**Pflichtenheft**

**Fachlicher Teil**



**«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, 29.03.2019

Inhaltsverzeichnis

[1 Übersicht 3](#_Toc4760341)

[1.1 Ausgangslage 3](#_Toc4760342)

[1.2 Projektziele 4](#_Toc4760343)

[1.3 Wunschziele 4](#_Toc4760344)

[1.4 Nicht-Ziele 5](#_Toc4760345)

[1.5 Lieferobjekte 5](#_Toc4760346)

[Softwarekonzept 6](#_Toc4760347)

[1.6 Anforderungen 6](#_Toc4760348)

[1.7 Software/Hardware 6](#_Toc4760349)

[1.8 Mock-Up 6](#_Toc4760350)

[1.9 Menüleiste (noch nicht realisiert) 6](#_Toc4760351)

[1.10 Anzeigefenster DM/CM 7](#_Toc4760352)

[1.11 Haupt- und parasitäre Parameter 7](#_Toc4760353)

[1.12 Beschreibung Programmablauf 7](#_Toc4760354)

[2 Softwarestruktur 9](#_Toc4760355)

[2.1 Klassendiagramm 9](#_Toc4760356)

[3 Theoretische Grundlagen 10](#_Toc4760357)

[3.1 Einleitung EMI Filter 10](#_Toc4760358)

[3.2 Vorgehensweise Berechnung Einfügungsverluste 10](#_Toc4760359)

[4 Testkonzept 11](#_Toc4760360)

[4.1 Kontrolle bei der Entwicklung 11](#_Toc4760361)

[4.2 Überprüfung mit MATLAB 11](#_Toc4760362)

[4.3 Integrationstest 11](#_Toc4760363)

[4.4 Softwaretest 11](#_Toc4760364)

[5 Literaturverzeichnis 13](#_Toc4760365)

[6 Anhang 14](#_Toc4760366)

# Übersicht

## Ausgangslage

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

## Projektziele

Folgende Anforderungen wurden als Pflichtziele definiert. Diese sind notwendig und müssen beim Endprodukt komplett erfüllt werden.

|  |  |  |
| --- | --- | --- |
| **Nr.** | **Bereich** | **Beschreibung** |
| 1 | GUI-Struktur | * Das GUI ist schlicht und logisch * Programm als MVC Framework * GUI soll selbsterklärend sein |
| 2 | Art der Dateneingabe der Messdaten | Messwerteingabe über Textfelder und einstellbar über Slider |
| 3 | Berechnungen | Berechnungen müssen korrekte Ergebnisse liefern |
| 4 | Ausgabe der Messwerte | * Ausgabe der Resultate mit Graphen * 2-dimensionale Darstellung |
| 5 | Bedienbarkeit | Soll einfach zu bedienen sein |
| 6 | Design | Soll übersichtlich dargestellt werden |

Tabelle 1: Projektziele

## Wunschziele

Je nach zeitlichem Verlauf des Projektes können (müssen aber nicht) Zusatzfunktionen implementiert werden. Voraussetzung ist, dass das Projekt keine Verzögerungen erfährt.

|  |  |  |
| --- | --- | --- |
| **Nr.** | **Bereich** | **Beschreibung** |
| 1 | Ausgabedatei | Speichert die Graphen und Parameter in eine Pdf Datei ab. |
| 2 | Eingabedatei | Gespeicherte Datei kann wieder mit den eingegebenen Daten wiederhergestellt werden. (Save-Load-Option) |
| 3 | Programm | Animierte Funktionen (z.B. vibrierende Taster bei falscher Eingabe, etc.)  Eine Funktion, bei der sich der Wert beim Slider selbst immer wieder verändert (Sweeping) und man die Veränderung graphisch sieht. |

Tabelle 2: Wunschziele

## Nicht-Ziele

Es gibt auch einige Ziele, welche bewusst nicht zu erreichen sind. Diese würden den Rahmen des Projektes sprengen, da sie zu zeitintensiv sind.

|  |  |  |
| --- | --- | --- |
| **Nr.** | **Bereich** | **Beschreibung** |
| 1 | Darstellung | 3-dimensionale Darstellung |
| 2 | Eingabefelder | Zusätzliche Eingabefelder für gewünschte Kurve. Dabei werden die optimalen Bauteile dazu berechnet |
| 3 | Webapplikation | Tablet-Version und Smartphone Version |
| 4 | Simulation | Einberechnung der Permeabilität der Spule, in Abhängigkeit der Frequenz für die Einfügungsdämpfung. |

Tabelle 3: Nicht-Ziele

## Lieferobjekte

Die Lieferobjekte sind in der unten aufgeführten Tabelle festgelegt. Die Berichte werden per Mail an die Herren Dalessandro, Niklaus, Gaulocher und Buchschacher geschickt.

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Lieferobjekte** |  |  |
| 28.02.2019 | KIS | | |
| 24.03.2019 | Pflichtenheft Version 1 | | |
| 31.03.2019 | Statusbericht 1 | | |
| 07.04.2019 | Pflichtenheft Endversion | | |
| 11.04.2019 | Zwischenpräsentation | | |
| 02.05.2019 | Disposition, Einleitung und Statusbericht 2 | | |
| 05.05.2019 | Statusbericht 2 | | |
| 06.05.2019 | Berechnungen mit Matlab | | |
| 10.05.2019 | Beta Version der Software | | |
| 19.05.2019 | Statusbericht 3 | | |
| 13.06.2019 | Statusbericht 4 | | |
| 13.06.2019 | Fachbericht und Präsentation | | |

Tabelle 4: Lieferobjekte

# Softwarekonzept

## Anforderungen

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.

## Software/Hardware

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.

## Mock-Up



Abbildung 10: GUI Übersicht

Die grafische Bedienoberfläche besteht voraussichtlich aus einer Menüleiste, einem Anzeigefenster für jeweils DM und CM, Haupt- und parasitär Parameter.(Abbildung X) Diese Bestandteile werden in den folgenden Unterkapitel genauer beschrieben.

## Menüleiste (noch nicht realisiert)

Von hier aus haben Nutzende Zugriff auf verschiedene gröbere Einstellungen und Funktionen. Per Mausklick sollen Daten gespeichert, geladen oder ausgedruckt werden können.

## Anzeigefenster DM/CM

Hier werden die CM- und DM-Einfügungsverluste als voneinander getrennte Kurvendiagramme dargestellt. Eine logarithmische Skalierung der Frequenzachse ermöglicht die Darstellung der Verluste innerhalb eines Frequenzbereichs von 1Hz bis 30MHz. Pro Diagramm können auch mehrere Kurven gezeichnet werden.

## Haupt- und parasitäre Parameter

****

Abbildung 11: Parametereinstellungen

Jeder Hauptparameter wird durch eine gewisse Anzahl parasitärer Parameter dargestellt. Letztere können einzeln via Schieberegler um maximal ±30% von ihrem ursprünglichen Wert verschoben werden. Jede Veränderung eines Schiebereglers bewirkt eine erneute Darstellung der Kurvendiagramme. Oberhalb der Regler werden die momentanen Werte der parasitären Parameter aufgezeigt. Siehe Abbildung X

## Beschreibung Programmablauf

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.

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. Siehe Abbildung X. 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 X)



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 X)



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. In der Abbildung X als blaue Kurve dargestellt. Somit können die unterschiedlichen Auswirkungen aller parasitären Parameter schnell eingesehen werden, ohne dass die Nutzenden die Schieberegler einzeln verschieben müssen.

# Softwarestruktur

In der Abbildung X ist das Klassendiagramm der Software ersichtlich.

## Klassendiagramm



Abbildung 15: Klassendiagramm

# Theoretische Grundlagen

## Einleitung 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.

## 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 vereinfachte Schaltung ist in der Abbildung X sichtbar



Abbildung 7: Vereinfachte CM-Schaltung

Mit der vereinfachten Schaltung bestimmen wir für alle Bauteile die Impedanzen für eine bestimmte Frequenz. Danach berechnen wir die Impedanz (Z), sowie die Admittanz (Y0). Daraus ergeben sich Zweitore. Daraus ergeben sich die Kettenmatrix (A1) für die Längsimpedanz (Z) sowie (A2) für die Queradmittanz (Y0).

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

Nun können wir die Einfügungsdämpfung a in dB berechnen:

Dieser Parameter wird mit verschiedenen Frequenzen berechnet, damit man die Einfügungsdämpfung über dem ganzen Spektrum darstellen kann.

# Testkonzept

Das Testkonzept beschreibt Methoden, welche für die Überprüfung des Programms verwendet werden. Das Programm soll am Ende alle Anforderungen des Auftraggebers erfüllen und einwandfrei funktionieren. Ebenfalls sollte das Programm benutzerfreundlich und selbsterklärend sein. Damit alle Forderungen erfüllt werden können, müssen alle Teilbereiche des Produkts fortgehend kontrolliert und getestet werden.

## Kontrolle bei der Entwicklung

Damit keine schwerwiegenden Fehler auftreten, wird das Programm während der Entwicklung schon getestet. Die erkannten Fehler sollen dabei korrigiert und protokolliert werden. Dies hat den Vorteil, dass keine grossen Zeitverluste durch Fehlerbehebungen entstehen.

## Überprüfung mit MATLAB

Die Richtigkeit der Berechnungen und der Rückgabewerte der Software werden mit MATLAB überprüft. Dabei müssen die Werte in allen Bereichen übereinstimmen. Ausserdem werden die Kurven von MATLAB mit Simulationen der Filterhersteller verglichen.

## Integrationstest

Die verschiedenen Klassen und Methoden werden isoliert geprüft. Damit kann sichergestellt werden, dass alle Klassen/Methoden die vorgesehenen Rückgabewerte und Aktionen korrekt ausführen. Soweit die Prüfungen mit den erwarteten Werten übereinstimmen, werden die Klassen/Methoden in das Programm integriert.

## Softwaretest

Die Software wird auf verschiedene Weisen und Ebenen getestet. Das Programm sollte allen Anforderungen des Auftraggebers gerecht werden und die im Auftrag beschriebenen Anforderungen erfüllen. Dafür kontrollieren die Teammitglieder, ob die Benutzeroberfläche komplett ist und das Programm so funktioniert wie erwünscht. Falls auffällt, dass etwas fehlt oder nicht in Ordnung ist, wird dies dann beseitigt.

Durch regelmässige Absprache mit dem Auftraggeber, soll verhindert werden, dass der Auftraggeber mit der Benutzeroberfläche unzufrieden ist. Der regelmässige Austausch soll dafür sorgen, dass das Programm mit den Vorstellungen des Auftraggebers übereinstimmt.

Fehlerhafte Eingaben sollen vom Programm erkannt und bewältigt werden. Dafür wird durch absichtliche Falscheingaben vom Projektteam überprüft, ob das Programm richtig reagiert. Sollte dies nicht der Fall sein, müssen die Probleme schnellstmöglich behoben werden.

Für die Prüfung der Benutzerfreundlichkeit wird eine Prüfung durch Dritte gemacht. Das Projektteam kann anhand der Feedbacks Dritter nötige Anpassungen machen. So können eventuelle Verbesserungsmöglichkeiten bezüglich der Handhabung gefunden und das Programm verbessert werden.

Damit das Programm auf den Betriebssystemen MacOS und Windows einwandfrei funktioniert, wird innerhalb des Projektteams das Programm auf beiden Systemen getestet. Dies soll verhindern, dass das Programm nur auf einem Betriebssystem laufen kann.

# Literaturverzeichnis

**Im aktuellen Dokument sind keine Quellen vorhanden.**

# Anhang

Diese MATLAB-Datei berechnet für einen Frequenzvektor (1kHz – 10GHz) die Einfügungsverluste und stellt diese logarithmisch dar.

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

% CM Mode

% Luca Krummenacher

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

f = logspace(4,8,100); %Frequenzvektor von 10^4 - 10^8 Hz

for k=1 : length(f)

H(k) = -20 \* log10(abs(get\_s21(f(1,k))));

end

semilogx(f,H) %Logartihmische Skala verwenden

grid on

title('Insertion Loos')

xlabel('Frequency [Hz]')

ylabel('Insertion Loos [db]')

Die Funktion get\_s21 gibt für eine bestimmte Frequenz den S21 Parameter zurück. Die Bauteilparameter sind in dieser Funktion bestimmt.

function [result] = get\_s21(frequency)

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

% Projekt: s21 Parameter berechnen

% fuer die CM-Schaltung eines EMI-Filters

% Autor: Luca Krummenacher

% Datum: 13. Maerz

% Ort : FHNW Brugg-Windisch

% Version: 1.0

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

% Gibt den s21 parameter fuer eine bestimmte Frequenz zurück

% Die Bauteilparameter sind in dieser Funktion bestimmt

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Rref = 50; %50 Ohm Eingangs und Lastwiderstand (Bezugswiderstand)

Rw = 11.3e-3;

Lr = 9.2e-6;

C1C2 = 12.93e-12;

Rp = 7.92e3;

L0 = 0.5e-3;

f = frequency;

w = 2\*pi\*f;

ZZtot = 0.5\*Rw + 0.5\*Lr\*w\*j + rpara(L0\*w\*j,rpara(0.5\*Rp,(1/(j\*w\*C1C2))));

%ZY aus vereinfachter Schaltung:

Ry1 = 246e-3;

Cy1 = 3.85e-9;

Ly1 = 8e-9;

Ry2 = 100.1e-3;

Cy2 = 4.67e-9;

Ly2 = 10.1e-9;

ZY1 = Ry1 + Ly1\*w\*j + (1/(j\*w\*Cy1));

ZY2 = Ry2 + Ly2\*w\*j + (1/(j\*w\*Cy2));

ZYtot = rpara(ZY1,ZY2);

%A Matrizen aus Impedanzen berechnen:

%Laengsimpedanz A1:

A1 = [1 ZZtot; 0 1];

%Querimpedanz A2:

A2 = [1 0; (1/ZYtot) 1];

%Gesamtmatrix berechnen

A = A1 \* A2;

%s21 berechnen

result = 2 / ( A(1,1) + (A(1,2)/Rref) + (A(2,1)\*Rref) + A(2,2) );

end