**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, 14. Juni 2019

**Abstract**

*Jedes Elektrogerät enthält heutzutage einen Filter. Der Filter dient dem Gerät, um andere Geräte nicht zu stören und umgekehrt. Dieser Filter besteht aus verschiedenen realen Bauteilen, welche einen Einfluss auf die Dämpfungen des Filters haben. Für die Herstellung von solchen EMI Filtern ist es sinnvoll, den Einfluss der einzelnen Bauteile zu kennen. Dafür eignet sich ein Tool, in dem man die Werte der verschiedenen Bauteile verändern kann und die Veränderung der Einfügedämpfungen graphisch darstellt. Die Firma Schaffner Group produziert EMI Filter und hat der Fachhochschule Nordwestschweiz den Auftrag zu dieser Arbeit gegeben.*

Das Ziel dieser Arbeit ist es, eine Software zu programmieren, welche eine Sensitivitätsanalyse der Einfügedämpfung eines EMI-Filters ermöglicht. Dazu sollen die einzelnen Parameter eines Bauelements um ± 30 % verstellt werden können, die Veränderung des Filterverhaltens soll in einer Einfügedämpfung-Frequenz-Kurve ersichtlich sein. Ausserdem müssen die Schaltungen des Filters analysiert werden und so vereinfacht werden, dass die Berechnungen möglichst einfach in die Software implementiert werden können. Die Darstellung der Software soll in Form eines «DJ-Mischpults» sein. Die Werte der Parameter der Schaltung sollen mit einem Slider und durch nummerische Eingabe verändert werden können. Zusätzlich soll die Software die beiden Graphen und die Schemas der Schaltung aufzeigen.

Für die Software eignet sich objektorientierte Programmierung in Java. Die Java-Spezifikation JavaFX simplifiziert zusätzlich die Programmierung der interaktiven Benutzeroberflächen, indem das Design klar von den Berechnungen getrennt wird. Die Berechnungen sind erst mit Matlab gemacht worden. *Um den Auftrag angemessen ausführen zu können, sind die elektrotechnischen Aspekte hinter den Störungsverhalten eines EMI-Filters erarbeitet und anschliessend in die Software implementiert worden.*

Beim Aufstarten des Programms erscheinen zwei Diagramme mit Graphen zu den jeweiligen Störungsarten. Durch «Doppelklick» auf einen der beiden Graphen soll dessen Ansicht vergrössert oder wieder verkleinert werden. Darunter befindet sich das Mischpult, welches jedem Parameter des Filters einen Slider zuordnet. Wenn mit der Maus über einen Slider gefahren wird, werden die dazugehörigen Abweichungen von ± 30% im Graphen farbig angezeigt. Durch Anklicken der Slider, passen sich die Werte des Parameters und die Kurve an. Die Buttons oben rechts in den Sliderfeldern ermöglichen den Wechsel zur Textfeldanzeige, in denen dann die Werte der einzelnen Komponenten manuell in ENG- Format eingegeben werden können. Die Eingabe wird anschliessend übernommen und die resultierende Kurve dargestellt.

In der Menüleiste können *noch in Arbeit was man genau alles kann.* Unterhalb der Menüleiste kann zwischen verschiedenen Tabs gewechselt werden. Im ersten Tab sieht man die Graphen der beiden Störungsarten. Im zweiten Tab sind die Schaltbilder mit den Bezeichnungen der Komponenten ersichtlich.

Inhaltsverzeichnis

[1 Einleitung 4](#_Toc10196595)

[2 Theoretische Grundlagen 5](#_Toc10196596)

[2.1 EMI Filter 5](#_Toc10196597)

[2.2 Störungsarten 5](#_Toc10196598)

[2.3 Definition Einfügedämpfung «Insertion loss» 6](#_Toc10196599)

[2.4 Parasitäre Parameter 7](#_Toc10196600)

[3 Software 9](#_Toc10196601)

[3.1 Benutzeroberfläche 9](#_Toc10196602)

[3.2 MVC 9](#_Toc10196603)

[3.3 View-Klassen 10](#_Toc10196604)

[3.3.3 Eingabe der Daten 10](#_Toc10196605)

[3.4 Controller-Klassen 11](#_Toc10196606)

[3.4.1 Laden und Speichern von Daten 12](#_Toc10196607)

[3.5 Model-Klassen 12](#_Toc10196608)

[3.6 Implementierung der Berechnungen in JAVA 12](#_Toc10196609)

[3.6.1 Architektur 12](#_Toc10196610)

[4 Elektrotechnik 13](#_Toc10196611)

[4.1 Vereinfachung der CM-Schaltung 13](#_Toc10196612)

[4.2 Vereinfachung der DM-Schaltung 15](#_Toc10196613)

[4.3 Vorgehensweise Berechnung Einfügungsverluste 18](#_Toc10196614)

[4.4 S-Parameter 19](#_Toc10196615)

[4.5 Realisierung mit Matlab 21](#_Toc10196616)

[5 Validierung 22](#_Toc10196617)

[5.1 Berechnungen 22](#_Toc10196618)

[5.2 Software 23](#_Toc10196619)

[5.2.1 Eingabe 23](#_Toc10196620)

[5.2.2 Benutzerfreundlichkeit 23](#_Toc10196621)

[6 Schlusswort 24](#_Toc10196622)

[Quellenverzeicnis 25](#_Toc10196623)

[Anhang 26](#_Toc10196624)

# Einleitung

Damit Elektrogerätekeine Störungen ins Netz aussenden und nicht selber von diesen beeinflusst werden, enthalten Geräteeinen EMI (elektromagnetische Interferenzen) Filter. Jedes Bauteil des Filters hat wiederum unterschiedliche Auswirkungen auf die Einfügedämpfung eines Filters. Damit die Firma Schaffner, welche solche Filter produziert, die Einflüsse der verschiedenen Parameter auf die Einfügedämpfung simulieren kann, wird die entsprechenden Dämpfung graphisch dargestellt.

Der Auftrag des Moduls pro2E des Studiengangs Elektro und Informationstechnik an der Fachhochschule Nordwestschweiz ist es, ein Anwendungsprogramm zu entwickeln, welche es ermöglicht die Einfügedämpfung eines EMI-Filters in Abhängigkeit zur Frequenz darzustellen. Anhand eines GUIs, welches ein Mischpult mit Schiebereglern enthalten soll, lassen sich die Werte der Parameter beliebig verändern. Anschliessend werden die Einflüsse auf das Verhalten in einem Graphen aufgezeigt. Die Berechnungen zu den Graphen sind in der Software implementiert und sorgen für eine korrekte blabla Darstellung.

Gemäss Aufgabenstellung (im Anhang) soll das GUI gebrauchstauglich (nach DIN EN ISO 9241-11) sein. Somit ist ein Ziel dieser Arbeit, dass ein Laie das Programm ohne Erklärung und ohne Anleitung bedienen kann. Ausserdem soll die Softwareanwendung auf den Betriebssystemen MacOS (ab Version 10.11) und auf Windows (7 oder neuer) problemlos funktionieren. Die Ziele der Berechnungen sind, dass die Simulationen der Schaltungen mit deren Berechnungen übereinstimmen und korrekt sind.

Das Programm wird in Java geschrieben und basiert auf dem Model-View-Controller-Framework. Damit die Software später einfach verändert werden kann, ist sie 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, eignet sich JavaFX, welches Layouteinstellungen mit Hilfe von Cascading Style Sheets (CSS) schlang und einfach hält. Für die Berechnung der Graphen liest das Model die Daten aus und berechnet die Kurven. Der Controller verbindet die View und das Model miteinander.

Die Basis für die Berechnungen bilden die Schemata, welche von Schaffner zur Verfügung gestellt wurden. Für die Berechnungen der Gleichtakt- und Gegentaktstörungen sind die Schemata vereinfacht worden und die darin enthaltenen Bauelemente sind in Längs- und Querimpedanzen eingeteilt. Diese sind anschliessend zu einer Gesamtmatrix zusammengeführt worden. Mit Hilfe von Matlab berechnen sich daraus die Einfügungsverluste und stellt diese in einem Kurvendiagramm dar.

Um in die fachlichen Kapitel einzuleiten, werden in diesem Bericht als erstes die Grundlagen erläutert. Danach wird im Kapitel «Elektrotechnik» die Theorie des EMI- Filters und die Vorgehensweise bei den Berechnungen genauer aufgezeigt. Die Softwarestruktur und die Bedienoberfläche sind im Kapitel XY erklärt. Anschliessend folgt die Überprüfung der Berechnungen und der Software.

# 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 (Abbildung 1) 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 mithilfe der X-Kondensatoren gedämpft.

Der Widerstand, welcher parallel zu CX-1 liegt, wird aus Sicherheitsgründen benötigt. Dieser ist typischer Weise 1 MΩ gross. Er entlädt den CX-1 Kondensator, nachdem das Filter vom Netz getrennt wird. [1]

## Störungsarten

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

Gleichtaktstörungen wirken auf beide Leitungen gleichermassen ein. Die Störspannungen besitzen in beiden Leitungen die gleiche Amplitude, sowie Phasenlage. CM Störungen entstehen oft durch kapazitive Kopplung, wie in Abbildung 2 sichtbar ist. Diese entstehen aufgrund von verschiedenen Potentialen entlang des Übertragungsweges. Damit das Signal wirklich gestört wird, ist ein langer Stromweg nötig. Dies ist oft der Fall bei gemeinsamen Bezugssignalen, zum Beispiel einer gemeinsamen Masse oder Erde.

Die Gegentaktstörungen fliessen in die genau gleiche Richtung, wie die Nutzsignale. Das heisst sie Überlagern das eigentliche Signal. Die Ursache bei diesen Störungen kann eine induktive Kopplung sein. Dabei beeinflussen benachbarte Wechselstrom Signale aufgrund ihres Magnetfeldes die Nutzleiter. Der Leiter erzeugt mit seinem Magnetfeld in der gestörten Schaltung eine Spannung, welche sich als zusätzliche Quellspannung verhält. Diese ist eine Ursache für das Gegentaktrauschen. In der Abbildung 2 sind die Störungen innerhalb einer Schaltung dargestellt.

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

## Definition Einfügedämpfung «Insertion loss»

Die Leistung eines EMI Filters wird mit den Einfügedämpfung 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: (Abbildung 4)

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 in Abbildung 5, die parasitären Parameter gemäss folgenden Ersatzschemas in Abbildung 6 für die CM-Schaltung. Interessant dabei ist der 1MΩ Widerstand, welcher parallel zu CX2 liegt. Dieser wird aus Sicherheitsgründen benötigt, um den CX2 Kondensator zu entladen. In der Funktion des Filters hat er jedoch keinen Einfluss, weshalb er nicht in unsere Simulationen und Berechnungen einfliesst.



Abbildung 5: CM-Ersatzschaltung ohne parasitäre Parameter



Abbildung 6: CM-Ersatzschaltung mit parasitären Parametern

Für die DM-Ersatzschaltung in Abbildung 7 hat man ebenfalls mit parasitären Parametern ergänzt. Der 1MΩ Widerstand fällt genau wie bei der CM-Schaltung weg. Diese Schaltung ist in der Abbildung 8: DM-Ersatzschaltung mit parasitären Parametern ersichtlich.



Abbildung 7: DM-Ersatzschaltung ohne parasitäre Parameter



Abbildung 8: DM-Ersatzschaltung mit parasitären Parametern

# Software

In diesem Kapitel wird das Softwaretool ”EMJ” vorgestellt.

BLBLBLBLBLBLLBLBLB

## Benutzeroberfläche

*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.*



*BLBLLBLB*

*BLBLBLLBLB*

## 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.



## View-Klassen

Die View ist für das GUI verantwortlich.

BBLLBLBLBLBLBL

## 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 9: Anpassung der Parameter

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

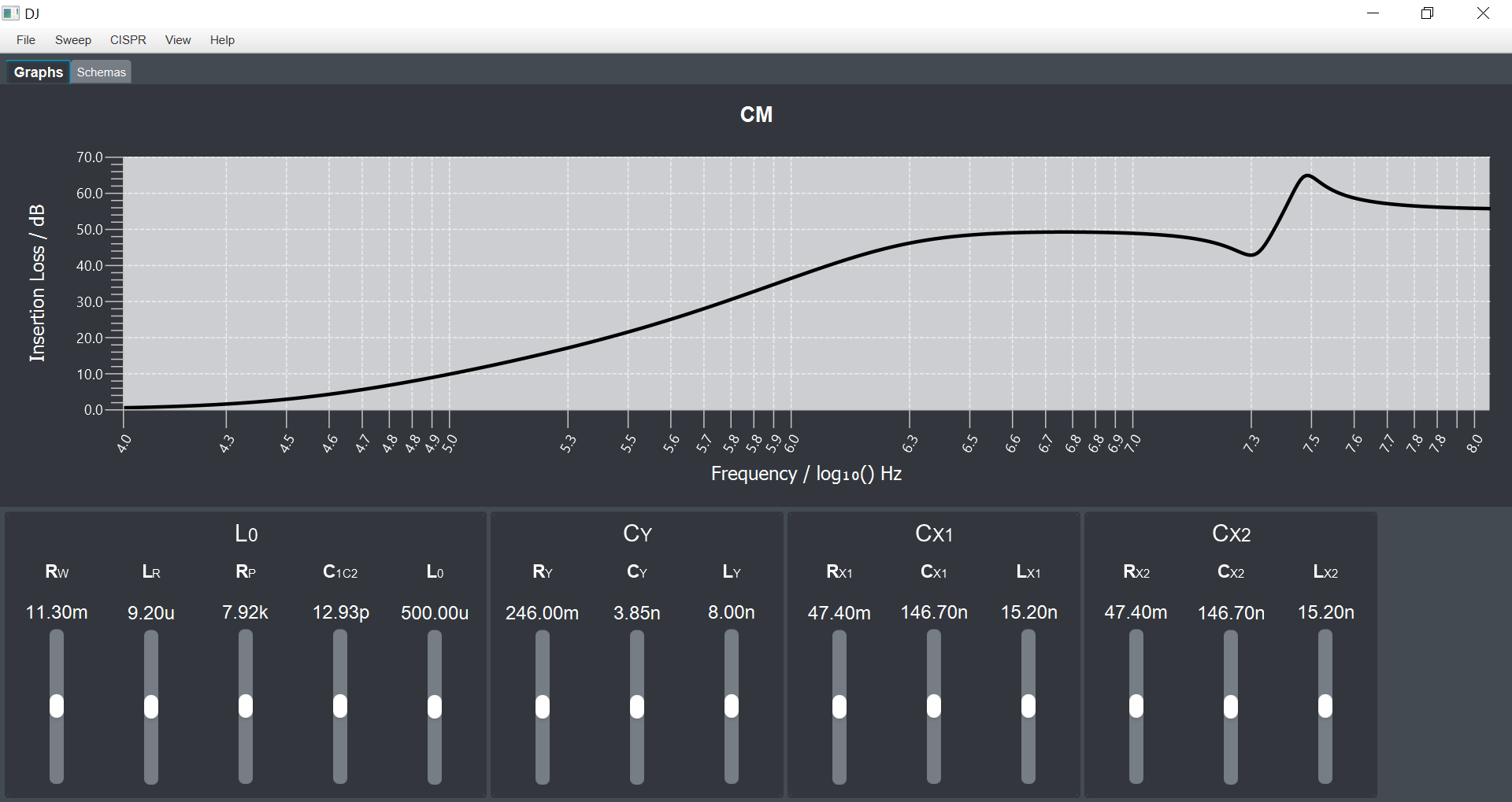


Abbildung 10: Kurvenansicht

*Bei Doppelklick auf ein Diagramm, übernimmt dieses die ganze Breite des Fensters und kann so genauer angesehen werden, wie man in Abbildung 10 sehen kann. Bei erneutem Doppelklick wird der Vorgang rückgängig gemacht, so dass wieder beide Diagramme sichtbar sind.*



Abbildung 11: Parameterauswirkung

*Wenn Nutzende den Mauszeiger über einen Schieberegler bewegen, wird im Diagramm gezeigt, 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, was man in Abbildung 11 sehen kann.*

## Controller-Klassen

LLBLBLBLBLBLBLBLBLBLBLALBALBALBALBL

### 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.

Der Explorer wird gestartet, wenn diese Befehle betätigt werden. Darin muss man einen eigenen Dateinamen reinschreiben. Bei der Dateiformation kann nur die Form txt, rtf oder pdf ausgewählt werden.

Um die Daten 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.

Diese Lade und Speichern-Funktionen wurden im Controller implementiert.

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

## Model-Klassen

BLBLBLBLLBLBLBLB

LBLBLBLBLBLB

## Implementierung der Berechnungen in JAVA

BLLBLBLBLBLBL

### 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.*

## Validierung der Software

Damit sich das Projektteam vergewissern kann, ob alle Ziele erreicht wurden, wurde eine Validierung durchgeführt. Im Pflichtenheft wurde beschrieben, dass die Software alle im Auftrag beschriebenen Anforderungen erfüllen soll, dazu ist regelmässiger Kontakt mit dem Auftraggeber gepflegt worden. In Absprache mit dem Auftraggeber funktioniert die Software nun wie im Auftrag beschrieben.

*.*

Das Verhalten der Software bei fehlerhaften Eingaben wurden intern im Team geprüft. Dafür wurden von Teammitgliedern Fehleingaben gemacht und getestet, wie die Software reagiert. Dann wurde sie so präpariert, dass sie bei Fehleingaben reagiert blabla und die Software auch bei Fehleingaben dem Benutzer möglichst keine Probleme macht.

Für die Benutzerfreundlichkeit schien eine Prüfung durch Dritte die sinnvollste Validierung. Um dies zu ermöglichen, wurde die Software xx Aussenstehenden vorgeführt. Dabei handelte es sich um Ingenieure und um Laien. Um die Zufriedenheit der Testenden auszuwerten, wurde eine kleine Umfrage erstellt, welche sie nach der Vorführung ausfüllen durften. Gemäss Umfrage…

Zusätzlich ist das Programm in der Lage auf den Betriebssystemen MacOS und Windows zu laufen. Dies wurde durch die Verwendung von IntelliJ unterstützt

# Elektrotechnik

Für die Berechnungen sind die Schaltungen mit den parasitären Elementen von Interesse, welche bereits in den theoretischen Grundlagen der Grundschaltung ergänzt wurden. Danach wurden die beiden Schaltungen gemäss den Regeln der Elektrotechnik vereinfacht, so dass Zweitore gebildet werden können. Daraus berechnen sich die S-Parameter. Mit diesen kann schlussendlich die Einfügedämpfung berechnet werden. Nachfolgend werden alle Schritte erklärt.

## Vereinfachung der CM-Schaltung



Abbildung 12: CM Grundschaltung

Für die Vereinfachung der Schaltung gehen wir von der Grundschaltung in Abbildung 12 aus. Die CM-Störungen auch symmetrische Störungen genannt, fliessen in beiden Leitern in die gleiche Richtung. Das heisst die beiden Leiter liegen auf demselben Potential, somit haben die Bauteile zwischen den beiden Leitern (CX1, CX2, RX1, RX2, LX1, LX2, C4, C3) keinen Einfluss. Nun können die beiden Leiter zusammengefasst werden, da beide gegenüber Masse symmetrisch sind. Das Zusammenfassen der beiden Leiter entspricht einer Parallelschaltung der beiden Leiter. Deshalb ergeben sich folgende Faktoren für die vereinfachte Schaltung: Der parasitäre Widerstand Rw und die parasitäre Spule Lr halbieren ihre Werte, genau wie der Widerstand Rp1. Durch die Parallelschaltung der Kondensatoren C1 und C2 addieren sich bekanntlich die beiden Kapazitäten zu einer Gesamtkapazität C1 + C2. Die beiden Spulen L0 und L1 sind auf einem Ferritkern aufgewickelt. Da beide Spulen den gleichen Wicklungssinn besitzen, verdoppelt sich die Induktivität L0. Durch das Zusammenfassen der beiden Spulen halbiert sich ihr Wert wieder, woraus der ursprüngliche Wert der Spule resultiert. Die beiden Kondensatoren CY1 und CY2 mit ihren parasitären Elementen bleiben unbeeinflusst, da sie bereits gegen Masse geschalten sind. Nach diesen Vereinfachungen ergibt sich die Schaltung in Abbildung 12.



Abbildung 13: Vereinfachte CM Schaltung

Die Validierung der Umformungen wurde mit zwei Simulationen in «MPLAB» bewerkstelligt. Im folgenden Diagramm (Abbildung 14) liegen die beiden Ergebnisse der Simulationen übereinander. Darin ist sichtbar, dass die Vereinfachung zum selben Ergebnis führt wie die Grundschaltung. Die beiden Schaltungen dazu sind im Anhang hinterlegt.



Abbildung 14: Vergleich Grundschaltung / Vereinfachte Schaltung

## Vereinfachung der DM-Schaltung



Abbildung 15:Grundschaltung DM

Die Grundschaltung der DM Schaltung ist in der Abbildung 15 sichtbar. Die DM-Störungen, auch asymmetrische Störungen genannt, fliessen in einem Leiter hin und im anderen Leiter wieder zurück. Daher lassen sich die Bauelemente nicht so leicht vernachlässigen wie in der CM Schaltung. Der Ringkern mit den beiden Spulen L0 und L5 gekoppelt sind, hat in der DM-Schaltung keinen Einfluss, da sich die beiden Induktivitäten aufgrund des entgegengesetzten Feldes aufheben. Dadurch heben sich auch die parasitären Elemente (C1, C2, Rp, Rp1) auf. Es bleiben nur noch die die parasitären Parameter Rw und Lr der Spulen übrig. Damit wir wie auch schon in der CM Schaltung die beiden Leiter zusammenfassen können, müssen wir die Bauteile zwischen den beiden Leitern verdoppeln. Dafür werden die Widerstände und Spulen halbiert und die Kondensatoren verdoppelt. Dies betrifft auch den Innen- und den Lastwiderstand. Dabei wird die Funktion der Schaltung nicht verändert. Die 1:1 Trafos werden dabei vernachlässigt, da diese als ideal angenommen werden und im Filter nicht als Bauteil vorkommen. Die Kondensatoren C3 und C4 werden weggelassen, da sie einen vernachlässigbaren Einfluss auf die Schaltung haben. Dies wurde vom Auftraggeber akzeptiert. In Abbildung 16 sieht man die angepasste Schaltung.



Abbildung 16: DM Schaltung angepasst

Bei der blau eingezeichneten Achse wurde die Schaltung aufgetrennt und gegen Masse gelegt (Abbildung 16). Dies ist möglich, da die gleichen Störungen in Hin- und Rückleiter fliessen und man diese halbieren kann. Die parasitären Bauteile der Spule (Lr und Rw) werden zusammengefasst, das heisst sie werden beide halbiert. Einer der beiden Y-Kondensatoren fällt weg. Dabei spielt es keine Rolle welcher, da die beiden immer gleichgross sind. Deshalb muss man annehmen, dass die parasitären Werte davon gleichgross sind. In der Software kann man deshalb auch nur einen Y-Kondensatorwert verstellen. Die vereinfachte Schaltung, welche zur Berechnung verwendet wurde, ist in Abbildung 17 sichtbar.



Abbildung 17: Vereinfachte DM Schaltung

Die Validierung der Umformungen wurde mit zwei Simulationen in «MPLAB» bewerkstelligt. Im folgenden Diagramm (Abbildung 18) liegen die beiden Ergebnisse der Simulationen übereinander. Darin ist sichtbar, dass die Vereinfachung zum selben Ergebnis führt wie die Grundschaltung. Die beiden Schaltungen dazu sind im Anhang hinterlegt.



Abbildung 18: Vergleich Simulationen

## Vorgehensweise Berechnung Einfügungsverluste

Wir befassen uns einfachheitshalber mit der CM Schaltung, um das Prinzip der Berechnungen zu erklären. Bei der DM Schaltung wird wiederum genau die gleiche Vorgehensweise angewendet. Jedoch werden die Zweitore anders aufgeteilt. In Abbildung 19 wurde die CM Schaltung in Zweitore unterteilt und beschriftet.

Z1

Z2

Z3



Abbildung 19: Schaltung CM mit Einteilung in Zweitore

Zuerst wird die Gesamtimpedanz (Z1), sowie (Z2) und (Z3) bestimmt.

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

Nachdem die Impedanzen in Zweitore zusammengefasst wurden, kann man diese in Kettenmatrizen umwandeln. Dies funktioniert für die unterschiedlichen Zweitore folgendermassen:





Wir berechnen nun die Kettenmatrix (A1) für die Längsimpedanz (Z1) sowie (A2) für die Querimpedanz (Z2) und (A3) für (Z3).

Für den S21-Parameter benötigen wir die Gesamtmatrix, diese berechnet man mit Hilfe der Kettenschaltung, in dem man die drei Matrizen miteinander multipliziert. Dabei ist es wichtig, dass zuerst A1 mit A2 multipliziert wird und danach mit A3. [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 Zweitor 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 20: 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 (R1 und R2), welche bei uns am Eingange, sowie am der Last 50 Ω betragen. Daraus berechnen wir den Streuparameter folgendermassen:

: Wert der Gesamtmatrix an bestimmter Position.

: Eingangswiderstand

: Lastwiderstand

Damit werden schlussendlich die Einfügungsverluste a in dB berechnet:

## Realisierung mit Matlab

Bevor die Berechnungen in Java implementiert wurden, sind die Einfügungsverluste in Matlab berechnet worden. Dies wurde gemäss der Vorgehensweise in 4.3 bewerkstelligt. [4] [5]



Abbildung 21: Vergleich Berechnung / Simulation



Abbildung 22: Insertion loss Kurve aus MATLAB

Der Vergleich der beiden Kurven (Abbildung 21, Abbildung 22) zeigt, dass die Berechnungen mit der Simulation übereinstimmten. Der Matlab Code für die Berechnungen ist im Anhang vorhanden.

## Validierung der Elektrotechnik

Die Implementation der Berechnung ist im Softwareteil dokumentiert. Um zu überprüfen, ob die Berechnungen in Java mit den Berechnungen in Matlab übereinstimmen, wurden die beiden Kurven (Abbildung 23, Abbildung 24) wiederum übereinandergelegt und verglichen.



Abbildung 23: Vergleich Java / Matlab CM



Abbildung 24: Vergleich Java / Matlab DM

Dabei geht hervor, dass die Berechnungen korrekt in die Software implementiert wurde.

# 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 das Verstellen der Bauteilwerte dessen Einfluss im Diagramm beobachten und vorhersagen können. Auch können eingegebene Werte abgespeichert werden und für die spätere Verwendung wieder ins 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.*

Was wurde im Projekt erreicht? Was wurde übertroffen?

Wo besteht Verbesserungsbedarf? Was (und warum) konnte nicht realisiert werden

Optimierungs/Weiterentwicklungsmöglichkeiten

# Quellenverzeicnis

Mößlang, B. (2016). *EMC Filter Insertion Loss Simulation.* Reutlingen: OMICRON Lab.

Maier, A., & Rupp, S. (2016). *Hochfrequenztechnik, Teil 2 - Anwendungen.* Stuttgart: DHBW.

Niklaus, P. (2018). *Zweitore (2-Tore).* Windisch: FHNW.

Schaffner. (2018). *Single-Stage Filters FN 2020.* Luterbach: Schaffner Group. Abgerufen am 14. März 2019 von https://www.schaffner.com/products/emcemi/

Schaum, A. (2017). *Zweitore - Zusammenfassung.* Universität zu Kiel: Lehrstuhl für Regelungstechnik.

# Anhang

**Schaltungssimulationen**

Für die Simulationen in MPLAB wurde immer der gleiche Schaltungsaufbau verwendet. Dabei wurde mit einer Wechselspannungsquelle (V1) eine Wechselspannung mit verschiedenen Frequenzen angelegt. In der Blackbox wurden jeweils die entsprechenden Schaltungen eingefügt. Zur Auswertung wurde Ein Bodesimulator verwendet. Dieser misst am Eingang die Spannung, welche nach dem Filter über dem Lastwiderstand abfällt und am Ausgang die Spannung, welche nur über dem einen Widerstand des Spannungsteilers abfällt. Somit kann man das Ergebnis mit den Einfügungsverlusten vergleichen, welche gleich definiert sind.



**Blackbox**

**Schaltung**

**MATLAB Code**

**CM\_DM.m**

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

% Projekt: Plot CM and DM mode

% Autor: Luca Krummenacher

% Datum: 6. Mai 2019

% Ort : FHNW Brugg-Windisch

% Version: 2.0

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

% used functions: get\_s21\_dm\_4.m, get\_s21\_CM.m,

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

clear;

clc;

clf; %clear figures

f = logspace(4,8,1000); %Frequenzvektor von 10^4 - 10^8 Hz

for k=1 : length(f)

H(k) = -20 \* log10(abs(get\_s21\_dm\_4(f(1,k))));

H2(k) = -20 \* log10(abs(get\_s21\_CM(f(1,k))));

end

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

% Plot DM

subplot(2,1,1)

semilogx(f,H,'b') %Logartihmische X-Achse verwenden

grid on

title('Insertion Loss DM')

xlabel('Frequency / Hz')

ylabel('Insertion Loss / db')

legend("DM",'Location','bestoutside')

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

% Plot CM

subplot(2,1,2)

semilogx(f,H2,'r')

grid on

title('Insertion Loss CM')

xlabel('Frequency / Hz')

ylabel('Insertion Loss / db')

legend("CM",'Location','bestoutside')

adjust\_fig

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

**Funktion get\_s21\_dm\_4.m**

function [result] = get\_s21\_dm\_4(frequency)

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

% Projekt: s21 Parameter berechnen

% fuer die DM-Schaltung eines EMI-Filters

% Autor: Luca Krummenacher

% Datum: 18. April

% Ort : FHNW Brugg-Windisch

% Version: 4.0

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

% Gibt den s21 parameter fuer eine bestimmte Frequenz zurück

% Die Bauteilparameter sind in dieser Funktion bestimmt

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

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

Rref1 = Rref/2;

Rref2 = Rref/2;

%für L0

Lr = 9.2e-6;

C1 = 6.46e-12;

C2 = C1;

Rw = 11.3e-3;

Ry1 = 246e-3;

Cy1 = 3.85e-9;

Ly1 = 8e-9;

Rx1 = 47.4e-3;

Lx1 = 15.2e-9;

Cx1 = 146.7e-9;

Rx2 = Rx1;

Lx2 = Lx1;

Cx2 = Cx1;

f = frequency;

w = 2\*pi\*f;

% ZY2, Widerstand und Spule hablieren, Kondensator verdoppeln

Rx2 = 0.5 \* Rx2;

Lx2 = 0.5 \* Lx2;

Cx2 = 2.0 \* Cx2;

ZY2 = Rx2 + Lx2\*j\*w + 1/(Cx2\*j\*w);

Z = Lr\*w\*j + Rw;

% ZCX1, Rx1 und Lx1 halbieren, Cx1 verdoppeln

Cx1 = 2.0 \* Cx1;

Rx1 = 0.5 \* Rx1;

Lx1 = 0.5 \* Lx1;

ZCX1 = Rx1 + Lx1\*j\*w + 1/(Cx1\*j\*w);

% ZCY2 faellt weg

ZCY1 = Ry1 + Ly1\*w\*j + (1/(j\*w\*Cy1));

ZY1 = rpara(ZCY1,ZCX1); %Parallelschaltung der X und Y Kondensatoren

%A Matrixen aus Impedanzen berechnen:

%Querimpedanz A1:

A1 = [1 0; (1/ZY2) 1]; % Y2 in Schema

%Längsimpedanz A2 % Z in Schema

A2 = [1 Z; 0 1];

%Querimpedanz A3

A3 = [1 0; (1/ZY1) 1]; % Y1 in Schema

%Gesamtmatrix berechnen

A = A1 \* A2;

A = A \* A3;

%s21 berechnen

result = 2 / ( (A(1,1)\*sqrt(Rref2/Rref1)) + (A(1,2)/sqrt(Rref1\*Rref2)) + (A(2,1)\*sqrt(Rref2\*Rref1)) + (A(2,2)\*sqrt(Rref1/Rref2)));

end

**Funktion get\_s21\_CM.m**

function [result] = get\_s21\_CM(frequency)

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

% Projekt: s21 Parameter berechnen

% fuer die CM-Schaltung eines EMI-Filters

% Autor: Luca Krummenacher

% Datum: 30. April 2019

% Ort : FHNW Brugg-Windisch

% Version: 3.0

% Erneuerungen: Matlabcode stimmt nun mit Softwarecode überein.

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

% Gibt den s21 parameter fuer eine bestimmte Frequenz zurück

% Die Bauteilparameter sind in dieser Funktion bestimmt

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

% used functions: par.m

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

Rref1 = 50; % Rref1 = Eingangswiderstand

Rref2 = 50; % Rref2 = Lastwiderstand

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 + par(L0\*w\*j,par(0.5\*Rp,(1/(j\*w\*C1C2))));

%ZY aus vereinfachter Schaltung:

Ry1 = 246e-3;

Cy1 = 3.85e-9;

Ly1 = 8e-9;

%beide CY Kondensatoren gleich setzten

Ry2 = Ry1;

Cy2 = Cy1;

Ly2 = Ly1;

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

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

%A Matrizen aus Impedanzen berechnen:

%Laengsimpedanz A1:

A1 = [1 ZZtot; 0 1];

%Querimpedanz A2:

A2 = [1 0; (1/ZY1) 1];

%Querimpedanz A3:

A3 = [1 0; (1/ZY2) 1];

%Gesamtmatrix berechnen

A = A1 \* A2;

A = A \* A3;

%s21 berechnen

result = 2 / ( (A(1,1)\*sqrt(Rref2/Rref1)) + (A(1,2)/sqrt(Rref1\*Rref2)) + (A(2,1)\*sqrt(Rref2\*Rref1)) + (A(2,2)\*sqrt(Rref1/Rref2)));

end