

«DJ» EMI Filter für Schaltnetzteil

Fachbericht

Windisch, 20.04.2019



Hochschule Hochschule für Technik - FHNW

Studiengang Elektro- und Informationstechnik

Auftraggeber Dr. Luca Dalessandro

Betreuer Prof. Dr. Sebastian Gaulocher
Prof. Peter Niklaus
Prof. Dr. Richard Gut
Dr. Anita Gertiser
Pascal Buchschacher

Autoren **Gruppe 1**
Niklaus Schwegler
Lukas von Däniken
Pascal Puschmann
Simon Rohrer
Marco Binder

Version 2.0

Abstract
und im hauptfile

Zusammenfassung

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen	2
2.1	Elektrotechnik	2
2.1.1	2-Tore	2
2.1.2	Kettenmatrix	2
2.1.3	Streuparameter	4
2.1.4	Parasitäre Paramter	6
2.1.5	Einfügedämpfung	6
2.1.6	Aufbau eines EMI-Filters	6
2.1.7	Gleich- Gegentaktschaltung	7
2.2	Programmieren	8
2.2.1	MVC-Struktur	8
3	Umsetzung	9
3.1	Berechnung der Einfügedämpfung	9
4	Software	13
4.1	Übersicht	13
4.2	Model	13
4.3	View	13
4.4	Controller	14
4.5	Trace	14
5	Testkonzept	15
5.1	Aufbau	15
5.2	Validierung	15
5.3	Erwartungen	16
6	Schluss	17
7	Anhang	18
7.1	Testkonzept	18

1 Einleitung

Entwurf Einleitung P2 EMI-Filter Team 1

Gemäss des Lastenhefts wurde der Auftrag erteilt, eine Simulationssoftware zu entwickeln. Diese Simulationssoftware soll die Einfügedämpfung eines EMI-Filters simulieren und grafisch darstellen. EMI-Filter werden üblicherweise in Schaltnetzteile verbaut, um zu verhindern, dass Störungen zurück ins Netz gespeist werden. Netzgeräte können unter Umständen hohe Frequenzen erzeugen, die sich nicht gut mit der Netzfrequenz von 50 Hz vertragen. Der EMI-Filter filtert genau diese hochfrequenten Signale heraus, um zu verhindern, dass andere Geräte, die auch ans Netz angeschlossen werden, nicht davon beeinträchtigt werden.

Um die Einfügedämpfung zu ermitteln, soll das EMI-Filter bezüglich der Gleich- und Gegentaktschaltung untersucht werden. Dies geschieht anhand zweier Funktionen für Gleich- und Gegentaktschaltung. Die Funktionen zeigen die Einfügedämpfung für einen Frequenzbereich bis 30MHz. Die beiden Schaltungen beinhalten die parasitären Parameter der elektrischen Komponenten, sodass eine möglichst wahrheitsgetreue Simulation gemacht werden kann. Die entwickelte Software soll die Einfügedämpfung grafisch darstellen. Die Resultate werden für die Gleich- und Gegentaktschaltung in separaten Funktionen dargestellt. Des Weiteren sollen die elektrischen Komponenten der Schaltungen in der Simulationssoftware variiert werden können.

Damit sichergestellt werden kann, dass die Simulationen mathematisch korrekt sind, werden alle Berechnungen zuerst in MATLAB durchgerechnet. Diese Ergebnisse werden mit Simulationen der Simulationssoftware MPLAB mindi verglichen. Des Weiteren wird überprüft, wie die Schaltung vereinfacht werden kann. Dies erfolgt einerseits durch Symmetrien der Schaltung, was dazu führt, dass Komponenten zusammengefasst werden können. Andererseits auch durch Weglassen aufgrund von vernachlässigbarem Einfluss auf die Simulationen. Die Softwarestruktur orientiert sich am gängigen Prinzip der MVC(Model-View-Control). Diese Strukturierung begünstigt einen modularen Aufbau, was die Software einfach erweiterbar macht und zudem eine unkomplizierte Wartung ermöglicht. Des Weiteren wird die Software anhand des Testkonzepts Modul für Modul getestet.

Im Fokus des Fachberichts befindet sich die Software, da das zu entwickelnde Produkt eine Simulationssoftware ist. Der Fachbericht ist nach dem Top-Down-Prinzip aufgebaut. In einem ersten Schritt wird die Software als Ganzes beschrieben. In den darauffolgenden Kapiteln befinden sich die Dokumentationen der beiden Teile der Software. Die Software wird aufgegliedert in den Teil Benutzeroberfläche und den Teil Ermittlung der Einfügedämpfung. Um den Fachbericht schlank zu gestalten, werden sämtliche theoretische Grundlagen im Anhang platziert. Falls in Kapiteln entsprechende Theorie wichtig ist wird darauf verwiesen.

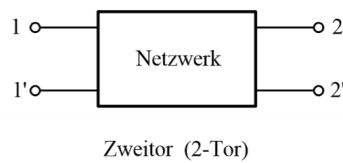
2 Grundlagen

In diesem Kapitel werden die technischen Grundlagen beschrieben, welche benötigt werden um Kapitel 3 und Kapitel 4 zu verstehen.

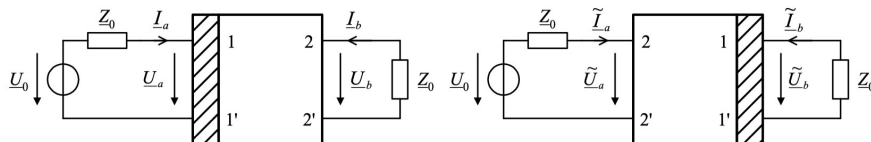
2.1 Elektrotechnik

2.1.1 2-Tore

2-Tore sind Netzwerke mit 2 Klemmenpaaren. Ein Klemmenpaar wird auch als Tor bezeichnet. Diese sind intern aus beliebigen R (ohmscher Widerstand), L (Induktivität), C (Kapazität) und M (Gegeninduktivität) Komponenten aufgebaut. Dabei wird ein Tor als Eingang für ein Elektrisches Signal verwendet. Folglich wird am Anderen Tor das Ausgangssignal abgegriffen. Es wird zwischen aktiven (Signal verstärkend) und passiven (Signal abschwächend) Zweitoren unterschieden. Eine einfache Methode um 2-Tore zu berechnen ist die Kettenmatrix.



Lineare 2-Tore sind zudem Reziprok. Reziprozität bedeutet, dass es egal ist, in welcher Richtung das Netzwerk betrieben wird, solange der Innenwiderstand der Quelle gleich gross ist wie die Lastimpedanz.



Im dargestellten Netzwerk gilt somit

$$\underline{I}_b = \underline{\tilde{I}}_b \quad (2.1)$$

sowie

$$\underline{U}_b = \underline{\tilde{U}}_b \quad (2.2)$$

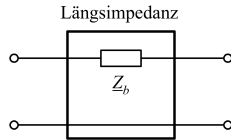
Die Reziprozität des Netzwerks sorgt für eine grosse Rechnerleichterung.

2.1.2 Kettenmatrix

Die Kettenmatrix ist eine Variante, um das Verhalten von 2-Toren zu beschreiben. Andere Varianten sind die Z-Matrix oder die Y-Matrix. Die Kettenmatrix hat jedoch den Vorteil, dass man in Serie geschaltene 2-Tore ohne grossen Aufwand zusammen rechnen kann. Sobald man die

einzelnen Kettenmatrizen gebildet hat und die Schaltung soweit vereinfacht ist, dass nur noch in Serie geschaltene ABCD-Matrizen vorzufinden sind, können diese miteinander multipliziert werden. Das Matrix-Produkt stellt dann die Kettenmatrix der Gesamtschaltung dar. Folgende gängigen Schaltungen helfen die Kettenmatrizen der einzelnen Schaltungsteilen zu bilden

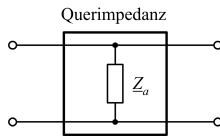
Die Längsimpedanz lässt sich anhand der Kettenmatrix A_L (Formel 2.1) darstellen



$$A_L = \begin{bmatrix} 1 & Z_b \\ 0 & 1 \end{bmatrix} \quad (2.3)$$

Abbildung 2.1: Längsimpedanz
2torTabelle

Die Querimpedanz lässt sich anhand der Kettenmatrix A_Q (Formel 2.2) darstellen



$$A_Q = \begin{bmatrix} 1 & 0 \\ \frac{1}{Z_a} & 1 \end{bmatrix} \quad (2.4)$$

Abbildung 2.2: Querimpedanz
2torTabelle

Sobald die Kettenmatrix einer Schaltung gebildet wurde, kann diese direkt in die Streuparameter umgewandelt werden. Der s_{21} Parameter kann wie in Formel 2.3 beschrieben, durch einsetzen der Kettenmatrix bestimmt werden. Für den Widerstand R_w muss die verwendete Bezugsimpedanz eingesetzt werden.

$$s_{21} = \frac{2}{A_{11} + \frac{A_{12}}{R_w} + A_{21} * R_w + A_{22}} \quad (2.5)$$

Die Indexierung der Kettenmatrix wird in Abbildung 2.4 gezeigt.

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad (2.6)$$

2.1.3 Streuparameter

Der Streuparameter, auch S-Parameter S_{21} gibt den Transmissionsgrad der Wellen an, die vom Tor 1 zum Tor2 übertragen wird. Die S-Parameter sind abhängig von den Bezugswiderständen (Innenwiderstand der Quelle sowie Lastwiderstand). In unserem Fall sind die Bezugswiderstände mit 50Ohm gegeben.

$$IL = |H(j\omega)| = 20 * \log\left(\frac{|U_{20}|}{|U_2|}\right) \quad (2.7)$$

In der Definition kann das Spannungsverhältnis durch den Streuparameter ?? (S-Parameter) S_{21} ersetzt werden 2.14.

$$IL = -20 * \log(|S_{21}|) \quad (2.8)$$

Dieser Parameter beschreibt den Transmissionsgrad des Filters. Die Einfügungsverluste wären auch mit dem Verhältnis von eingehende zu abegegebene Leistung zu berechnen, jedoch eignet sich diese Methode mehr beim messtechnischen bestimmen der Einfügungsverluste.

Die Streuparameter (S-Parameter) werden in der Hochfrequenztechnik verwendet, um das Verhalten von n-Toren zu beschreiben. Bei einem 2-Tor sind vier Streuparameter von nöten um das Verhalten zu beschreiben. Sie beschreiben die Transmission von Tor 1 zu Tor 2, sowie von Tor 2 zu Tor 1. Des weiteren zeigen sie die Reflexion an den Toren auf. Abbildung 2.3 2-Tor Wellengrößen und Anschlussleitungen **hftech** zeigt die Streuparameter an einem 2-Tor.

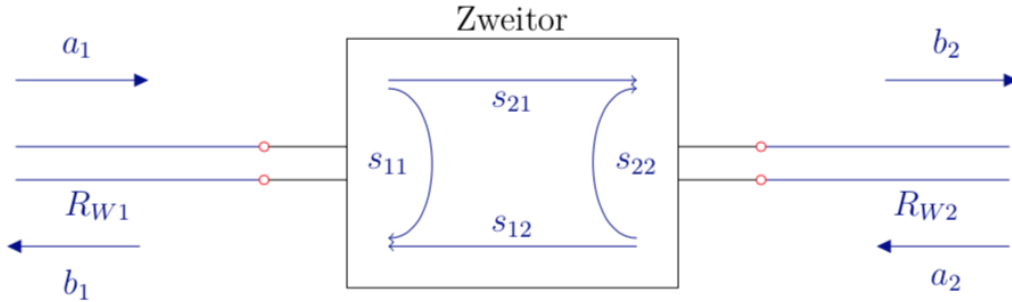


Abbildung 2.3: 2-Tor Wellengrößen und Anschlussleitungen **hftech**

Bei den S-Parameter werden die Eingangs- und Ausgangsgrößen nicht direkt anhand elektrischer Ströme und Spannungen beschrieben. Sie werden mithilfe von Wellengrößen beschrieben, wobei a_i die einlaufenden Wellen sind und b_i die Reflektierenden Wellen. Der Index i stellt den Torindex dar. Formel 2.7 und 2.8 zeigen wie die Wellengrößen a_i sowie b_i definiert sind.

$$a_i = \frac{U_i + R_{Wi} I_i}{2 * \sqrt{R_{Wi}}} \quad (2.9)$$

$$b_i = \frac{U_i - R_{Wi} I_i}{2 * \sqrt{R_{Wi}}} \quad (2.10)$$

Die Wellengrößen gelten nur für den gegebenen Bezugswiderstand R_{Wi} . Der Bezugswiderstand kann einerseits der Innenwiderstand der angeschlossenen Quelle sein oder der Lastwiderstand der angeschlossenen Last.

Aus der Abbildung 2.3 lässt sich folgende Streumatrix darstellen (Formel 2.9):

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} * \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \quad (2.11)$$

Die Elemente der S-Matrix sind:

$$s_{11} = b_1/a_1 \text{ Eingangsreflexionsfaktor bei angepasstem Ausgang (} a_2=0 \text{)} \quad (2.12)$$

$$s_{12} = b_1/a_2 \text{ Rückwärtstransmissionsfaktor bei angepasstem Eingang (} a_1=0 \text{)} \quad (2.13)$$

$$s_{21} = b_2/a_1 \text{ Vorwärtstransmissionsfaktor bei angepasstem Ausgang (} a_2=0 \text{)} \quad (2.14)$$

$$s_{22} = b_2/a_2 \text{ Ausgangsreflexionsfaktor bei angepasstem Eingang (} a_1=0 \text{)} \quad (2.15)$$

2.1.4 Parasitäre Paramter

In diesem Unterkapitel werden grundsätzlich die Einflüsse und Eigenschaften von Parasitären Parametern in Realen Bauteilen, besonders Spule und Kondensator, erklärt

Ideale Bauteile beschreiben eine Funktion. Da reale Bauteile aus Materialien mit physikalischen Eigenschaften bestehen treten bei der Umsetzung dieser Funktion unerwünschte Nebeneffekte auf. Sie entstehen, weil die einzelnen Bauteile im betrieb elektrische Felder oder Magnetfelder erzeugen. Oder einfach durch die Leitfähigkeit eines Materials. Diese physikalisch bedingten Effekte werden als parasitär bezeichnet. Sie treten als Widerstand, Induktivität oder Kapazität auf. Da sie gut klassifiziert werden können werden sie als Parameter bezeichnet. Um eine Schaltung präzise zu Simulieren ist es unerlässlich, die elektrischen Bauelemente mit den passenden parasitären Parameter zu ergänzen. In Abbildung ?? und ?? werden die parasitären Parameter von Spule und Kondensator gezeigt. Man zeigt sie als zusätzliche Bauteile auf.

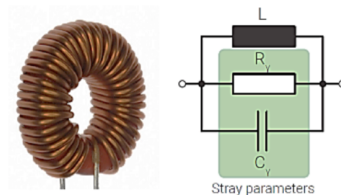


Abbildung 2.4: Parasitäre Elemente einer Induktivität **aufgabenstellung**

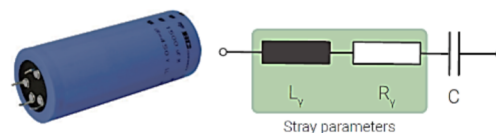


Abbildung 2.5: Parasitäre Elemente einer Kapazität **aufgabenstellung**

2.1.5 Einfügedämpfung

Die Einfügedämpfung(engl. Insertion loss) ist eine Grösse, die verwendet wird, um das Verhalten einer Schaltung zu beschreiben. Sie beschreibt das Verhältnis zwischen der eingehenden Leistung zur abgegebenen Leistung. Sie wird in Dezibel (dB) angegeben. Es handelt sich um eine logarithmische Grösse. Um die Einfügedämpfung in einem Bereich von bis 30MHz abzudecken wird die Formel 2.14 verwendet. In der Formel 2.14 wird die Einfügedämpfung mittels Streuparameter(S-Parameter) berechnet. Sie wird in Dezibel (dB) angegeben. Der Streuparameter S_{21} beschreibt im wesentlichen den Transmissionsgrad des eingehenden Signals. Der Transmissionsgrad beschreibt, welchen Anteil des eingehenden Signals am Ausgang wieder herauskommt.

$$IL = -20 * \log(|S_{21}|) \quad (2.16)$$

2.1.6 Aufbau eines EMI-Filters

Ein EMI-Filter ist ein lineares Netzwerk aus R, L, C Gliedern und einem Transformator. Somit besitzen sie eine reziproke Übertragungssymmetrie, was eine einfache Berechnung von verschiedenen Zusammenhängen erlaubt.

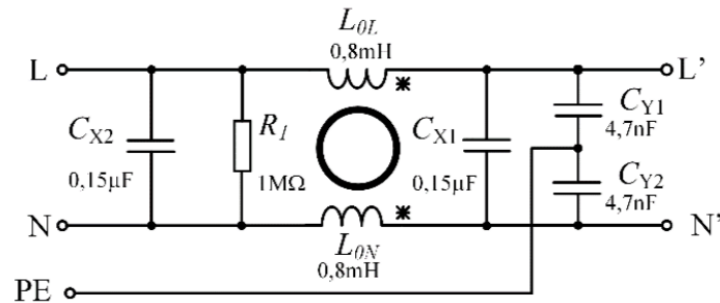


Abbildung 2.6: Original Schaltung aufgabenstellung

2.1.7 Gleich- Gegentaktschaltung

In der realen Stromverteilung wird beabsichtigt, dass der Stromfluss über einen Zuleiter zum Verbraucher hinein-, respektive über einen Ableiter herausgeführt wird. Diese Art der Signalübertragung wird als Gegentakt-Betrieb bezeichnet. Im realen Stromnetz ist allerdings auch der sogenannte Gleichtakt-Betrieb vorhanden. Dabei wirken alle Leiter als Zuleiter, der Gesamte Strom wird durch die Erde weggeleitet. Durch das Gesetz der Superposition ist es möglich, den Gleichtakt- und den Gegentaktanteil getrennt voneinander zu betrachten. Dieses Phänomen wird Anhand der Abbildung 2.7 klar.

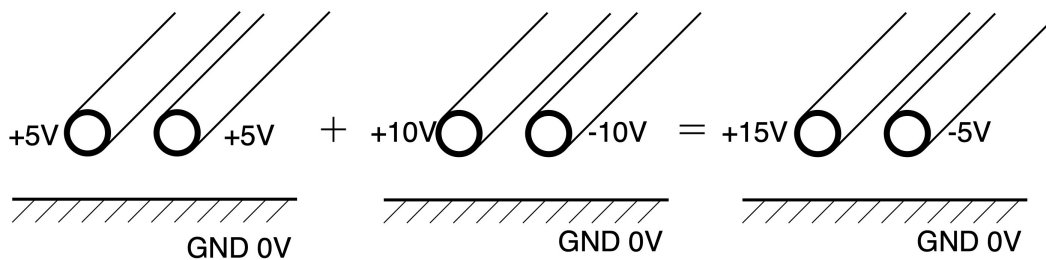


Abbildung 2.7: Beispiel der aufgetrennten Leitung

An einen geerdeten Verbraucher sind 2 Phasen angeschlossen. An der Zuleitung liegt eine Spannung von 15 Volt an, an der Rückleitung liegen -5 Volt an. Diese Leitung wird nun aufgeteilt in eine Gleichtaktleitung, bei welcher über beide Phasen 5 Volt eingespeist werden und in eine Gegentaktleitung, in welcher durch die Zuleitung 10 Volt, respektive in der Rückleitung -10V eingespeisen werden. Während in der Gleichtaktleitung die Addierten 10 Volt gegenüber der Erde anliegen, werden sie in der Gegentaktleitung abgeführt.

2.2 Programmieren

2.2.1 MVC-Struktur

Das MVC-Framework wird zur Softwarestrukturierung verwendet. Durch diese Strukturierung werden die Berechnungen der Daten (eng. model), die Steuerung (engl. controller) und dessen graphischer Repräsentation (engl. view) getrennt. In der Abbildung (TODO) ist dieser Aufbau in einem Beispielklassendiagramm dargestellt.

Der Ablauf dieser Struktur ist wie folgt:

1. Benutzereingabe löst Event aus
2. Die Aktion wird dem Controller übergeben. Dieser holt die Daten in der View, leitet diese dem Model weiter und löst die Berechnungen aus
3. Das Model führt die Berechnungen aus und informiert den Observer
4. Das Model führt die Berechnungen aus und informiert den Observer
5. Der Observer löst ein Event in der View aus. View kann die Daten vom Model holen und Ausgeben

3 Umsetzung

Mithilfe der Grundlagen aus Kapitel 22 wird nun erläutert, wie aus der Schaltung des EMI-Filters mithilfe der Auftrennung in eine Gegentakt - und eine Gleichtaktschaltung die Einfügedämpfung(engl. Insertion Loss, IL) berechnet wird.

3.1 Berechnung der Einfügedämpfung

Die ursprüngliche Schaltung wird in eine Gleich- und Gegentschaltung aufgetrennt. Dieses Prinzip ist eine gängige Methode, da so Symmetrien entstehen, welche vereinfacht werden können. Für die beiden Ersatzschaltungen, werden jeweils als Kettenmatrizen $??$ gebildet. Diese bestehen wiederum aus Kettenmatrizen der einzelnen Längs- und Querimpedanzen. Aus der Kettenmatrix für die Ersatzschaltbilder wird mithilfe der Streuparameter direkt die Einfügedämpfung berechnet. Kettenmatrizen werden durch eine Kaskadierung vorgenommen, dies entspricht im wesentlichen der Matrixmultiplikation.

Zusammenfassen der Schaltungen

Folgendes Kapitel zeigt Schritt für Schritt auf, wie die gegebenen Schaltungen vereinfacht werden können. Zudem wird beschrieben welche Komponenten vernachlässigt werden können. Der erste Abschnitt behandelt die Gleichtaktschaltung und in einem zweiten Abschnitt wird die Gegentaktschaltung behandelt.

Gleichtaktschaltung

Abbildung 3.1 zeigt die Schaltung aus der Aufgabenstellung.

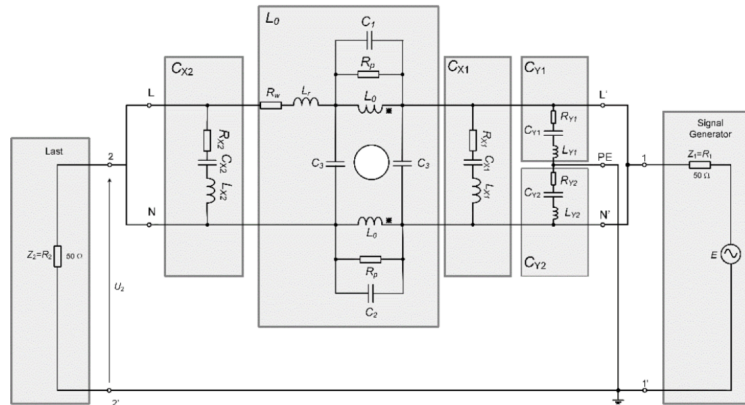


Abbildung 3.1: Originale Gleichtaktschaltungsaufgabenstellung

Die Originalschaltung ist mit den Komponenten R_w und L_r ergänzt worden, sodass sie symmetrisch ist(siehe Abbildung 3.2). Dies macht es möglich, dass die Schaltung zur Simulation wie folgt vereinfacht werden kann.

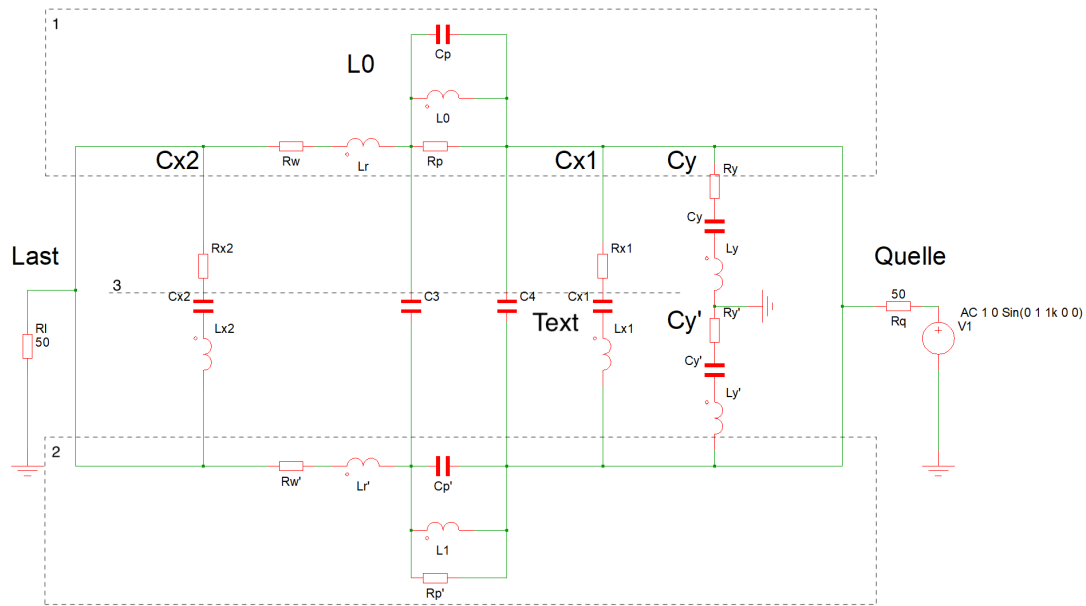


Abbildung 3.2: Ergänzte Gleichtaktschaltung

Da der obere (siehe Abbildung 3.2, Nr. 1) und untere Strang (siehe Abbildung 3.2, Nr. 2) identisch sind und es keinen Potentialunterschied zwischen ihnen gibt, kann die Schaltung, wie folgt, zusammen gefasst werden (siehe Abbildung 3.3). Die Schaltung wird entlang der Symmetrie-Achse (siehe Abbildung 3.2, Nr. 3) aufgetrennt. Somit fallen die Kondensatoren C_3 , C_4 , C_{x1} und C_{x2} komplett weg. Die übrigen Komponenten von L_0 bilden eine Parallelschaltung, welche sich durch halbieren der Widerstände und Induktivitäten und verdoppeln der Kapazitäten zusammenfassen lässt. Zusätzlich werden die beiden C_y und C'_y parallel auf das Bezugspotential geschaltet. Da C_y und C'_y identisch sind, werden sie wie in Abbildung 3.3 zusammengefasst. Diese vereinfachte Schaltung bildet die Grundlage für die Berechnungen der Software.

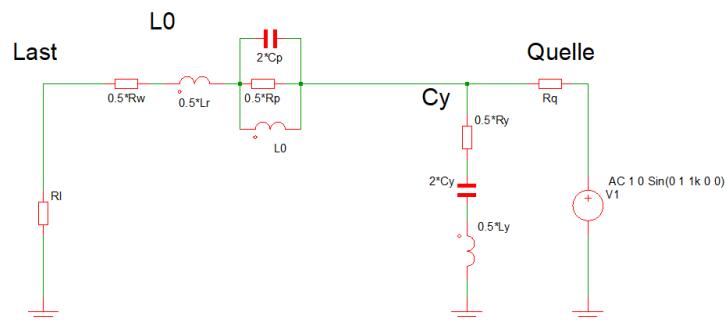


Abbildung 3.3: Vereinfachte Gleichtaktschaltung

Gegentaktschaltung

Abbildung 3.4

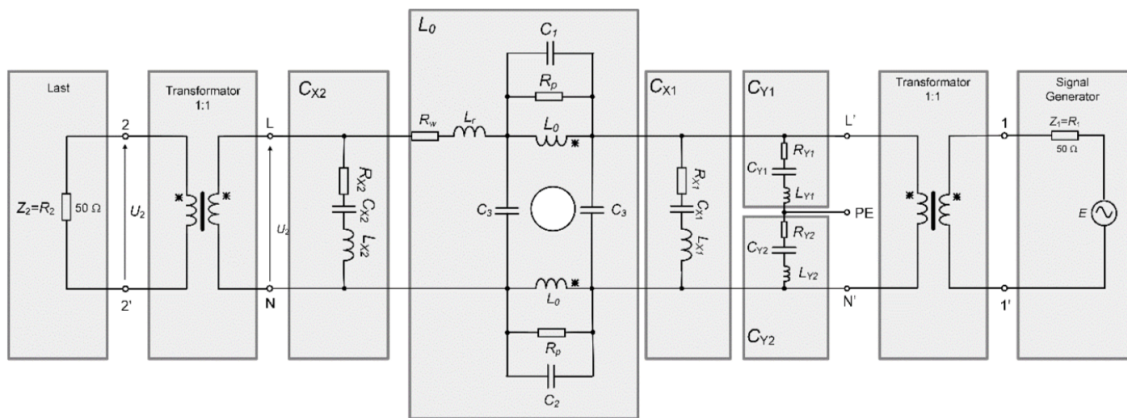


Abbildung 3.4: Ergänzte Gegentaktschaltung

Dieser Abschnitt fehlt noch

Rückwandlung

Die Kettenmatrizen führen dazu, dass die einzelnen Schaltungsteile entweder eine Längs-, oder eine Querimpedanz bilden müssen. Die Aufteilung der einzelnen Schaltungsteile in Längs- und Querimpedanzen und ist in den Abbildungen (Verweis Aufteilung 1 und 2) grafisch dargestellt, wobei „LI“ eine Längsimpedanz kennzeichnet und „QI“ eine Querimpedanz.

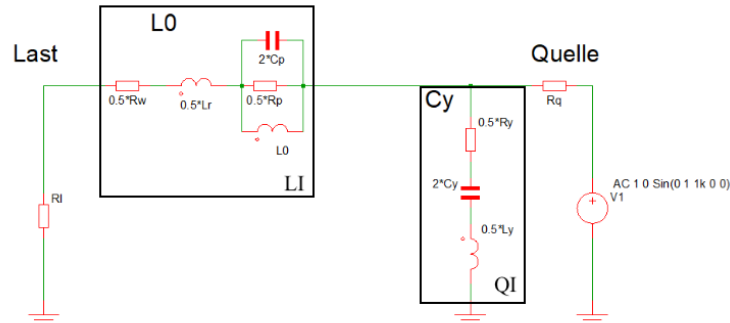


Abbildung 3.5: Einteilung der Gleichtaktschaltungsteile

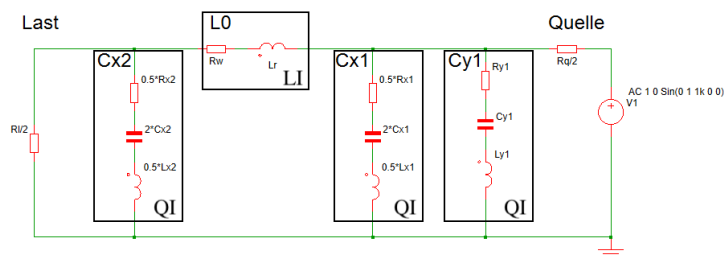


Abbildung 3.6: Einteilung der Gegentaktschaltungsteile

Im nächsten Schritt werden die Impedanzen der einzelnen Schaltungsteile gebildet, welche in die passenden Kettenmatrizen eingesetzt werden.

$$s_{21} = \frac{2}{A_{11} + \frac{A_{12}}{R_w} + A_{21} * R_w + A_{22}} \quad (3.1)$$

Anhand der Kettenmatrizen der beiden Schaltungen, wird mit der Formel 3.1 der Streuparameter S_{21} berechnet. A1 bis A4 entsprechen den einzelnen Einträge der Matrix. R_w ist die Bezugsimpedanz. Die Bezugsimpedanz bezieht sich auf die Innenimpedanz der Quelle und die Lastimpedanz. Bezüglich der Aufgabenstellung (Verweis Aufgabenstellung) ist dieser auf 50Ohm festgelegt. Ausserdem ist es wichtig, dass die beiden Impedanzen gleich gross sind damit die Schaltung Reziprok ist (Verweis Theoretische Grundlagen: Reziproke Schaltungen). (Wieso ?) Bei der reduzierten Gegentaktschaltung (Abbildung ??) werden die beiden Impedanzen mit 25Ohm aufgeführt. Dies ergibt sich durch das Zusammenfassen der Gegentaktschaltung.

Die Berechnungen werden in MATLAB getätigt und geplottet. Diese Plots werden dann mit Simulationen in MPLAB Mindi verglichen um festzustellen ob diese korrekt sind. Nach Überprüfung auf vollständigkeit und korrektheit der Berechnungen, können diese in das Java implementiert werden.

4 Software

In diesem Kapitel wird die Umsetzung der Berechnung (Kapitel 3) und der anderen Anforderungen an die Software in Java dokumentiert. Die programmiertechnischen Grundlagen sind in Kapitel 2 beschrieben.

4.1 Übersicht

Das Programm simuliert die Einfügedämpfung eines EMI-Filters. Solche Filter werden häufig in Schaltnetzteile verbaut, um zu verhindern, dass Störungen zurück ins Netz gespeist werden. Die Benutzeroberfläche des Programms ist in der Abbildung ?? ersichtlich.

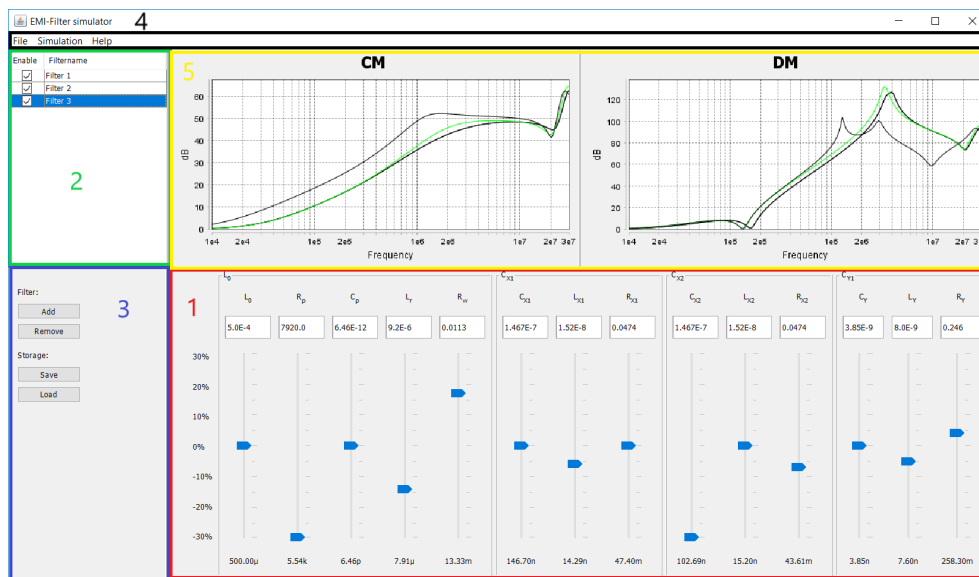


Abbildung 4.1: GUI

Die Software ist nach dem Model-View/Controller Entwurfsmuster **MVCDesignPattern** aufgebaut. Das Klassendiagramm der Software befindet sich im Anhang (TODO: Verweis). Die View ist in fünf Panels unterteilt. Das InputPanel (1) beinhaltet für jeden Filterparameter ein Textfeld, einen Slider und ein Anzeigelabel, welche die Parameterwerte definieren. Das FiltertablePanel (2) verwaltet die verschiedenen Filter in einer Tabellen, in der auch die Parameterwerte hinterlegt werden. Das ButtonPanel (3) besitzt vier Buttons, die für die Speicher- und Filterverwaltung zuständig sind. Die Menubar (4) besitzt verschiedene Menüs für die Bedienung des Programms. Das PlotPanel (5) dient zur grafischen Darstellung der Einfügeverluste des Filters. Der Controller leitet Informationen von der View zu dem Model weiter. Das Model besitzt Klassen, um mit die Komponente der modellierten realen Bauteile zu rechnen. Via eines Observer wird ein Update in der View ausgelöst.

4.2 View

Die View ist als GridBagLayout organisiert und enthält die im Kapitel 4.1 erwähnten Panels.

Inputpanel

Das InputPanel hat den Controller und ist im GridBagLayout organisiert. Das Panel wird in mehreren Subpanels unterteilt. Diese sind in der Abbildung (TODO: verweis) abgebildet.

Das InformationPanel(1) ist im Null-Layout organisiert. Es werden die Prozentzahlen der Schieberegler dargestellt. Diese sind an einer festen Position, da die Grösse des Inputpanels nicht verändert werden kann. Die weiteren Subpanels (2-5) sind im GridBagLayout organisiert. In diesen Panels befinden sich die Komponenten, der modellierten realen Bauteile. Jede Komponente besitzt ein Textfeld, einen Slider und ein Label. Im Textfeld kann der Benutzer seine Werte für die Komponente eintragen. Beim Aufstarten des Programmes wird das Feld mit einem Standardwert, der vom Auftragsdokument übernommen wurde, geladen. Mit der Klasse JEngineerField werden die Eingaben geprüft. Es ist nur möglich Zahlen einzugeben. Grosse und kleine Zahlen können zur Vereinfachung in wissenschaftlicher Schreibweise (18e-12) oder in Einheiten-Schreibweise (18p) eingetragen werden. Die Ausgabe ist als Einheiten-Schreibweise vordefiniert. Mit einem Rechtsklick auf das Textfeld ist es möglich die Schreibweise der Eingabe und Ausgabe individuell zu verändern. Mit dem Schieberegler wird der im Textfeld eingegebener Wert um $\pm 30\%$ verändert und im Label als effektiver Wert ausgegeben. Das Textfeld besitzt einen ActionListener und der Slider einen changeListener. Bei einem Ereignis wird die dazugehörige Methode im Controller aufgerufen.

Filtertablepanel

Das Filtertablepanel hat den Controller und ist im GridBagLayout organisiert. Es enthält eine Tabelle, in der die Parameterwerte hinterlegt und zu dem entsprechenden Filter zugeordnet werden. Jeder Filter in der Tabelle besitzt eine Checkbox, um den Filter zu aktivieren, und ein Textfeld, um den Filter zu benennen. Die Tabelle hat einen TableModelListener und einen ListSelectionListener. Bei einem Ereignis wird die dazugehörige Methode im Controller aufgerufen.

Buttonpanel

Das ButtonPanel hat den Controller und ist im Null-Layout organisiert. Es enthält die vier Buttons und zwei dazugehörige Labels. Mit diesen kann die Filtertabelle verwaltet und die Speicherverwaltung aufgerufen werden. Jeder Button hat einen ActionListener. Bei einem Ereignis wird die dazugehörige Methode im Controller aufgerufen.

Plotpanel

Das PlotPanel ist im BorderLayout organisiert. Mit Hilfe des Package JfreeChart (TODO: verweis) wird die Einfügedämpfung des EMI-Filters in einem Plot dargestellt. Das Package übernimmt viele Funktionen wie das Zoomen des Plots oder das Ändern der Darstellung. Löst der Observer ein Ereignis aus, werden die Daten vom Model geholt und geplottet.

Menubar

Das Menubar hat den Controller und ist im Null-Layout organisiert. Es besitzt die Menus File, in dem die Speicherverwaltung aufgerufen werden kann, Circuit, in dem die Schaltungen, von denen die Berechnungen ausgehen angezeigt werden und About, in dem Informationen über das Programm hinterlegt sind. Die Menüs haben einen ActionListener. Bei einem Ereignis wird die dazugehörige Methode im Controller aufgerufen.

4.3 Controller

Der Controller dient als Schnittstelle zwischen dem Model und der View. Dabei werden die von View aufgerufenen Funktionen an das Model weitergeleitet. Diese Werte, die das Model berechnet hat, werden anschliessend wieder an den Controller gesendet und zurück zur View geleitet. Der Controller besteht aus 9 Methoden, wobei eine der Konstruktor ist. Von diesen Methoden wird das Model nur von den beiden Methoden `calculateInsertionloss()` und `deleteRowInCalculationData()` aufgerufen. Die Methode `calculateInsertionloss()` ist dafür zuständig den Insertionloss für die Schaltungen, mit den in der GUI eingestellt Parametern, zu berechnen.

4.4 Model

to be done => Beispiel Anschauen von Hr Gut

4.5 Trace

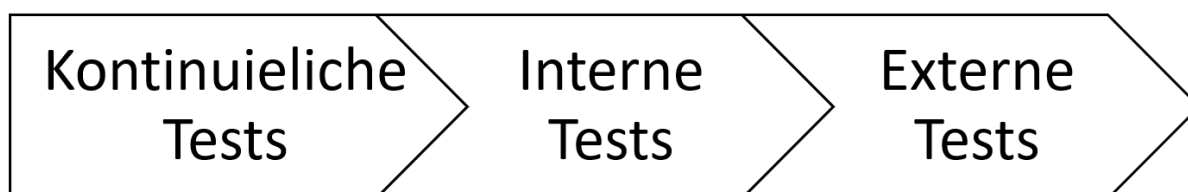
Um den Ablauf innerhalb der Software nachvollziehen zu können wird nachfolgend die sogenannte "Trace" beim Aufstarten der Software und beim Auslösen eines Ereignisses aufgezeigt. Die "Trace" wird innerhalb der Software mithilfe der Klasse `traceV4` in die Konsole ausgegeben.

5 Testkonzept

5.1 Aufbau

In der unteren Tabelle wird aufgeführt, welche Tests durchgeführt werden, um einen Überblick zu geben wie das Testkonzept aufgebaut ist. Die detaillierten Testprotokolle, die für die internen und externen Test verwendet werden, wird auf den Anhang verwiesen.

Tests Übersicht	Intern	Extern	Ablauf	Ziel des Tests
Code vor der Implementation Testen	X		Die Methoden werden einzeln ausgeführt	Fehler von Vorhinein zu Verhindern
Struktur des Codes überprüfen.	X		Es wird geschaut, ob die geplante Struktur vorhanden ist.	Übersichtlichkeit der Software gewährleisten
Berechnete Werte Validieren	X		Die Berechneten Resultate werden mit Hilfe von MATLAB und MPLAB kontrolliert.	Folgefehler vermeiden.
Kompatibilitätstest	X	X	Die Software wird auf verschiedenen Betriebssystemen und Displays ausgeführt.	Soll eine fehlerfreie Darstellung der Software gewährleisten.
Fehleingaben	X	X	Es werden Eingaben getätigt, die die Software an die Grenzen bringen dürfte.	Die Software soll so stabil wie möglich sein.
Den optische Aufbau der GUI betrachten.	X	X	Beurteilen ob der Aufbau der GUI Sinnvoll ist.	Die GUI Soll schlüssig aufgebaut sein.
Die einzelnen Funktion der GUI testen.	X	X	Überprüfen Slider, Buttons, Menu etc. die richtige Action auslösen.	Die Funktionalität der GUI wird gewährleistet.
Test durch den Auftraggeber		X	Nach Vollendung der Version 0.9.5 wird die Software dem Auftraggeber abgegeben, damit er seine Meinung und Ideen einbringen kann.	Der Auftraggeber soll zufrieden mit dem Endprodukt sein.



5.2 Validierung

Für die Berechnung des Insertionloss wird die Schaltung in mehreren Schritten vereinfacht. Die Zwischenschritte müssen auf ihre Korrektheit geprüft werden, um Folgefehler zu vermeiden, die wertvolle Zeit kosten können. Diese werden mithilfe von mit Hilfe von MATLAB und MPLAB überprüft. Mit MATLAB werden die Rechenschritte noch einmal durchgeführt. Während in MPLAB die vereinfachte Schaltung Simuliert wird. Die erhaltenen Werte der beiden Tools werden miteinander verglichen, ob sie übereinstimmen.

5.3 Erwartungen

Die zu erwartenden Testresultate variieren stark von den Testperson, die die Software testet. Die internen Tests werden die groben Fehler herausfiltern und schaffen eine stabile Grundlage auf die aufgebaut werden kann. Zudem wird überprüft, ob die alle Ziele erreicht wurden. Bei den Fachpersonen wird das Feedback höchst wahrscheinlich sehr umfangreich ausfallen. Es ist zu erwarten, dass Fehler entdeckt werden, die noch nicht bekannt sind oder die Software gar zum Absturz gebracht wird. Die Fachfremden Tester werden dies nicht erreichen, jedoch erhalten wir eine Hilfreiche Rückmeldung, was die Benutzerfreundlichkeit betrifft, weil diese Personen einen anderen Blick auf das grosse ganze haben. Das Feedback des Auftraggebers wird sehr detailliert ausfallen, weil er genaue Vorstellungen hat was er von dem Produkt haben möchte. Es wird sich, aber vermutlich mehrheitlich, um die Funktionen drehen und nicht welche Fehler es gibt.

6 Schluss

7 Anhang

7.1 Testkonzept

Testprotokoll Team 1

Name des Testers:




Datum :




Softwareversion: 0.9




Test Art (intern oder extern):




Test Übersicht


1. Test	Software öffnen und schliessen. Schieberegler und Buttons auf die Funktion prüfen. Das Fenster grösser kleiner machen.
2. Test	Das Aussehen der GUI betrachten. Auf Vollständigkeit testen. (Gesamteindruck)
3. Test	Kontrollieren, ob der Plot gezeichnet wird. Und Filtertypen gespeichert werden können.
4. Test	Die Menu-Funktionen testen. Shurcuts ausprobieren
5. Test	Fehleingaben machen.
6. Test	Verschiedene Betriebssysteme und Displays verwenden (falls vorhanden).
7. Test	Code auf Übersichtlichkeit prüfen. (Experten)

Test Nr.	Der Test	Bewertung	Anmerkungen (Fehler)
1.	Software öffnen und schliessen. Schieberegler und Buttons auf die Funktion prüfen. Das Fenster grösser kleiner machen.	Gut  Mässig  Schlecht 	
Kommentar / Verbesserungsvorschlag			




Test Nr.	Der Test	Bewertung	Anmerkungen (Fehler)
2.	Das Aussehen der GUI betrachten.	Gut  Mässig  Schlecht 	
Kommentar / Verbesserungsvorschlag			

Test Nr.	Der Test	Bewertung	Anmerkungen (Fehler)
3.	Kontrollieren, ob der Plot gezeichnet wird. Und Filtertypen gespeichert werden können.	Gut  Mässig  Schlecht 	
Kommentar / Verbesserungsvorschlag			




Test Nr.	Der Test	Bewertung	Anmerkungen (Fehler)
4.	Die Menu-Funktionen testen	Gut  Mässig  Schlecht 	
Kommentar / Verbesserungsvorschlag			

Test Nr.	Der Test	Bewertung	Anmerkungen (Fehler)
5.	Fehleingaben machen.	Gut  Mässig  Schlecht 	

Kommentar / Verbesserungsvorschlag

Test Nr.	Der Test	Bewertung	Anmerkungen (Fehler)
6.	Verschiedene Betriebssysteme und Displays verwenden (falls vorhanden).	Gut  Mässig  Schlecht 	

Kommentar / Verbesserungsvorschlag

Test Nr.	Der Test	Bewertung	Anmerkungen (Fehler)
7.	Code auf Übersichtlichkeit prüfen.(Experten).	Gut  Mässig  Schlecht 	
Kommentar / Verbesserungsvorschlag			

Abnahmeprotokoll Software

Ersteller: _____

Auftraggeber: _____

Auftrag: _____

Version: _____

Datum: _____

Ziele

Berechnungen und GUI getrennt	
Berechnungszeit < 500 ms	
Verstellbare Parameter	
CM-, DM-Berechnung	
CM-, DM-Darstellung	
Feinjustierung mit Schieberegler +/- 30%	
Darstellung im Frequenzbereich bis 30MHz	
Zahlenwerte könne eingegeben werden	
Mehrere Plots gleichzeitig darstellen	
Unabhängigkeit von Betriebssystemen	
Schutz vor Fehleingaben	
Anpassung von Farbe, Darstellung und Schrift	
Zoom Möglichkeit	

Kommentare

Test

Kontinuierliche Tests	
Interne Tests	
Externe Tests durch Fachpersonen und Laien	
Test der 0.9.5 durch den Auftraggeber	

Projektleiter

Auftraggeber

Unterschrift, Ort, Datum

Unterschrift, Ort, Datum