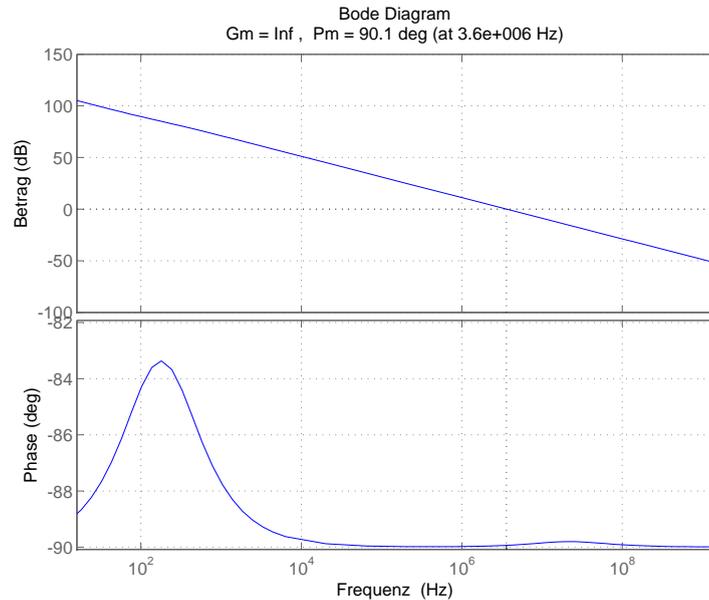


Abb. 5.31: Bode-Diagramm $L(q)$

$$R(z) = \frac{7,646z + 7,646}{z - 1} \quad (5.80)$$

Um dies in das VHDL-Modell einfügen zu können, muss noch mit dem Faktor $\frac{I_R}{I_{LSB}}$ multipliziert werden.

$$\begin{aligned} \frac{7,646z + 7,646}{z - 1} &= K_p \frac{z - 1 + \frac{T_A}{T_N}}{z - 1} \\ K_p &= 7,646 \cdot \frac{I_R}{I_{LSB}} = \underline{\underline{0,162}} \\ T_N &= T_A/2 = \underline{\underline{7,1428 \text{ ns}}} \end{aligned}$$

Setzt man die gewonnenen Parameter in das Modell ein und startet eine Simulation, kann man den verschwindenden Regelfehler bzw. das Führungsverhalten verfolgen. Bezieht man die Führungsgröße nun noch auf die Referenz, erhält man eine Sprungantwort mit dem Zielwert 1. Abbildung 5.32 zeigt die Sprungantwort des idealen Modells im Vergleich mit Daten aus dem schon mehr implementierungsnahen VHDL-AMS Modell, welches bereits mit quantisierten Werten rechnet.

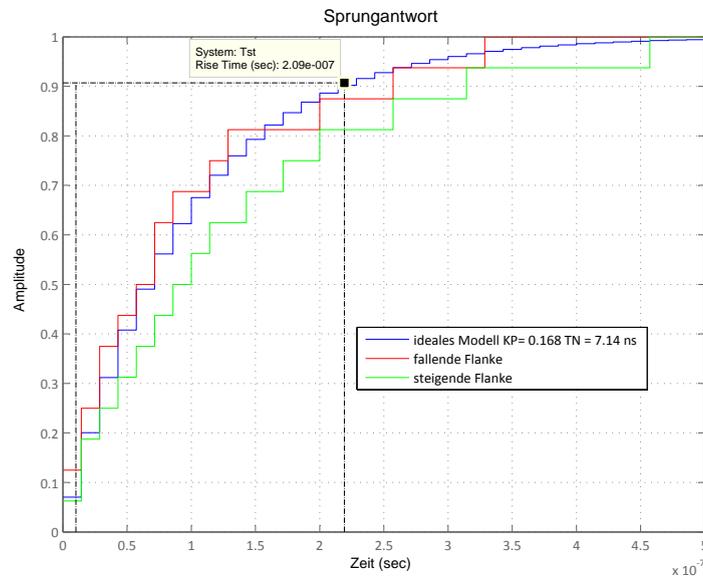


Abb. 5.32: Sprungantwort des geregelten Systems

Dimensionierung der Reglerparameter mit Wurzelortskurven

Für dieses Verfahren wird der Parameter T_N zuerst passend gewählt $T_N \approx \frac{1}{2}T_A$. Dies ist insofern zulässig, da der Regelfehler unabhängig vom gewählten T_N verschwindet [RZ08].

Für T_N wird ein Wert von 6,26 ns gewählt. Nun kann die Struktur des Reglers angegeben werden:

$$R_d(z) = \frac{z - 1 + \frac{T_A}{T_N}}{z - 1} \cdot \frac{I_{LSB}}{I_R}$$

Die diskretisierte Strecke sieht wie folgt aus:

$$P_d(z) = \frac{0,021277(z - 1)(z - 0,007262)}{z(z - 1)} \cdot \frac{I_{LSB}}{I_R}$$

Nun bildet man den Vorwärtskreis $L_d = R_d P_d$ und zeichnet die Wurzelortskurve:

Abbildung 5.33 zeigt die Wurzelortskurve des offenen Kreises.

Die charakteristische Gleichung lautet:

$$1 + K_p R_d(z) P_d(z) = 0 \tag{5.81}$$

Die wichtigsten Eigenschaften von Wurzelortskurven in der z -Ebene seien hier kurz

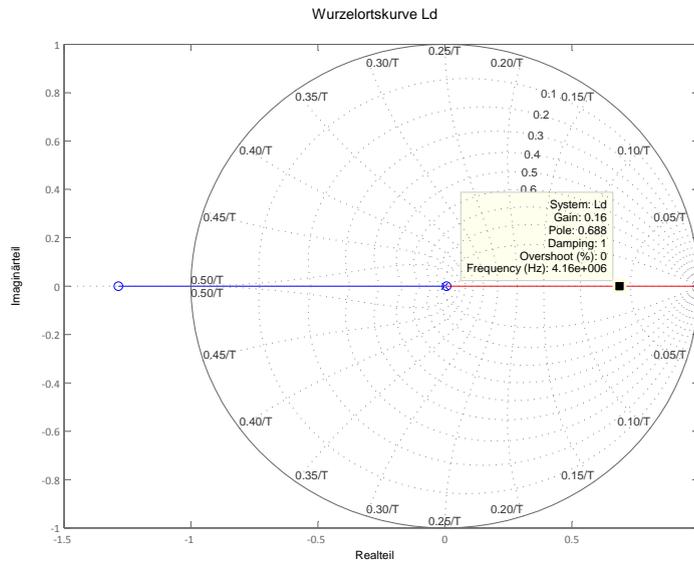


Abb. 5.33: Wurzelortskurve des Vorwärtszweigs

aufgelistet [DB06]:

- Die Wurzelortskurve beginnt an den Polen und läuft zu den Nullstellen.
- Sie ist symmetrisch zur reellen Achse und liegt in Bereichen links von einer ungeraden Anzahl von Polen und Nullstellen.

Der Verlauf der Wurzelortskurve liegt auf der reellen Achse. Nun wählt man einen Wert innerhalb des Einheitskreises, um ein stabiles System zu erhalten. Am besten eignet sich die Ortskurve im rechten Halbkreis, da in diesem Bereich kein Überschwingen stattfindet. Mit dem gewählten Verstärkungsfaktor kann nun wieder die Sprungantwort zur Überprüfung der Anstiegszeit betrachtet werden.

Abbildung 5.35 zeigt Simulationsergebnisse des VHDL-Modells, wobei im oberen Abschnitt Bedingung (5.36) verletzt wurde. Das Ergebnis sind auftretende Grenzyklus-Schwingungen. Im unteren Bereich wurde dieselbe Simulation unter Einhaltung der Bedingung durchgeführt. Die Schwingungen sind so nicht mehr vorhanden!

Abbildung 5.36 zeigt die Ergebnisse einer Simulation der digitalen Regelung. Die Steigung der Flanke verläuft konstant, was auch am Strom I_{FB} abzulesen ist, da dieser ebenfalls während des Schaltvorgangs einen konstanten Wert hat.

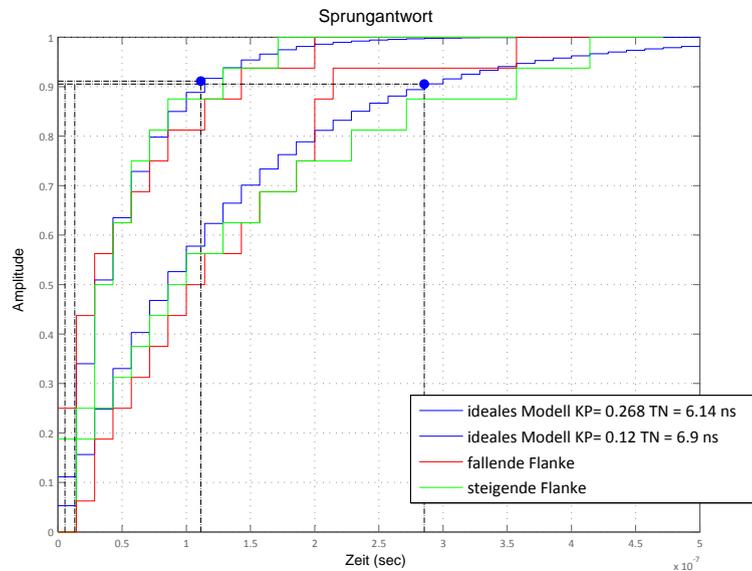


Abb. 5.34: Sprungantworten für diverse Parameter

In Abbildung 5.37 ist eine weitere Simulation dargestellt. Hier ist auch der Drain-Strom durch den DMOS-Transistor abgebildet. Hier wurde die Anstiegszeit auf 360 ns verlängert.

In Abbildung 5.38 wurde eine Simulation mit einem 4 Bit ADC für dasselbe Beispiel durchgeführt. Die Regelabweichung ist etwas größer, jedoch noch immer akzeptabel. Der maximale Fehler beträgt, wie in dem Bild zu sehen ist, $18,4\text{ V}/\mu\text{s}$, was ungefähr dem Wert in Tabelle 5.2 ($19,47\text{ V}/\mu\text{s}$) für 4 Bit und einem δ_M von 0,4 entspricht.

Dieses Entwurfsbeispiel zeigt, dass durch die gewählte Konfiguration zusammen mit der digitalen Reglerstruktur die Vorgabe der Flanke sehr gut erfüllt werden kann.

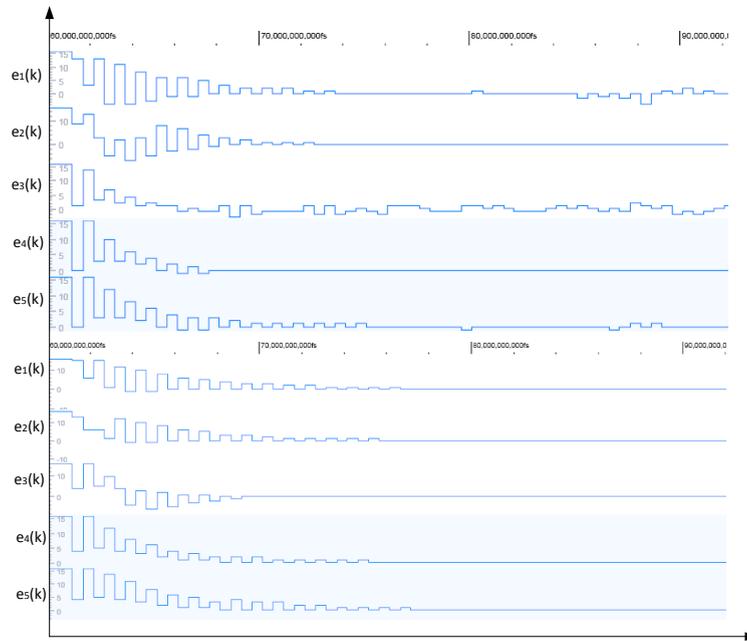


Abb. 5.35: Vergleich der Fehlerverläufe mit und ohne Grenzyklus-Schwingungen

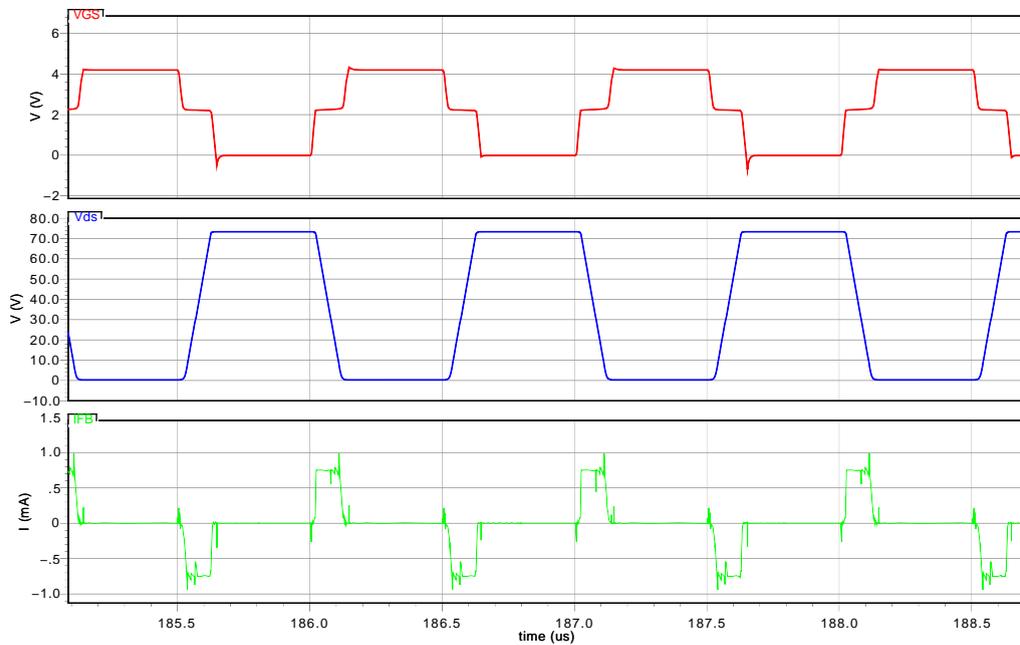


Abb. 5.36: Transiente Simulation der Drain-Source-Spannung (blau), V_{GS} (rot) und I_{FB} (grün) im eingeregelt Zustand

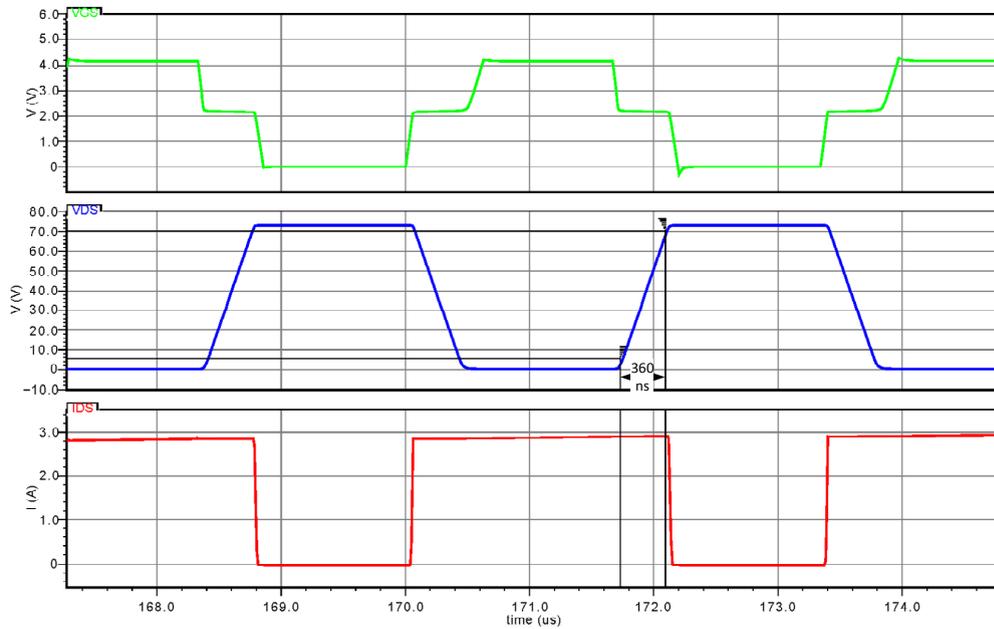


Abb. 5.37: Transiente Simulation der Drain-Source-Spannung (blau), V_{GS} (grün) und I_{DS} (rot), $t_r = 360$ ns

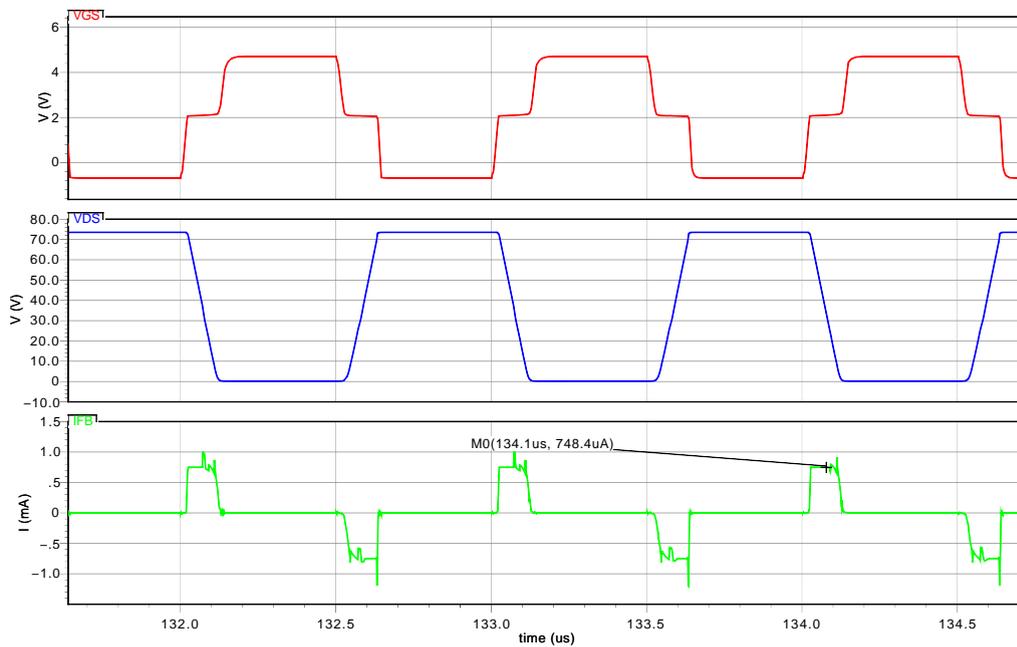


Abb. 5.38: Transiente Simulation mit 4 Bit ADC

6 Zusammenfassung und Ausblick

6.1 Kapitelübersicht

Kapitel 2 beschreibt die verschiedenen Kopplungsarten sowie die Grundursachen für **EMV** Störungen. Auch der Zusammenhang zwischen Schaltsignalen im Zeit- und Frequenzbereich wird erörtert, mit dem Ziel, die grundlegenden Maßnahmen zur Abschwächung gegen **EMV** Störaussendungen, bedingt durch steile Flanken, zu identifizieren.

Kapitel 3 zeigt den Aufbau von hochvoltfähigen MOS-Transistoren und deren parasitären Kapazitäten. Weiters wird das Schaltverhalten analysiert sowie Möglichkeiten aufgezeigt, dies gezielt zu beeinflussen.

Kapitel 4 ist ein Fallbeispiel, indem eine rein analoge Regelung der Flanke gemäß der Spezifikation implementiert wird. Schrittweise wird beschrieben, wie man von einem analogen Konzept zur Schaltung übergeht, mit welchen Mitteln die Stabilität sichergestellt wird und wie die finale Schaltung gegen Störeinflüsse, wie etwa **ESD**, geschützt wird.

Kapitel 5 beschreibt, wie sich durch den Einsatz digitaler Flankenregelung neue Möglichkeiten im Hinblick auf Änderbarkeit der äußeren Beschaltung in puncto Größe (Leistungs-MOSFET und *Solenoid*) ergeben und welche Komponenten dafür notwendig sind. Auch die Dimensionierung mit gängigen Entwurfsverfahren, ausgehend von Kleinsignalmodellen, für den benötigten PI-Regler wurde anhand von Beispielen erläutert.

6.2 Ergebnisse und Diskussion

Abschließend sollen nun die Ergebnisse dieser Arbeit sowie die zentralen innovativen Punkte an dieser Stelle zusammengefasst werden. Aus den durchgeführten Voruntersuchungen zum Thema **EMV**-Abstrahlung bzw. Flankenformen, resultieren folgende Punkte, die eingehalten werden müssen, um Störeinflüsse, hervorgerufen durch Hochvolt-Schalterapplikationen, zu vermeiden.

- Die Flanke sollte so langsam wie möglich verlaufen, um hochfrequente Anteile zu vermeiden.
- Die Signalform sollte frei von un stetigen Übergängen ("Ecken") sein, da ansonsten ebenfalls die hochfrequenten Anteile zunehmen.
- Die Signalform sollte einen möglichst linearen Anstieg besitzen, da auf diese Art der Übergang gleichmäßig (konstantes $\frac{dU}{dt}$) verläuft.

Als weitere Grundlage wurde der Aufbau sowie das Schaltverhalten von Leistungs-MOSFETs in mehreren Zeitabschnitten während des Schaltvorganges analysiert. Auch die parasitären Kapazitäten, die den Schaltvorgang entscheidend beeinflussen, wurden erörtert. Eine Möglichkeit, die Flankenform mittels Rückkopplung gezielt vorgeben zu können, wurde prinzipiell vorgestellt. Ein wichtiger Kompromiss von gutem EMV-Verhalten gegenüber der Verlustleistung ist vor jeder Implementierung sinnvoll zu definieren.

Die implementierte Schaltung für den **LIN**-Transmitter wurde in SPT9 (Smart Power Technology (**SPT**)) als Testschaltung auf Silizium gefertigt.

Vor allem die Abmessungen des Gate-Treibers konnten klein gehalten werden, da zur Hochvolt-Abtrennung lediglich eine Kapazität diente und der Regler zum größten Teil mit Niederspannungskomponenten gebaut werden konnte.

Die Untersuchungen für die Modellbildung und Spezifikation der digitalen Regelung in Kapitel 5 bilden den Großteil der Arbeit und sind Ausgangspunkt für die Implementierung eines solchen Systems.

Es wurde ausgehend von einer Standard *High-Side*-Schalter-Struktur sowie einer Spezifikation der Flanke, ein Konzept zur digitalen Regelung erarbeitet.

Die Genauigkeit bzw. die Auflösung der Wandler wurde auf die Ausgangsgröße zurückgerechnet und so der maximale Fehler bestimmt.

Ausgehend vom **ESB** wurde die Strecke als Kleinsignalmodell, sowohl im Zustandsraum als auch als Übertragungsfunktion, beschrieben.

Als geeigneter Regelalgorithmus, vorgegeben durch das Verhalten der Strecke, wurde ein zeitdiskreter **PI**-Regler gewählt, der durch nachlaufende Berechnung der Ausgangskoeffizienten den Aufwand in einem leistbaren Rahmen hält.

Die Struktur und der Aufbau des **ADCs** bzw. des **DACs** wurden festgelegt und beschrieben. Auch ein Vorschlag für die digitale Implementierung des **PI**-Reglers wurde im Hinblick auf möglichst effiziente Verarbeitung gemacht.

Die Eigenschaften bezüglich Stabilität des digitalen Regelkreises wurden erörtert sowie anhand eines Entwurfsbeispiels beschrieben. Auch geeignete Entwurfsverfahren für zeitdiskrete Regelungssysteme wurden zur Bestimmung der Reglerkoeffizienten vorgestellt.

Jeder der beschriebenen Blöcke wurde in VHDL-AMS geeignet modelliert, um Simulationen für eine vorgegebene Streckenstruktur durchführen zu können. Die Simulationsergebnisse wurden abschließend präsentiert.

Die Simulation mit einer Hardware-Beschreibungssprache wie VHDL/ VHDL-AMS bringt ein Modell schon näher an eine Implementierung heran als beispielsweise ein Modell in MATLAB. Da sowohl analoge Flussgrößen und digitale Signale zusammen simuliert werden können, erweist sich der Umgang mit solchen Simulatoren als sehr komfortabel. Weiters können die Modelle schrittweise durch die Hardware-Implementierung ersetzt werden, was einem Top-down-Entwurf entspricht.

Ein entscheidender Punkt war die Prüfung auf Implementierbarkeit, sodass dies in einem effizienten Rahmen durchgeführt werden kann. Erst durch den Einsatz einer Technologie die mit *Deep Submicron*-Komponenten kann die geforderte Taktfrequenz (z.B. für **ADC**) erreicht werden.

Digitale Lösungen für diese Problematik sind noch nicht sehr gängig, doch vor allem die Vorteile, die dadurch im Hinblick auf ein System mit austauschbarer Beschaltung möglich gemacht werden, sind von der Kundenseite her sehr gefragt.

In der vorliegenden Lösung müssen die Transistorparameter für den externen Schalter dem System mitgeteilt werden. Danach können auch die Reglerparameter digital justiert werden. Die in diesem Zusammenhang wichtigen Größen sind:

- C_{gs} und C_{gd} bzw. C_{iss} und C_{rss} woraus erstere bestimmt werden können.
- V_{TH} des MOSFETs bzw. die Gateladungen zum Erreichen dieser
- g_m des MOSFETs

Auch bei einem sich ändernden Spannungsbereich müssen keine großen Änderungen vorgenommen werden, dieser muss nur im Rahmen einer Implementierung nach oben

hin begrenzt werden. So kann entweder der Takt entsprechend der betrachteten Punkte reduziert werden bzw. Punkte außerhalb des Übergangs werden einfach verworfen.

Mit dem digitalen Regler ist es unter anderem auch möglich, auf unterschiedliche Betriebszustände der Strecke entsprechend zu reagieren. Dies geschieht vor allem am Beginn der Schaltflanke, wenn sich die Gate-Source-Spannung noch unterhalb der Schwellenspannung befindet (erhöhtes Stromprofil).

Generell kann festgestellt werden, dass beim Einsatz der digitalen Regelung, im Vergleich zur analogen Variante, für die betrachtete Anwendung ein Vorteil im Hinblick auf leichte Änderbarkeit des Systems zu tragen kommt. Neben der Tatsache, dass dies von Kunden für externe Transistoren gewünscht wird, ist das auch besonders wichtig, wenn in neueren Unternehmensstrukturen Lösungen in Form von Modulkonzepten entwickelt werden, die so in wesentlich kürzerer Entwicklungszeit für verschiedene Applikationen eingesetzt werden können (*Reuse & Time to market*).

6.3 Ausblick

Der nächste Schritt nach der durchgeführten Arbeit besteht in der Implementierung des Gesamtsystems der digitalen Regelung. Hier stellen vor allem das Design des Gate-Treibers, der die variablen Stromquellen enthalten muss, sowie die Messumformung zusammen mit dem ADC die größten Herausforderungen dar. Die Umformung in eine Messspannung kann auch übersprungen werden, wenn der Strom von der Rückkopplung geeignet skaliert und anschließend direkt gewandelt wird.

Auch ein System, das ausreichend Intelligenz besitzt, um sich die Transistordaten selbst zu beschaffen, wäre in der Zukunft sehr wünschenswert. So würde sich der Aufwand der Kunden durch sich ändernde Größen der Transistoren bzw. Spulen weiter sehr verringern.

In diesem Zusammenhang ist ein System zur Messung der Gate-Kapazität vorstellbar, aber auch eine Lösung in Richtung adaptive Systeme, wobei zur Systemidentifikation diverse Algorithmen eingesetzt werden, ist denkbar. Dies wurde auch in Kapitel 5.6 bereits kurz angeschnitten. In diese Richtung sollen auch im Rahmen von Nachfolgearbeiten weitere Untersuchungen durchgeführt werden.

Für die Implementierung des digitalen Reglers ist das vorgestellte Konzept sehr vielversprechend, da durch das Prinzip der Kanalumschaltung kombiniert mit einer schnellen Berechnung durch die Topologie, der Regler unabhängig von der Anzahl der Messpunkte pro Flanke eingesetzt werden kann. Lediglich ein Maximum der betrachteten Punkte muss vorgegeben werden.

Anhang

A Schematic der Regelung

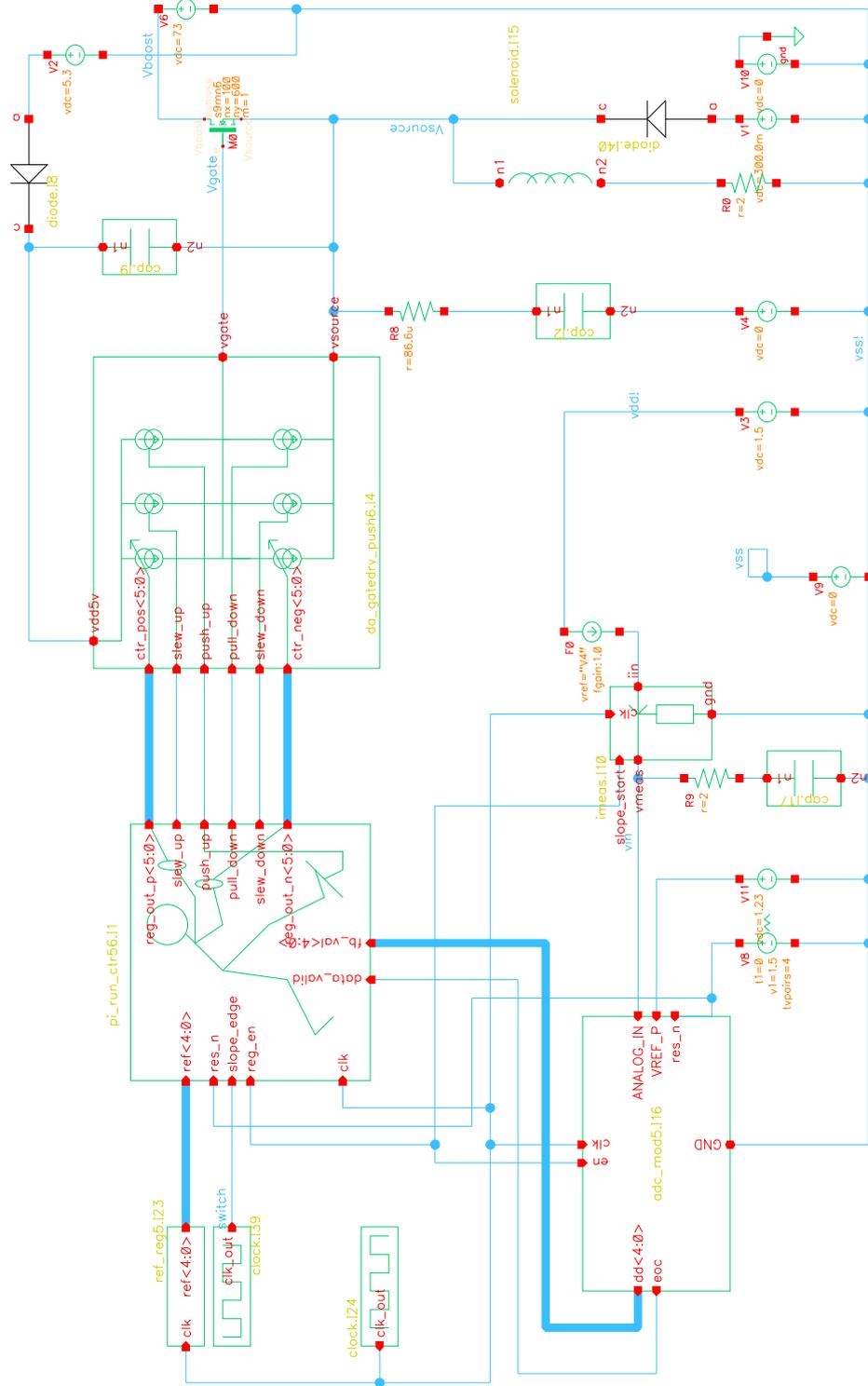


Abb. .1: Toplevel Schematic der digitalen Regelung

B VHDL Files der Regelung

Messblock:

```

1
2 library ieee, std;
3 use ieee.std_logic_1164.all;
4 use ieee.electrical_systems.all;
5 use ieee.numeric_std.all;
6
7 entity imeas is
8     generic( sensit: real := 0.2;
9             ifnom : real := 730.0e-6;
10            Cgate : real := 420.0e-15);
11 port(terminal Iin,gnd : electrical;
12      terminal vmeas: electrical;
13      signal clk : in std_logic;
14      signal slope_start : out std_logic);
15 end imeas;
16
17 architecture behav of imeas is
18     quantity vfb across ifb through Iin to gnd;
19     quantity v1 across i1 through vmeas to gnd;
20
21     constant imax : real := ifnom * (1.0+sensit);
22     constant imin : real := ifnom * (1.0-sensit);
23     constant Rmeas : real := 1.2/(imax-imin);
24     constant hub : real := imax*Rmeas-1.2;
25     constant ILSE : real := Cgate*(imax-imin)/(1.0e-12*(2**5-1));
26     constant threshold : real := 100.0e-6;
27     begin -- behav
28
29         ifb == vfb/10.0;
30
31         break on ifb 'above(imax);
32         break on ifb 'above(imin);
33         break on ifb 'above(-imax);
34         break on ifb 'above(-imin);
35
36         if ifb >= imax use
37             v1 == 1.23;
38         elsif ifb < imax and ifb >= imin use
39             v1 == ifb*Rmeas -hub;
40         elsif ifb < imin and ifb > -imin use
41             V1 == 0.0;
42         elsif ifb <= -imin and ifb >= -imax use
43             v1 == abs(ifb)*Rmeas -hub;
44         else
45             v1 ==1.23;
46         end use;
47
48
49     thres: process(clk)
50
51     begin
52         if clk'event and clk='0' then
53             if abs(ifb) > threshold then
54                 slope_start <= '1';
55             else
56                 slope_start <= '0';
57             end if;
58         end if;
59
60     end process;
61     end behav;

```

Variable Stromquelle P:

```
1  library ieee, std;
2  use ieee.std_logic_1164.all;
3  use ieee.electrical_systems.all;
4
5  entity bias_high_v5 is
6      generic( bits :integer := 5;
7              ibias_lsb :real := 125.0e-6);
8      port (
9          terminal p, m : electrical;
10         ctr : in std_logic_vector(4 downto 0));
11
12 end bias_high_v5;
13
14
15 library ieee, std;
16
17 use ieee.std_logic_1164.all;
18 use ieee.electrical_systems.all;
19 use ieee.numeric_std.all;
20
21 architecture behav of bias_high_v5 is
22
23     quantity vb across ib through p to m;
24     signal ctrl : integer := 0;
25     constant lsb : integer := 2**bits;
26     signal ibias : real := 0.0;
27
28 begin -- behav
29
30     p1:process(ctr)
31     begin
32
33         ibias <= real(ibias_lsb) * real(to_integer(unsigned(ctr)));
34         if to_integer(unsigned(ctr)) = 0 then
35             ctrl<=0 after 1 ns;
36         else
37             ctrl<=1;
38         end if;
39
40     end process;
41
42     break on ib 'above(ibias);
43     break on ctrl;
44
45     if ctrl=1 and not ib 'above(ibias) use
46         ib==vb/(0.8/ibias);
47     elsif ctrl=1 and ib 'above(ibias) use
48         ib==ibias+vb/1.0e10;
49     else
50         ib==vb/1.0e10;
51     end use;
52
53 end behav;
```

Variable Stromquelle N:

```
1  library ieee, std; use
2  ieee.std_logic_1164.all; use ieee.electrical_systems.all;
3
4  entity bias_low_v6 is
5      generic( bits :integer := 6;
6              ibias_lsb :real := 125.0e-6);
7      port (
8          terminal p, m : electrical;
9          ctr : in std_logic_vector(5 downto 0));
```

```

10
11 end bias_low_v6;
12
13 library ieee, std; use ieee.std_logic_1164.all; use
14 ieee.electrical_systems.all; use ieee.numeric_std.all;
15
16 architecture behav of bias_low_v6 is
17
18     quantity vb across ib through p to m;
19     signal ctrl : integer := 0;
20     constant lsb : integer := 2**bits;
21     signal ibias : real := 0.0;
22
23 begin -- behav
24
25     p1:process(ctrl)
26     begin
27
28         ibias <= real(ibias_lsb) * real(to_integer(unsigned(ctrl)));
29         if to_integer(unsigned(ctrl)) = 0 then
30             ctrl<=0 after 1 ns;
31         else
32             ctrl<=1;
33         end if;
34
35     end process;
36
37     break on ib 'above(ibias);
38     break on ctrl;
39
40     if ctrl=1 and not ib 'above(ibias) use
41         ib==vb/(0.75/ibias);
42     elsif ctrl=1 and ib 'above(ibias) use
43         ib==ibias+vb/1.0e10;
44     else
45         ib==vb/1.0e10;
46     end use;
47
48 end behav;

```

Konstante Stromquelle P:

```

1 Library ieee, std;
2
3 use ieee.electrical_systems.all; use ieee.std_logic_1164.all; use
4 ieee.math_real.all;
5
6 entity bias_high is
7     generic(ibias : real);
8     port(
9         terminal p, m : electrical;
10        signal en : in std_logic);
11 end bias_high;
12
13 library ieee, std;
14
15 use ieee.electrical_systems.all; use ieee.std_logic_1164.all; use
16 ieee.math_real.all;
17
18
19 architecture behav of bias_high is
20
21     quantity vb across ib through p to m;
22     signal ctrl : integer := 0;
23
24 begin -- behav
25

```

```
26  p1:process(en)
27  begin
28    if en='1' then
29      ctrl<=1 after 1 ns;
30    else
31      ctrl<=0;
32    end if;
33  end process;
34
35  break on ib 'above(ibias);
36  break on ctrl;
37
38  if ctrl=1 and not ib 'above(ibias) use
39    ib==vb/(0.8/ibias);
40  elsif ctrl=1 and ib 'above(ibias) use
41    ib==ibias+vb/1.0e10;
42  else
43    ib==vb/1.0e10;
44  end use;
45
46  end behav;
```

Konstante Stromquelle N:

```
1  library ieee, std; use ieee.std_logic_1164.all; use
2  ieee.electrical_systems.all;
3
4
5  entity bias_low is
6    generic(ibias : in real);
7    port (
8      terminal p, m : electrical;
9      signal en : in std_logic);
10  end bias_low;
11
12  library ieee; use ieee.std_logic_1164.all; use
13  ieee.electrical_systems.all;
14
15
16  architecture behav of bias_low is
17
18    quantity vb across ib through p to m;
19    signal ctrl : integer := 0;
20
21  begin
22
23    p1:process(en)
24    begin
25      if en='0' then
26        ctrl<=1;
27      else
28        ctrl<=0;
29      end if;
30    end process;
31
32    break on ib 'above(ibias);
33    break on ctrl;
34
35    if ctrl=1 and not ib 'above(ibias) use
36      ib==vb/(0.75/ibias);
37  elsif ctrl=1 and ib 'above(ibias) use
38    ib==ibias+vb/1.0e10;
39  else
40    ib==vb/1.0e10;
41  end use;
42
43  end behav;
```

Gate Driver:

```

1  library ieee, std;
2  use ieee.std_logic_1164.all; use ieee.electrical_systems.all;
3
4  entity da_gatedrv_push6 is
5
6      generic( ILSB : real := 10.0e-6;
7              push_current : real := 100.0e-6;
8              slope_current : real := 100.0e-6;
9              pull_current : real := 100.0e-6;
10             N : integer := 6);
11
12     port(
13         terminal VDD5V, Vgate, Vsource : electrical;
14         signal ctr_pos : IN std_logic_vector(5 downto 0 );
15         signal ctr_neg : IN std_logic_vector(5 downto 0);
16         signal slew_up : IN std_logic;
17         signal slew_down : IN std_logic;
18         signal push_up : in std_logic;
19         signal pull_down : in std_logic
20     );
21 end da_gatedrv_push6;
22
23 library ieee, std; use ieee.std_logic_1164.all; use
24 ieee.electrical_systems.all;
25
26
27 architecture behav of da_gatedrv_push6 is begin
28
29     ctrpos : entity work.bias_high_v6(behav) generic map (bits=>N, ibias_lsb=>ILSB)
30     port map (p=>VDD5V, m=>Vgate, ctr=>ctr_pos );
31
32     ctrlneg : entity work.bias_low_v6(behav) generic map (bits=>N, ibias_lsb=>ILSB)
33     port map (p=>Vgate, m => Vsource, ctr => ctr_neg );
34
35     slup : entity work.bias_high(behav) generic map (ibias => slope_current)
36     port map (p =>VDD5V, m =>Vgate, en =>slew_up );
37
38     sldwn : entity work.bias_low(behav) generic map (ibias => slope_current)
39     port map (p =>Vgate, m => Vsource, en =>slew_down );
40
41     pushup : entity work.bias_high(behav) generic map (ibias => push_current)
42     port map (p =>VDD5V, m =>Vgate, en =>push_up );
43
44     pulldwn : entity work.bias_low(behav) generic map (ibias => pull_current)
45     port map (p =>Vgate, m =>Vsource, en =>pull_down );
46
47 end behav;

```

clock Signal:

```

1  library ieee;
2
3  use ieee.std_logic_1164.all;
4
5  entity clock is
6      generic(freq :      real := 1.0e6);    -- clock frequency
7      port(signal clk_out : out std_logic);  -- clock output
8  end entity;
9
10 library ieee;
11

```

```
12 use ieee.std_logic_1164.all;
13
14
15 architecture behav of clock is
16     constant periode : time := 1 sec/freq;
17     signal clk_int : std_logic := '0';
18 begin
19
20     clock_gen: process(clk_int)
21     begin
22         if clk_int='0' then
23             clk_int<='1' after periode/2, '0' after periode;
24         end if;
25         clk_out <= clk_int;
26     end process;
27
28 end;
```

Referenz Register:

```
1  library ieee; use ieee.std_logic_1164.all; use
2  ieee.numeric_std.all; use ieee.math_real.all;
3
4  entity ref_reg5 is
5      generic (
6          n : integer := 5;
7          value : integer :=0 );
8      port ( signal clk : in std_logic;
9             signal ref : out std_logic_vector(4 downto 0));
10 end ref_reg5;
11
12 library ieee; use ieee.std_logic_1164.all; use
13 ieee.numeric_std.all; use ieee.math_real.all;
14
15
16 architecture behav of ref_reg5 is begin
17
18     p1:process(clk)
19
20     begin
21         if clk'event and clk='0' then
22             ref <= std_logic_vector(to_unsigned(value,n));
23         end if;
24     end process;
25
26 end behav;
```

Flash ADC :

```
1  library IEEE; use IEEE.math_real.all; use
2  IEEE.electrical_systems.all; USE IEEE.STD_LOGIC_1164.ALL; use
3  IEEE.numeric_std.all;
4
5  entity flash_run5 is
6      port (
7          terminal vin,vip: Electrical;
8          signal clk : in std_logic;
9          signal en : in std_logic;
10         signal eoc : out std_logic;
11         signal bb : out std_logic_vector(4 downto 0);
12         signal init_en : out std_logic;
13         signal res_n : in std_logic
14     );
15 end flash_run5;
16
```

```

17 -----
18 -- flash_run4.vhd -- Description: -- Author:          Janschitz
19 Johannes
20
21 -- Ideal Flash 5 Bit  ADC
22
23 library IEEE; use IEEE.math_real.all; use
24 IEEE.electrical_systems.all; USE IEEE.STD_LOGIC_1164.ALL; use
25 ieee.numeric_std.all;
26
27
28 architecture behav of flash_run5 is
29     quantity q_vin across vin;
30     quantity q_vip across vip;
31
32     constant Dmax : unsigned(4 downto 0) := (others => '1');
33     constant bits :real := 5.0;
34     constant vref :real := 1.23;
35
36     signal tmp : std_logic_vector(4 downto 0);
37
38
39 begin
40
41     doadc: process (clk,res_n)
42         variable v_in,lsb,val : real := 0.0;
43         variable ival : integer := 0;
44
45         begin
46             if res_n = '0' then
47                 eoc <= '0';
48                 bb <= std_logic_vector(to_unsigned(0,integer(bits)));
49                 tmp<= std_logic_vector(to_unsigned(0,integer(bits)));
50
51             else
52
53                 eoc <= '0' after 1 ns;
54                 if clk'event and clk = '0' and en='1' then
55
56
57                     v_in := (q_vip - q_vin);
58                     tmp <= std_logic_vector(to_unsigned(
59                         integer(v_in/vref*real(to_integer(dmax))),integer(bits))) after 1 ns;
60                     bb <= std_logic_vector(to_unsigned(
61                         integer(v_in/vref*real(to_integer(dmax))),integer(bits))) after 1 ns;
62                     eoc <= '1' after 1 ns;
63                     elsif en = '0' then
64
65                         tmp <= std_logic_vector(to_unsigned(0,integer(bits)));
66                         bb <= std_logic_vector(to_unsigned(0,integer(bits)));
67
68                     end if;
69                 end if;
70             end process doadc;
71
72     doiniten: process (clk,tmp,res_n)
73         -- variable init_status : integer := 0;
74
75         begin
76
77             if res_n = '0' then
78                 -- init_status :=0;
79                 init_en <= '0';
80             else
81
82                 if to_integer(unsigned(tmp)) > 0 then
83                     init_en <= '1' ;
84

```

6 Zusammenfassung und Ausblick

```
85     elsif clk'event and clk = '0' then
86         init_en <= '0' ;
87
88         end if;
89
90     end if;
91
92
93 end process;
94
95
96 end architecture behav;
```

Solenoid:

```
1  library ieee;
2
3  use ieee.electrical_systems.all;
4
5  entity solenoid is
6      generic(L : real := 10.0e-3);
7      port (terminal n1,n2 : electrical);
8  end;
9  library ieee;
10 use ieee.electrical_systems.all;
11
12 architecture behav of solenoid is
13     quantity v across i through n1 to n2;
14 begin
15     v == L * i'dot;
16 end;
```

PI Controller:

```
1  \begin
2  library ieee, std; use ieee.std_logic_1164.all; use
3  ieee.electrical_systems.all; entity pi_run_ctr56 is
4      generic(n          : integer := 5; -- bit width
5              t1         : real    := 0.0; -- factor for delay system
6              Tn         : real    := 6.25e-9; -- factor for I system
7              Tv         : real    := 1.0e-9; -- factor for D system
8              k_p        : real    := 2.0; -- factor for P system
9              fa         : real    := 50.0e6);
10 port (
11     signal reg_out_p : out std_logic_vector(5 downto 0);
12     signal reg_out_n : out std_logic_vector(5 downto 0);
13     signal slew_up   : out std_logic;
14     signal slew_down : out std_logic;
15     signal push_up   : out std_logic;
16     signal pull_down : out std_logic;
17     signal clk       : in  std_logic;
18     signal res_n     : in  std_logic;
19     signal reg_en    : in  STD_LOGIC;
20     signal slope_edge : in  std_logic;
21     signal ref       : in  std_logic_vector(4 downto 0);
22     signal fb_val    : in  std_logic_vector(4 downto 0);
23     signal data_valid : in  std_logic);
24 end pi_run_ctr56;
25
26 library ieee, std; use ieee.std_logic_1164.all; use
27 ieee.std_logic_arith.all; use ieee.math_real.all;
28
29
30 architecture behav of pi_run_ctr56 is
31
```

```

32  signal clk_int : std_logic;
33  signal en_sld : std_logic;
34
35  signal count : integer:= 0;
36  signal neg_vz : integer := 0;
37  signal reg_on : std_logic;
38  constant Bitdrv : integer := 6;          -- Auflsung des Gate Treibers
39  constant BitADC : integer := n;        -- Auflsung ADC
40  constant drvmax : real := 2.0**Bitdrv;
41
42  signal sreg0 : std_logic_vector(BitADC-1 downto 0);
43  signal sreg1 : std_logic_vector(BitADC-1 downto 0); -- input regs for measurement
44  signal sreg2 : std_logic_vector(BitADC-1 downto 0);
45  signal sreg3 : std_logic_vector(BitADC-1 downto 0);
46  signal sreg4 : std_logic_vector(BitADC-1 downto 0);
47  signal sreg5 : std_logic_vector(BitADC-1 downto 0);
48
49  signal oreg0 : std_logic_vector(n downto 0); -- result registers for timed
50  signal oreg1 : std_logic_vector(n downto 0);
51  signal oreg2 : std_logic_vector(n downto 0);
52  signal oreg3 : std_logic_vector(n downto 0);
53  signal oreg4 : std_logic_vector(n downto 0);
54  signal oreg5 : std_logic_vector(n downto 0);
55
56  signal oreg6 : std_logic_vector(n downto 0);
57  signal oreg7 : std_logic_vector(n downto 0);
58  signal oreg8 : std_logic_vector(n downto 0);
59  signal oreg9 : std_logic_vector(n downto 0);
60  signal oreg10 : std_logic_vector(n downto 0);
61  signal oreg11 : std_logic_vector(n downto 0);
62
63
64
65  begin
66
67  takt_int:process(clk,res_n)
68    variable count : integer := 0;
69  begin
70    if res_n='0' then
71      clk_int<='1';
72      count:=0;
73    else
74      -- elsif clk 'eve6nt and clk='0' then
75      if clk 'event and clk='0' then
76        -- clk_int <= '0' after 3 ns, '1' after 13 ns ;
77        clk_int <= '0' after 2 ns;
78      elsif clk'event and clk='1' then
79        clk_int <= '1' after 2 ns;
80      end if;
81
82    end if;
83  end process;
84
85
86  grap_values :process(clk,data_valid,res_n,slope_edge)
87    variable scount : integer ;
88  begin
89
90    if res_n = '0' or slope_edge'event then
91      sreg0 <= std_logic_vector(conv_unsigned(0,n));
92      sreg1 <= std_logic_vector(conv_unsigned(0,n));
93      sreg2 <= std_logic_vector(conv_unsigned(0,n));
94      sreg3 <= std_logic_vector(conv_unsigned(0,n));
95      sreg4 <= std_logic_vector(conv_unsigned(0,n));
96      sreg5 <= std_logic_vector(conv_unsigned(0,n));
97      scount := 0;
98
99    elsif reg_en = '0' then

```

```

100     --   scount := 0;
101
102     elsif data_valid'event and data_valid = '1' and reg_en = '1' then
103
104
105         if scount <= 6 then
106
107
108             case scount is
109                 when 0 => sreg5 <= fb_val;
110                 when 1 => sreg4 <= fb_val;
111                 when 2 => sreg3 <= fb_val;
112                 when 3 => sreg2 <= fb_val;
113                 when 4 => sreg1 <= fb_val;
114                 when 5 => sreg0 <= fb_val;
115                 when others => break ;
116             end case;
117
118         -- sreg0 <= fb_val ;
119         -- sreg1 <= sreg0;
120         -- sreg2 <= sreg1;
121         -- sreg3 <= sreg2 ;
122         -- sreg4 <= sreg3;
123         -- sreg5 <= sreg4;
124
125         end if;
126         scount := scount +1;
127
128     end if;
129
130 end process;
131
132
133 calc : process (clk,res_n,reg_en)
134
135     variable xk_0 : real := 0.0;
136     variable xk_1 : real := 0.0;
137     variable xk_2 : real := 0.0;
138     variable xk_3 : real := 0.0;
139     variable xk_4 : real := 0.0;
140     variable xk_5 : real := 0.0;
141
142
143     variable zk_0 : real := 0.0;
144     variable zk_1 : real := 0.0;
145     variable zk_2 : real := 0.0;
146     variable zk_3 : real := 0.0;
147     variable zk_4 : real := 0.0;
148     variable zk_5 : real := 0.0;
149
150     variable c0_0 : real := 0.0;           -- variable coefficient
151     variable c1_0 : real := 0.0;
152     variable c2_0 : real := 0.0;
153     variable c3_0 : real := 0.0;
154     variable c4_0 : real := 0.0;
155     variable c5_0 : real := 0.0;
156
157     variable c0_1 : real := 0.0;
158     variable c1_1 : real := 0.0;
159     variable c2_1 : real := 0.0;
160     variable c3_1 : real := 0.0;
161     variable c4_1 : real := 0.0;
162     variable c5_1 : real := 0.0;
163
164
165     variable ek0 : real := 0.0;           -- error
166     variable ek1 : real := 0.0;
167     variable ek2 : real := 0.0;

```

```

168     variable ek3 : real := 0.0;
169     variable ek4 : real := 0.0;
170     variable ek5 : real := 0.0;
171
172
173     variable e_0 : real := 0.0;           -- error
174     variable e_1 : real := 0.0;
175     variable e_2 : real := 0.0;
176     variable e_3 : real := 0.0;
177     variable e_4 : real := 0.0;
178     variable e_5 : real := 0.0;
179
180     variable ek_10 : real := 0.0;        -- error
181     variable ek_11 : real := 0.0;
182     variable ek_12 : real := 0.0;
183     variable ek_13 : real := 0.0;
184     variable ek_14 : real := 0.0;
185     variable ek_15 : real := 0.0;
186
187     variable Tk_hu0 : real := 0.0;       -- error
188     variable Tk_hu1 : real := 0.0;
189     variable Tk_hu2 : real := 0.0;
190     variable Tk_hu3 : real := 0.0;
191     variable Tk_hu4 : real := 0.0;
192     variable Tk_hu5 : real := 0.0;
193
194     variable Tk_hd0 : real := 0.0;      -- error
195     variable Tk_hd1 : real := 0.0;
196     variable Tk_hd2 : real := 0.0;
197     variable Tk_hd3 : real := 0.0;
198     variable Tk_hd4 : real := 0.0;
199     variable Tk_hd5 : real := 0.0;
200
201
202     variable ref_reg : real := 0.0;      -- reference value for slope
203     variable ref_reg2 : real := 0.0;
204
205
206     variable TA : real := 1.0/fa;
207
208
209     variable TAoTN : real := TA/TN;
210     variable TVoTA : real := TV/TA;
211
212
213     variable calc_en : integer := 0;
214
215     variable KPR : real := k_p;
216
217     variable count_val : integer := 0;
218 begin
219
220     if res_n='0' then
221
222
223         xk_0 := 0.0;
224         xk_1 := 0.0;
225         xk_2 := 0.0;
226         xk_3 := 0.0;
227         xk_4 := 0.0;
228         xk_5 := 0.0;
229
230
231         zk_0 := 0.0;
232         zk_1 := 0.0;
233         zk_2 := 0.0;
234         zk_3 := 0.0;
235         zk_4 := 0.0;

```

```
236     zk_5 := 0.0;
237
238     c0_0 := 0.0;
239     c1_0 := 0.0;
240     c2_0 := 0.0;
241     c3_0 := 0.0;
242     c4_0 := 0.0;
243     c5_0 := 0.0;
244
245     c0_1 := 0.0;
246     c1_1 := 0.0;
247     c2_1 := 0.0;
248     c3_1 := 0.0;
249     c4_1 := 0.0;
250     c5_1 := 0.0;
251
252     ek_10 := 0.0;
253     ek_11 := 0.0;
254     ek_12 := 0.0;
255     ek_13 := 0.0;
256     ek_14 := 0.0;
257     ek_15 := 0.0;
258
259     e_0 := 0.0;
260     e_1 := 0.0;
261     e_2 := 0.0;
262     e_3 := 0.0;
263     e_4 := 0.0;
264     e_5 := 0.0;
265
266     calc_en := 0;
267
268     ref_reg := 0.0;
269     ref_reg2 := 0.0;
270
271     oreg0 <= "000000";
272     oreg1 <= "000000";
273     oreg2 <= "000000";
274     oreg3 <= "000000";
275     oreg4 <= "000000";
276     oreg5 <= "000000";
277     oreg6 <= "000000";
278     oreg7 <= "000000";
279     oreg8 <= "000000";
280     oreg9 <= "000000";
281     oreg10 <= "000000";
282     oreg11 <= "000000";
283
284     elsif reg_en'event and reg_en = '0' then
285         calc_en := 0;
286
287     else
288
289         if clk'event and clk = '1' and reg_en = '0' and calc_en = 0 then
290
291             c0_1 := zk_0;
292             c1_1 := zk_1;
293             c2_1 := zk_2;
294             c3_1 := zk_3;
295             c4_1 := zk_4;
296             c5_1 := zk_5;
297
298             zk_0 := c0_0;
299             zk_1 := c1_0;
300             zk_2 := c2_0;
301             zk_3 := c3_0;
302             zk_4 := c4_0;
303
```

```

304      zk_5 := c5_0;
305
306
307      ek_10 := e_0;
308      ek_11 := e_1;
309      ek_12 := e_2;
310      ek_13 := e_3;
311      ek_14 := e_4;
312      ek_15 := e_5;
313
314      e_0 := ek0;
315      e_1 := ek1;
316      e_2 := ek2;
317      e_3 := ek3;
318      e_4 := ek4;
319      e_5 := ek5;
320
321
322      xk_0:=real(conv_integer(unsigned( sreg0)));
323      xk_1:=real(conv_integer(unsigned( sreg1)));
324      xk_2:=real(conv_integer(unsigned( sreg2)));
325      xk_3:=real(conv_integer(unsigned( sreg3)));
326      xk_4:=real(conv_integer(unsigned( sreg4)));
327      xk_5:=real(conv_integer(unsigned( sreg5)));
328
329      ref_reg :=real(conv_integer(unsigned( ref )));
330      ref_reg2 :=real(conv_integer(unsigned( ref )));
331
332
333      ek0 :=  ref_reg - xk_0 ;
334      ek1 :=  ref_reg - xk_1 ;
335      ek2 :=  ref_reg - xk_2 ;
336      ek3 :=  ref_reg - xk_3 ;
337      ek4 :=  ref_reg - xk_4 ;
338      ek5 :=  ref_reg - xk_5 ;
339
340
341      c0_0 := c0_1 + KPR*TAoTN*ek_10 + KPR*(ek0 - ek_10);
342      c1_0 := c1_1 + KPR*TAoTN*ek_11 + KPR*(ek1 - ek_11);
343      c2_0 := c2_1 + KPR*TAoTN*ek_12 + KPR*(ek2 - ek_12);
344      c3_0 := c3_1 + KPR*TAoTN*ek_13 + KPR*(ek3 - ek_13);
345      c4_0 := c4_1 + KPR*TAoTN*ek_14 + KPR*(ek4 - ek_14);
346      c5_0 := c5_1 + KPR*TAoTN*ek_15 + KPR*(ek5 - ek_15);
347
348
349
350      if slope_edge= '0' then
351
352          Tk_hu0 := (ref_reg - ek0)/ref_reg;           -- Stepresponse
353          Tk_hu1 := (ref_reg - ek1)/ref_reg;
354          Tk_hu2 := (ref_reg - ek2)/ref_reg;
355          Tk_hu3 := (ref_reg - ek3)/ref_reg;
356          Tk_hu4 := (ref_reg - ek4)/ref_reg;
357          Tk_hu5 := (ref_reg - ek5)/ref_reg;
358
359          if c0_0 < 0.0 then
360              c0_0 :=0.0;
361          end if;
362          if c1_0 < 0.0 then
363              c1_0 :=0.0;
364          end if;
365          if c2_0 < 0.0 then
366              c2_0 :=0.0;
367          end if;
368          if c3_0 < 0.0 then
369              c3_0 :=0.0;
370          end if;
371          if c4_0 < 0.0 then

```

```

372         c4_0 :=0.0;
373     end if;
374     if c5_0 < 0.0 then
375         c5_0 :=0.0;
376     end if;
377
378
379     if c0_0 > drvmax then
380         c0_0 :=drvmax -1.0;
381     end if;
382     if c1_0 > drvmax then
383         c1_0 :=drvmax -1.0;
384     end if;
385     if c2_0 > drvmax then
386         c2_0 :=drvmax -1.0;
387     end if;
388     if c3_0 > drvmax then
389         c3_0 :=drvmax -1.0;
390     end if;
391     if c4_0 > drvmax then
392         c4_0 :=drvmax -1.0;
393     end if;
394     if c5_0 > drvmax then
395         c5_0 :=drvmax -1.0;
396     end if;
397
398     oreg0 <= std_logic_vector(conv_unsigned(integer(c0_0),bitdrv));
399     oreg1 <= std_logic_vector(conv_unsigned(integer(c1_0),bitdrv));
400     oreg2 <= std_logic_vector(conv_unsigned(integer(c2_0),bitdrv));
401     oreg3 <= std_logic_vector(conv_unsigned(integer(c3_0),bitdrv));
402     oreg4 <= std_logic_vector(conv_unsigned(integer(c4_0),bitdrv));
403     oreg5 <= std_logic_vector(conv_unsigned(integer(c5_0),bitdrv));
404
405     else
406
407         Tk_hd0 := (ref_reg - ek0)/ref_reg;           -- Stepresponse
408         Tk_hd1 := (ref_reg - ek1)/ref_reg;
409         Tk_hd2 := (ref_reg - ek2)/ref_reg;
410         Tk_hd3 := (ref_reg - ek3)/ref_reg;
411         Tk_hd4 := (ref_reg - ek4)/ref_reg;
412         Tk_hd5 := (ref_reg - ek5)/ref_reg;
413
414
415     if c0_0 < 0.0 then
416         c0_0 :=0.0;
417     end if;
418     if c1_0 < 0.0 then
419         c1_0 :=0.0;
420     end if;
421     if c2_0 < 0.0 then
422         c2_0 :=0.0;
423     end if;
424     if c3_0 < 0.0 then
425         c3_0 :=0.0;
426     end if;
427     if c4_0 < 0.0 then
428         c4_0 :=0.0;
429     end if;
430     if c5_0 < 0.0 then
431         c5_0 :=0.0;
432     end if;
433
434
435     if c0_0 > drvmax then
436         c0_0 :=drvmax -1.0;
437     end if;
438     if c1_0 > drvmax then
439         c1_0 :=drvmax -1.0;

```

```

440         end if;
441         if c2_0 > drvmax then
442             c2_0 :=drvmax -1.0;
443         end if;
444         if c3_0 > drvmax then
445             c3_0 :=drvmax -1.0;
446         end if;
447         if c4_0 > drvmax then
448             c4_0 :=drvmax -1.0;
449         end if;
450         if c5_0 > drvmax then
451             c5_0 :=drvmax -1.0;
452         end if;
453
454
455         oreg6 <= std_logic_vector(conv_unsigned(integer(c0_0),n+1));
456         oreg7 <= std_logic_vector(conv_unsigned(integer(c1_0),n+1));
457         oreg8 <= std_logic_vector(conv_unsigned(integer(c2_0),n+1));
458         oreg9 <= std_logic_vector(conv_unsigned(integer(c3_0),n+1));
459         oreg10 <= std_logic_vector(conv_unsigned(integer(c4_0),n+1));
460         oreg11 <= std_logic_vector(conv_unsigned(integer(c5_0),n+1));
461
462         end if;
463         calc_en := 1;
464
465     end if;
466 end if;
467 end process;
468
469
470 move_out:process(clk,res_n,slope_edge,reg_en)
471     variable out_count : integer := 0;
472     variable schcount : integer := 0;
473 begin
474
475     if res_n='0' then
476         reg_out_p<= std_logic_vector(conv_unsigned(0,Bitdrv));
477         reg_out_n<= std_logic_vector(conv_unsigned(0,Bitdrv));
478
479     elsif slope_edge'event then
480
481         out_count := 0;
482         reg_out_p<= std_logic_vector(conv_unsigned(0,Bitdrv));
483         reg_out_n<= std_logic_vector(conv_unsigned(0,Bitdrv));
484
485     elsif clk'event and clk='1' then
486
487
488
489         if slope_edge = '0' then
490
491
492             case out_count is
493                 when 0 => reg_out_p <= oreg5;
494                 when 1 => reg_out_p <= oreg4;
495                 when 2 => reg_out_p <= oreg3;
496                 when 3 => reg_out_p <= oreg2;
497                 when 4 => reg_out_p <= oreg1;
498                 when 5 => reg_out_p <= oreg0;
499                 when 6 => reg_out_p <= oreg0;
500
501                 when others => reg_out_p <= "000000";
502             end case;
503
504             if reg_en = '1' then
505                 out_count := out_count +1;
506             end if;
507

```

```
508
509     else
510
511         case out_count is
512             when 0 => reg_out_n <= oreg11;
513             when 1 => reg_out_n <= oreg10;
514             when 2 => reg_out_n <= oreg9;
515             when 3 => reg_out_n <= oreg8;
516             when 4 => reg_out_n <= oreg7;
517             when 5 => reg_out_n <= oreg6;
518
519             when others => reg_out_n <= "000000";
520         end case;
521
522         if reg_en= '1' then
523             out_count := out_count +1;
524         end if;
525     end if;
526
527
528 end if;
529 end process;
530
531
532 pstate:process(res_n,slope_edge)
533
534 begin
535
536     if res_n='0' then
537
538         slew_up<= '0';
539         slew_down<= '0';
540         push_up<= '0';
541         pull_down<='1';
542
543     end if;
544
545
546     if slope_edge= '0' and slope_edge'event then
547
548         slew_up  <= '0', '1' after 4 ns;
549         slew_down <= '0', '1' after 2 ns;
550         push_up  <= '1', '0' after 10 ns ;
551         pull_down <= '1';
552     else
553         slew_up  <= '1' , '0' after 2 ns;
554         slew_down <= '1' , '0' after 4 ns;
555         pull_down <= '0', '1' after 10 ns ;
556         push_up  <= '0';
557     end if;
558
559
560 end process;
561
562 end;
```

Literaturverzeichnis

- [AH02] ALLEN, Phillip E. ; HOLBERG, Douglas R.: *CMOS Analog Circuit Design*. 2. Auflage. Oxford University Press, Inc., 2002. – ISBN 0–19–511644–5
- [APT03] ASHENDEN, Peter J. ; PETERSON, Gregory D. ; TEEGARDEN, Darrell A.: *The system designer's guide to VHDL-AMS*. Morgan Kaufmann, 2003. – ISBN 9781558607491
- [Aue09] AUER, Mario: *Analog Integrated Circuit Design and Simulation I/II, Skriptum*. Institut für Elektronik, TU Graz, 2009
- [Bak05] BAKER, R. J.: *CMOS - Circuit Design, Layout and Simulation*. 2. Auflage. John Wiley & Sons, Inc., 2005. – ISBN 0–471–70055–X
- [Bar99] BARTSCH, Hans-Jochen: *Taschenbuch Mathematischer Formeln*. 18th ed. Fachbuchverlag Leipzig, 1999. – ISBN 3–446–21048–2
- [CH92] CHATTERTON, P.A. ; HOULDEN, M.A.: *EMC Electromagnetic Theory to Practical Design*. 1. Auflage. John Wiley & Sons Ltd, Baffins Lane, Chichester, West Sussex PO19 1UD, England, 1992. – ISBN 0–471–92878–X
- [DB06] DORF, Richard C. ; BISHOP, Robert H.: *Moderne Regelungssysteme*. 10. überarbeitete Auflage. Pearson Education Deutschland GmbH, ©, 2006. – ISBN 3–8273–7162–7
- [DD03] DEEPIKA DEVARAJAN, S. Knorr B. Stanton: Multi-domain modeling and simulation of a linear actuation system, 2003, S. 76 – 81
- [Dra06] DRAXELMAYR, Dieter: *Basics of Analog Circuit Design, Kursunterlagen*. Infineon Technologies AG., 2006
- [Gal93] GALLUZZO, Melito M. Berverde G. Musumeci S. Raciti A. Testa A. A.: Switching characteristic improvement of modern gate controlled devices, 1993, S. 374 –379 vol.2
- [GHLM01] GRAY, Paul R. ; HURST, Paul J. ; LEWIS, Stephen H. ; MEYER, Robert G.:

- Analysis and Design of Analog Integrated Circuits*. 4. Auflage. John Wiley & Sons, Inc., 2001. – ISBN 0-471-32168-0
- [GHS91] GAUSCH, F. ; HOFER, A. ; SCHLACHER, K.: *Digitale Regelkreise*. Institut für Regelungstechnik - Technische Universität Graz, 1991
- [Gon05] GONSCHOREK, Karl-Heinz: *EMV für Geräteentwickler und Systemintegratoren*. 1. Auflage. Springer-Verlag Berlin Heidelberg, 2005. – ISBN 978-3-540-23436-4
- [HD04] HORN, Martin ; DOURDOUMAS, Nicolaos: *Regelungstechnik*. 1. Auflage. Pearson Education Deutschland GmbH, 2004. – ISBN 3-8273-7059-0
- [Hof06] HOFFMANN, Kurt: *Systemintegration*. 2. Auflage. Oldenbourg Verlag, 2006. – ISBN 3-486-57894-2
- [IK04] IVANOV, S. C. ; KOSTOVA, E. K.: *Active Gate Drivers for MOSFETS with circuit for dv/dt Control*, 2004
- [Inf08] INFINEON TECHNOLOGIES AG (Hrsg.): *Infineon Datenblatt OptiMOS2 Power-Transistor*. Version: 2008. <http://www.infineon.com>. (IPP16CNE8N)
- [Ise06] ISERMANN, Rolf: *Fahrdynamik-Regelung Modellbildung, Fahrerassistenzsysteme, Mechatronik*. 1. Auflage. LinkFriedr. Vieweg & Sohn Verlag | GWV Fachverlage GmbH, Wiesbaden, 2006. – ISBN 978-3-8348-9049-8
- [JM97] JOHNS, David A. ; MARTIN, Ken: *Analog Integrated Circuit Design*. John Wiley & Sons, Inc., 1997. – ISBN 0-471-14448-7
- [Kag07] KAGERBAUER, Jahns T. J.D.: *Development of an Active dv/dt Control Algorithm for Reducing Inverter Conducted EMI with Minimal Impact on Switching Losses*, 2007. – ISSN 0275-9306, S. 894 -900
- [Kes05] KESTER, Walt: *Data conversion handbook*. Analog Devices ; Walter Kester, editor ; with the technical staff of Analog Devices, 2005. – ISBN 9780750678414 0750678410
- [KMR06] KÜPFMÜLLER, Karl ; MATHIS, Wolfgang ; REIBIGER, Albrecht: *Theoretische Elektrotechnik*. 17. Auflage. Springer-Verlag Berlin Heidelberg, 2006. – ISBN 3-540-29290-X
- [Krü04] KRÜGER, Manfred: *Grundlagen der Kraftfahrzeugelektronik*. 1. Auflage. Carl Hanser Verlag, 2004. – ISBN 3-446-22594-3

- [Kre06] KREYSZIG, Erwin: *Advanced Engineering Mathematics*. 9. Auflage. John Wiley & Sons, Inc., 2006. – ISBN 0-471-72897-7
- [Kut08] KUTTNER, Franz: *AD/DA-Converter, Kursunterlagen*. Infineon Technologies AG., 2008
- [LIN06] LIN CONSORTIUM (Hrsg.): *LIN Specification Package, Version 2.1*. Version: 2006. <http://www.lin-subbus.org>
- [LS94] LAKER, Kenneth R. ; SANSEN, Willy M.: *Design of Analog Integrated Circuits and Systems*. McGraw-Hill, 1994. – ISBN 0-07-113458-1
- [Lut06] LUTZ, Josef: *Halbleiter-Leistungsbaulemente*. 1. Auflage. Springer Verlag, 2006. – ISBN 3-540-34206-0
- [Mit95] MITTER, C.S.: Active Inrush Current Limiting Using MOSFETs. In: *Motorola semiconductor Application Note* (1995), Nr. AN1542
- [Nag97] NAGEL, De Doncker R. A.: Analytical approximations of interference spectra generated by power converters, 1997, S. 1564 –1570 vol.2
- [Nea06] NEACSU, Dorin O.: *Power- Switching Converters - Medium and High Power*. 1. Auflage. CRC Press Taylor & Francis Group, 2006. – ISBN 0-8247-2625-1
- [Oga95] OGATA, Katsuhiko: *Diskrete-Time Control Systems*. 2nd ed. Prentice-Hall, Inc, 1995. – ISBN 0-13-034281-5
- [Pet03] PETERCHEV, Sanders S. A.V.: Quantization resolution and limit cycling in digitally controlled PWM converters. In: *Power Electronics, IEEE Transactions on* 18 (2003), Jan, Nr. 1, S. 301 – 308. <http://dx.doi.org/10.1109/TPEL.2002.807092>. – DOI 10.1109/TPEL.2002.807092. – ISSN 0885-8993
- [Pro01] PRODIC, Maksimovic D. Erickson R. A.: Design and implementation of a digital PWM controller for a high-frequency switching DC-DC power converter, 2001, S. 893 –898 vol.2
- [Rei07] REIF, Konrad: *Automobilelektronik*. 2. Auflage. Friedrich Vieweg & Sohn Verlag, 2007. – ISBN 978-3-8348-0297-2
- [RZ08] REUTER, Manfred ; ZACHER, Serge: *Regelungstechnik für Ingenieure - Analyse, Simulation und Entwurf von Regelkreisen*. 12. korrigierte und erweiterte Auflage. 2008. – ISBN 978-3-8348-0018-3

- [San06] SANSEN, Willy M.: *Analog Design Essentials*. Springer Verlag, 2006. – ISBN 0–387–25746–2
- [Sau99] SAUNDERS, Simon R.: *Antennas and Propagation for Wireless Communication Systems*. 1. Auflage. John Wiley & Sons LTD, 1999. – ISBN 0–471–98609–7
- [Saw02] SAWEZYN, Idir N. Bausiere R. H.: Lowering the drawbacks of slowing down di/dt and dv/dt of insulated gate transistors, 2002. – ISSN 0537–9989, S. 551 – 556
- [Sch05] SCHULZE, Jörg: *Konzepte siliziumbasierter MOS-Bauelemente*. 1. Auflage. Springer-Verlag Berlin Heidelberg, 2005. – ISBN 978–3–540–27547–3
- [SL91] SLOTINE, Jean-Jacques E. ; LI, Weiping: *Applied Nonlinear Control*. 1. Auflage. Prentice-Hall, Inc Pearson Education, 1991. – ISBN 0–13–040890–5
- [Sös05] SÖSER, Peter: *Integrierte Schaltungen, Skriptum*. Institut für Elektronik, TU Graz, 2005
- [Sti94] STIFTINGER, Martin: *Simulation und Modellierung von Hochvolt-DMOS-Transistoren*, Technische Universität Wien Fakultät für Elektrotechnik, Diplomarbeit, August 1994
- [Tre10] TRESCASES, O.: *Switched Mode Power Supplies, Kursunterlagen*. Infineon Technologies AG., 2010
- [Wap07] WAPPIS, Herwig: *Auswirkungen und Einflüsse von automobilen Zuverlässigkeitsanforderungen auf analoge Schaltungskonzepte in hochvoltfähigen Deep-Sub-Micron Technologien*, Technische Universität Graz Institut für Elektronik, Diplomarbeit, September 2007
- [ZKLV05] ZHAO, Wei ; KIM, Byung H. ; LARSON, A.C. ; VOYLES, R.M.: FPGA implementation of closed-loop control system for small-scale robot, 2005, S. 70 –77
- [ZS06] ZIMMERMANN, Werner ; SCHMIDGALL, Ralf: *Bussysteme in der Fahrzeugtechnik Protokolle und Standards*. 2. aktualisierte und erweiterte Auflage. Vieweg+Teubner, 2006. – ISBN 978–3–8348–9188–4