**Fachbericht**



**«DJ» EMI Filter für Netzteil**

Pro2E - Team 5

**Auftraggeber:** Luca Dalessandro

**Dozierende:** Anita Gertiser

Pascal Buchschacher

Peter Niklaus

Sebastian Gaulocher

Richard Gut

**Projektteam:** Marina Taborda, Projektleiterin

Michel Alt, Stv. Projektleiter

Frank Imhof

Luca Krummenacher

Richard Britt

Fady Hanna

Windisch, 09.05.2019

**Abstract**

*Alle elektronischen Geräte benötigen heute Netzschaltteile, um die Netzspannung in die gewünschte Gerätespannung zu transformieren. Die verwendeten Bauteile in den Schaltnetzteilen erzeugen dabei leistungsgebundene Störungen.*

*Damit diese Störungen (Differential Mode und Common Mode) auf der Netzseite keine negativen Auswirkungen haben, müssen die Netzschaltteile die normativen Anforderungen erfüllen und dementsprechend dimensioniert werden.*

*Diese Normen werden von den europäischen Normengremien festgelegt und sind ständig in Bearbeitung.*

*Das Ziel dieser Arbeit ist die Entwicklung eines bedienerfreundlichen Programms, das in der Lage ist, die Einfügungsverluste des Filters zu berechnen, sowie graphisch darzustellen. Das Programm soll ausserdem aufzeigen, wie sich bestimmte Parameter, auf die zwei Störungsarten im ganzen Frequenzspektrum auswirken. Diese Informationen sollten dem Auftraggeber das Dimensionieren von Filtern vereinfachen.*

Inhaltsverzeichnis

[1 Einleitung 4](#_Toc8117372)

[2 Theoretische Grundlagen 5](#_Toc8117373)

[2.1 EMI Filter 5](#_Toc8117374)

[2.2 Störungsarten 5](#_Toc8117375)

[2.3 Definition Einfügungsverluste «Insertion loss» 6](#_Toc8117376)

[2.4 Parasitäre Parameter 6](#_Toc8117377)

[2.5 MVC 7](#_Toc8117378)

[3 Software 7](#_Toc8117379)

[3.1 Betriebssystem/Lizenzierung: 7](#_Toc8117380)

[3.2 Strukturierung 8](#_Toc8117381)

[3.2.1 Klassendiagram 8](#_Toc8117382)

[3.3 Programmablauf 9](#_Toc8117383)

[3.3.1 Programmstart 9](#_Toc8117384)

[3.3.2 Laden und Speichern von Daten 9](#_Toc8117385)

[3.3.3 Eingabe der Daten 9](#_Toc8117386)

[3.4 Implementierung der Berechnungen in JAVA 10](#_Toc8117387)

[3.5 GUI 10](#_Toc8117388)

[3.5.1 Architektur 10](#_Toc8117389)

[4 Elektrotechnik 10](#_Toc8117390)

[4.1 Vorgehensweise Berechnung Einfügungsverluste 10](#_Toc8117391)

[4.2 S-Parameter 11](#_Toc8117392)

[4.3 Vereinfachungen der Schaltungen: 12](#_Toc8117393)

[4.4 Realisierung *in Matlab* 12](#_Toc8117394)

[5 Validierung 13](#_Toc8117395)

[5.1 Berechnungen 13](#_Toc8117396)

[5.2 Software 13](#_Toc8117397)

[5.2.1 Eingabe 13](#_Toc8117398)

[5.2.2 Benutzerfreundlichkeit 13](#_Toc8117399)

[6 Schluss 13](#_Toc8117400)

[Quellenverzeicnis 14](#_Toc8117401)

[Anhang 15](#_Toc8117402)

# Einleitung

Der vorliegende Fachbericht dient als Darstellung der aktuellen Projektarbeit des Moduls pro2E. Der Auftrag des Projekts ist es, eine Softwareanwendung zu programmieren, welche es ermöglicht das Frequenzverhalten und die Einfügungsverluste eines EMI-Filters darzustellen. Anhand eines Mischpults mit Slidern sollen die Werte der Parameter verändert werden können und anschliessend die Einflüsse auf das Verhalten in einem Graphen aufzeigen. Ebenfalls Teil der Aufgabenstellung ist es, die korrekten Berechnungen der verschiedenen Störungsarten (Gleich- und Gegentaktstörungen) zu ermitteln und diese in die Softwareanwendung zu implementieren.

Gemäss Aufgabenstellung (im Anhang) soll die Bedienoberfläche benutzerfreundlich sein. Somit ist ein Ziel dieser Arbeit, dass ein Laie das Programm ohne Erklärung und ohne Anleitung bedienen kann (-> *Doppelklick*). Ausserdem soll die Softwareanwendung auf den Betriebssystemen MacOS (ab Version 10.11) und auf Windows (7 oder neuer) fehlerfrei funktionieren. Ziele der Berechnungen sind ihre Korrektheit und ihre Übereinstimmung mit den Simulationen des Auftraggebers.

*In dieser Projektarbeit soll eine Software entwickelt werden, welche es mit deren Hilfe das Verhalten eines EMI-Filters berechnet und grafisch dargestellt kann.*

*Das Programmsoll so Es wird eine benutzerfreundliche Software erstellt, mit der man die Einfügungsverluste in Abhängigkeit der Frequenz simulieren kann. Dabei wird der Filter bei zwei verschiedenen Störungsarten simuliert, den Gleichtaktstörungen (CM) und den Gegentaktstörungen (DM). Ausserdem soll das Programm die Auswirkungen der parasitäreren Parameter auf die Einfügungsverluste darstellen können.*

Die Software wurde in der Programmiersprache Java geschrieben. Damit die Software später einfach verändert werden kann, wurde die Software in drei Bereiche unterteilt: Die Berechnungen (Model), das Userinterface (View) und die Schnittstelle (Controller). In der View soll die Bedienoberfläche gebaut und die Graphen gezeichnet werden. *Um das Layout des Programms möglichst von den Berechnungen und Funktionen zu trennen, wurde auf JavaFX gegriffen. JavaFX ermöglicht es mit Hilfe von Cascading Style Sheets (CSS) Layouteinstellungen simpel zu machen.* Für die Berechnung der Graphen liest das Model die Daten aus und berechnet die Kurven. Die View und das Model werden mit dem Controller miteinander verbunden. Ausserdem wurden weitere Klassen erstellt, welche die Struktur des Programms vereinfachen.

Für die Realisierung der Berechnungen wurden die von der Firma Schaffner erhaltenen Schemas der Gleichtakt- und Gegentaktstörungen vereinfacht. Die darin enthaltenen Bauelemente wurden in Längs- und Querimpedanzen eingeteilt und zu einer Gesamtmatrix zusammengeführt. In Matlab wurden daraus die Einfügungsverluste berechnet und in einem Kurvendiagramm dargestellt. Die Berechnungen der vereinfachten Schaltungen wurden mit den Einfügungsverlusten der Originalschaltung verglichen und durch die Fachcoaches validiert.

Des Weiteren wird zusätzlich die Bibliothek JavaFX verwendet, welche es ermöglicht, mit Hilfe von «Cascading Style Sheets (CSS)» den Inhalt noch weiter von der Darstellung zu trennen. Dadurch wird der Code einfacher lesbar und das Design schlicht verwaltet.

# Theoretische Grundlagen

## EMI Filter

Nahezu jedes elektrische Gerät besitzt ein Schaltnetzteil um die Netzspannung auf die benötigte Spannung zu regeln. Betrachtet man die Eingangsspannung ohne Netzfilter, wird man auf dem ganzen Frequenzspektrum, d.h. von Netzfrequenz bis zu mehreren MHz Störungen feststellen. Die Aufgabe vom EMI (elektromagnetische Interferenzen) Filter ist es, diese Störungen zu filtern, so dass keine anderen Geräte gestört werden. Damit dieses eingesetzt werden darf, muss jedes Schaltnetzteil sich an bestimmte Normen im Bereich EMV halten.

Ein EMI Filter für einphasige Geräte besteht nur aus wenigen Bauteilen. Zwei X-Kondensatoren, zwei Y-Kondensatoren, einer Drossel mit zwei Windungen, welche um einen Ferit Ring gewickelt sind und einem Widerstand. Diese Schaltung kann sehr kompakt verbaut werden, was in folgendem Filter von Schaffner sichtbar wird.

Abbildung 1:Schaltung des FN 2020 Filters (Schaffner), sowie der Filter selbst

Die stromkompensierte Drossel (L) ist in der Lage, Gleichtaktstörungen (CM) zu filtern. Diese Störungen treten gleichzeitig auf beiden Leitungen auf.

Die Y-Kondensatoren, welche gegen Erde geschaltet sind, sind ebenfalls dazu da, um CM-Störungen zu filtern. Diese müssen jedoch eine sehr hohe Überspannungsfestigkeit besitzen, um beispielsweise bei einem Blitzschlag keinen Kurzschluss im Gehäuse zu verursachen.

Störungen zwischen den Zuleitungen, so genannte Gegentaktstörungen (DM), werden mit den X-Kondensatoren gedämpft. [1]

## Störungsarten

Die existierenden EMV Normen gelten für das Gesamtrauschen. Doch in der Praxis wird einfachheitshalber die Gesamtstörung in Gleichtaktrauschen, Comon Mode (CM) und Gegentaktrauschen, Differential Mode (DM) gesprochen.

Abbildung 2: Stromzirkulation der Störungen im DM- und CM-Mode

## Definition Einfügungsverluste «Insertion loss»

Die Leistung eines EMI Filters wird mit den Einfügungsverlusten in Abhängigkeit der Frequenz bestimmt. Diese Funktion lautet:

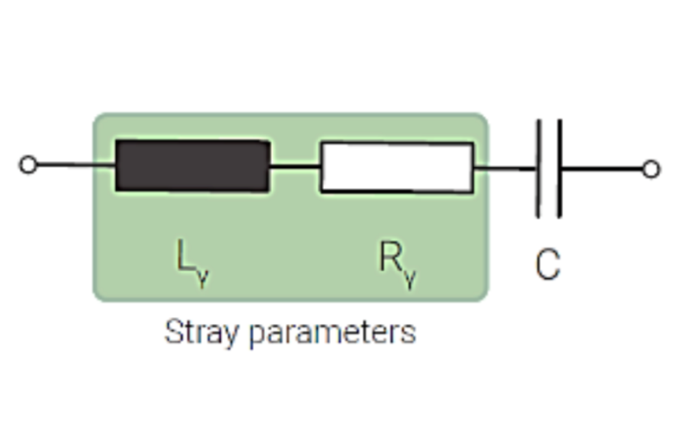
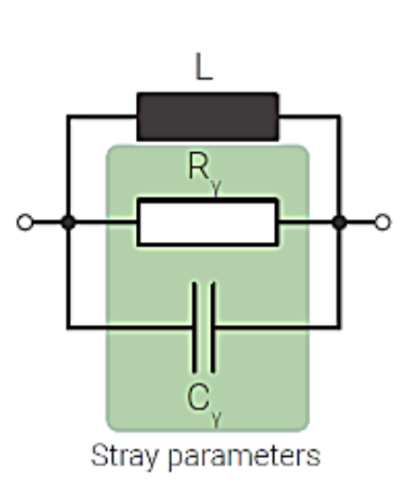
: Lastspannung gemessen ohne Filter mit einer Last von 50 Ω

: Lastspannung gemessen mit EMI Filter

Abbildung 3: Lastspannung ohne Filter und Lastspannung mit EMI Filter

## Parasitäre Parameter

In der Realität verhalten sich die Bauteile eines EMI-Filters leider nicht genau so, wie wir es idealerweise annehmen. Jedes Bauteil hat aufgrund der physikalischen Gegebenheiten eine Ersatzschaltung mit den parasitären Parametern. Diese Ersatzschaltungen sehen wie folgt aus:

A

Abbildung 4: Ersatzschaltung für einen realen Kondensator - eine reale Spule

Betrachtet man diese Ersatzschaltbilder, sieht man eigentlich einen Serie- und Parallelschwingkreis. Weit unterhalb der Resonanzfrequenz haben die parasitären Parameter jedoch keinen grossen Einfluss. Im Resonanzfall haben wir aber einen Serie- oder Parallelschwingkreis. Danach wirkt die Spule nicht mehr als Induktivität und der Kondensator nicht mehr als Kapazität. Die Resonanzfrequenzen liegen dabei meist bei mehreren MHz. Doch bei unserer Anwendung wollen wir auch über diesem Frequenzbereich unsere Schaltung berechnen. Der Widerstand hat auch ein Ersatzschema, dieses wurde aber in dieser Schaltung nicht berücksichtigt, da dieses keinen grossen Einfluss auf die Gesamtschaltung hat.

Deshalb hat man für das Grundschema des Filters, die parasitären Parameter gemäss folgenden Ersatzschemas ergänzt.



Abbildung 5: CM-Ersatzschaltung ohne parasitäre Parameter



Abbildung 6: CM-Ersatzschaltung mit parasitären Parametern

## MVC

Model View Controller (MVC) ist eine Architektur zur Unterteilung einer Software in die drei Komponenten; Datenmodell (Model), Präsentation (View) und Programmsteuerung (Controller). Die MVC-Architektur bildet die grundlegende Organisation und Interaktion zwischen Komponenten einer Applikation. Sie ermöglicht eine flexible Applikation, die Änderungen und Erweiterungen zulässt sowie die Wiederverwendung einzelner Komponenten ermöglicht. Das Model erhält Daten via Controller von der View und arbeitet mit diesen. Es bleibt dabei von der View und vom Controller unabhängig. Änderungen von Daten werden mittels Observer [gibt Datenänderungen an von diesen Datenabhängige Strukturen weiter] der View zurückgegeben. Die View stellt die Daten dar und ist die Schnittstelle zur Interaktion zwischen Benutzer und Applikation. Sie kennt das Modell, dessen Daten sie darstellt, ist aber nicht für die Verarbeitung dieser Daten zuständig. Die View ist vom Controller unabhängig. Bei Dateneingabe durch den Benutzer werden diese mittels Observer an den Controller weitergegeben. Der Controller verwaltet das Model und die View. Er wertet Datenänderungen von Model und View aus und leitet diese an den jeweils anderen weiter.

# Software

## Betriebssystem/Lizenzierung:

*Das Programm wird in der Sprache Java als Desktop-Applikation für die Betriebssysteme MacOS und Windows entwickelt. Um den Inhalt möglichst von der Darstellung trennen zu können, wird die Library JavaFX verwendet.*

## Strukturierung

## 3.2.1 Klassendiagram

Abbildung 15: Klassendiagramm

## Programmablauf

### Programmstart

Beim Starten des Programmes werden bestenfalls die Daten einer zuvor beendeten Session wieder geladen. Nach erfolgreichem Laden erfolgt automatisch das Zeichnen der Kurve und das Setzen der Schieberegler auf ihren Wert. Nutzende können auch via Menüleiste gespeicherte Sessionen laden oder mit einer neuen beginnen.

### Laden und Speichern von Daten

Über die Menüleiste, welche in der Klasse CustomMenüBar umgesetzt ist, lassen sich die eingegebenen Werte über den Befehl «Save as» abspeichern und über den Befehl «load» laden. Wenn diese Befehle betätigt werden wird der Explorer gestartet. Diese Lade und Speichern-Funktionen wurden im Controller implementiert. Die Funktion wurde so programmiert, dass im Explorer nur die Dateiendungen txt, rtf oder pdf auswählbar sind und als diese gespeichert werden können. Um ein Datenformat in einer lesbaren Textform zu bekommen wurde JSON verwendet. Dies hat auch den Zweck die Daten einfacher wieder zurück ins Programm zu laden. Nach erfolgreichem Laden erfolgt automatisch das Zeichnen der Kurve und das Setzen der Schieberegler auf ihren Wert.

//Beim Starten des Programmes werden bestenfalls die Daten einer zuvor beendeten Session wieder geladen.

## 3.3.3 Eingabe der Daten

*Durch Verändern der parasitären Parameter wird das Neuzeichnen der Kurven ausgelöst. Durch Verwendung verschiedener Farben in den Kurvendiagrammen könnten optimale Einstellungen hervorgehoben und von suboptimalen unterschieden werden.*



Abbildung 12: Anpassung der Parameter

*Bei Doppelklick auf einen Parameter wechselt die Darstellung in seinem Fenster.  
Nun können die ursprünglichen Werte der parasitären Parameter via Textfelder und anderen Kontroll-Einheiten geändert werden. Bei erneutem Doppelklick geht die Ansicht wieder zurück zu den Schiebereglern.*



Abbildung 13: Kurvenansicht

*Bei Doppelklick auf ein Diagramm, übernimmt dieses die ganze Breite des Fensters und kann so genauer angesehen werden. Bei erneutem Doppelklick wird der Vorgang rückgängig gemacht, so dass wieder beide Diagramme sichtbar sind.*



Abbildung 14: Parameterauswirkung

*Wenn Nutzende den Mauszeiger über einen Schieberegler bewegen, soll im Diagramm gezeigt werden, welche Auswirkungen dieser Schieberegler bei voller Auslenkung (±30%) auf die Kurve hätte. Somit können die unterschiedlichen Auswirkungen aller parasitären Parameter schnell eingesehen werden, ohne dass die Nutzenden die Schieberegler einzeln verschieben müssen.*

## Implementierung der Berechnungen in JAVA

## GUI

### Architektur

*Es soll ein bedienungsfreundliches Programm entwickelt und realisiert werden, mit dem das Frequenzverhalten und die Einfügungsverluste von CM und DM des EMI-Filters vorhergesagt werden können. Die Auswirkungen, die die parasitären Parameter auf die Einfügungsverluste des Filters haben, müssen im Programm eingesehen werden können. Das Programm soll die beiden Filtervarianten hinsichtlich der Leistung untersuchen und die bessere empfehlen. Optional ist eine Untersuchung der Filtervarianten auf Fussabdruck (CO2? Material? Graue Energie?) und Kosten.*

*Es ist erwünscht, dass das Programm eine Bedienoberfläche hat, welche einem Mischpult mit Schiebereglern ähnlich sieht.*

# Elektrotechnik

## Vorgehensweise Berechnung Einfügungsverluste

*Wir befassen uns zunächst mit der CM-Ersatzschaltung, da sich diese leichter vereinfachen lässt. Man nutzt die Symmetrie der Schaltung aus, indem man bei der CM-Schaltung die beiden Leiter zusammenfasst. Ausserdem lässt man die CX-Kondensatoren weg, da diese bei dieser Schaltung trivial sind. Die reduzierte Schaltung sieht nun schon wesentlich schlanker aus.*

**

Abbildung 7: Vereinfachte CM-Schaltung

*Nun bestimmen wir die Gesamtimpedanz (Z1), sowie die Gesamtimpedanz (Z2).*

*Wobei ist und wir später die Impedanzen für ein Frequenzspektrum von 50Hz bis 30MHz berechnen werden.*

*Wir berechnen nun die Kettenmatrix (A1) für die Längsimpedanz (Z1) sowie (A2) für die Querimpedanz (Z2).*

*Für den S21-Parameter benötigen wir die Gesamtmatrix, diese berechnet man mit Hilfe der Kettenschaltung, in dem man die beiden Matrizen miteinander multipliziert.* [2]

## S-Parameter

*Sobald man die Kettenmatrix für ein Zwei-Tor berechnet hat, lassen sich daraus die S-Parameter berechnen. Deshalb ist es der einfachste Weg, die Einfügungsverluste mit Hilfe dieser Parameter zu bestimmen. Im folgenden Zwei Tor sind die Wellengrössen zur Berechnung der S-Parameter angegeben.*



*Wobei und die Amplituden der einfliessenden Wellen sind.*

*Die Grössen und sind hingegen die Amplituden der reflektierten Wellen.*

Abbildung 8: Zwei Tor mit Wellengrössen

*Daraus lässt sich die Streumatrix S wie folgt berechnen:*

*Die einzelnen Parameter haben dabei folgende Bedeutung* [3]*:*

*: Eingangsreflexionsfaktor*

*: Rückwärtstransmissionsfaktor*

*: Vorwärtstransmissionsfaktor*

*: Ausgangsreflexionsfaktor*

*Wir benötigen den Rückwärtstransmissionsfaktor. Dieser benötigt die Bezugswiderstände Rw, welche bei uns am Ein- sowie am Ausgang 50Ω betragen. Daraus berechnen wir den Streuparameter folgendermassen:*

*Nun können wir die Einfügungsdämpfung a in dB berechnen:*

## Vereinfachungen der Schaltungen:

## Realisierung *in Matlab*

*Diese Berechnungen haben wir in MATLAB realisiert und für verschiedene Frequenzen die Dämpfungen berechnet, dabei haben wir folgende Kurven bekommen. Der Code für die Berechnungen befindet sich im Anhang.* [4][5]



Abbildung 9: Insertion loss Kurve aus MATLAB

# Validierung

## Berechnungen

## Software

### Eingabe

### Benutzerfreundlichkeit

# Schlusswort

*Dem Auftraggeber steht nach erfolgreichem Projektabschluss die Software ”emj-dj“ voll funktionsfähig zur Verfügung. Alle Pflichtziele sowie ein Wunschziel, das Speichern und Laden der Eingabewerte, konnten gemäss Pflichtenheft umgesetzt werden. Die CM und DM Graphen wurden auf Matlab richtig berechnet und ins Java Programm implementiert. Dabei kann man durch abändern der Bauteilwerte dessen Einfluss im Diagramm beobachten und vorhersagen können. Auch können eingegeben Werte abgespeichert werden und für die spätere Verwendung wieder in Programm geladen werden.*

*//Die Problematik*

*Die Problematiken konnten alle bewältigt werden und das Programm wurde fristgerecht geliefert. In weiteren Arbeitsprozessen könnten weitere Wunschziele umgesetzt werden.*

*Ebenfalls kann man das Tool noch beliebig optimieren*

*// Beispiele was man noch erweitern könnte.*

# Quellenverzeicnis

|  |  |
| --- | --- |
| [1] | Schaffner, «Single-Stage Filters FN 2020,» Schaffner Group, Luterbach, 2018. |
| [2] | P. Niklaus, «Zweitore (2-Tore),» FHNW, Windisch, 2018. |
| [3] | A. Schaum, «Zweitore - Zusammenfassung,» Lehrstuhl für Regelungstechnik, Universität zu Kiel, 2017. |
| [4] | A. Maier und S. Rupp, «Hochfrequenztechnik, Teil 2 - Anwendungen,» DHBW, Stuttgart, 2016. |
| [5] | B. Mößlang, «EMC Filter Insertion Loss Simulation,» OMICRON Lab, Reutlingen, 2016. |

# Anhang