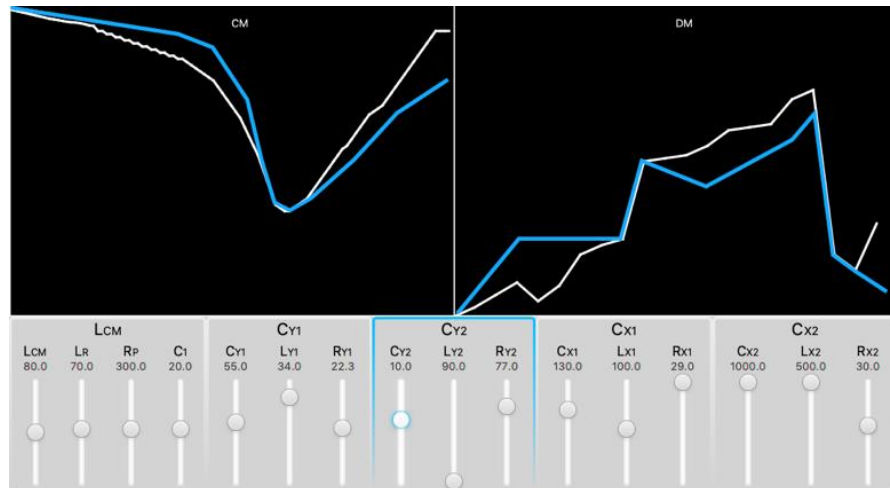


Pflichtenheft

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, 07.04.2019

Inhaltsverzeichnis

1 Übersicht.....	3
1.1 Ausgangslage	3
1.2 Projektziele.....	4
1.3 Wunschziele.....	4
1.4 Nicht-Ziele	5
1.5 Lieferobjekte.....	5
2 Softwarekonzept.....	6
2.1 Anforderungen	6
2.2 Software.....	6
2.3 Mock-Up.....	6
2.4 Menüleiste (noch nicht realisiert)	7
2.5 Anzeigefenster DM/CM	7
2.6 Haupt- und parasitäre Parameter	7
2.7 Beschreibung Programmablauf	7
3 Softwarestruktur	10
3.1 Klassendiagramm.....	10
3.2 Beschreibung	11
4 Theoretische Grundlagen	12
4.1 Einleitung EMI Filter	12
4.2 Vorgehensweise Berechnung Einfügungsverluste.....	12
5 Testkonzept.....	13
5.1 Kontrolle bei der Entwicklung	13
5.2 Überprüfung mit MATLAB	13
5.3 Integrationstest.....	13
5.4 Softwaretest	13
6 Literaturverzeichnis	15

1 Übersicht

1.1 Ausgangslage

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

Damit diese Störungen auf der Netzseite keine negativen Auswirkungen haben, beinhalten die Netzschaltteile einen Filter, welcher diese Störungen dämpft. Diese Netzfilter müssen die normativen Anforderungen erfüllen und dementsprechend dimensioniert werden. Diese Normen werden von den europäischen Normengremien festgelegt.

Das Ziel dieser Arbeit, ist die Entwicklung eines bedienungsfreundlichen 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 (Common- und Differentialmode) im ganzen Frequenzspektrum auswirken. Diese Informationen sollen dem Auftraggeber das Dimensionieren von Filtern vereinfachen.

1.2 Projektziele

Die Anforderungen in Tabelle 1 wurden als Pflichtziele definiert. Diese sind notwendig und müssen beim Endprodukt komplett erfüllt werden.

Nr.	Bereich	Beschreibung
1	Software	Die Software basiert auf der Programmiersprache Java, Version JDK11.
		Die Software läuft auf den aktuellen Betriebssystemen von MacOS (ab Version 10.11) und Windows (7 oder neuer) fehlerfrei.
		Die Software ist in der MVC-Architektur realisiert.
2	Bedienoberfläche	Die Bedienoberfläche soll für 80% der Nutzenden als bedienungsfreundlich und selbsterklärend empfunden werden.
		Die Bedienoberfläche wird mit JavaFX realisiert.
		Die Werte der parasitären Parameter sind einstellbar und zusätzlich um $\pm 30\%$ veränderbar via Schieberegler.
		Die Einfügungsverluste werden in Abhängigkeit der Frequenz [CM & DM] graphisch dargestellt.
3	Berechnungen	Die Berechnungen der Einfügungsverluste sind korrekt. Das heisst, sie stimmen mit den Simulationen vom Auftraggeber überein.
		Die Berechnungen der Einfügungsverluste dauern weniger als eine Sekunde.
		Die Schaltung wird mit den relevanten Bauteilen entsprechend der Aufgabenstellung simuliert.

Tabelle 1: Projektziele

1.3 Wunschziele

Je nach zeitlichem Verlauf des Projektes können (müssen aber nicht) Zusatzfunktionen implementiert werden. Voraussetzung ist, dass das Projekt keine Verzögerungen erfährt. In Tabelle 2 werden die Wunschziele definiert.

Nr.	Bereich	Beschreibung
1	Ausgabedateien	Die Ergebnisse (Graphen & Parameter) können als PDF- oder Textdatei gespeichert werden.
2	Eingabedateien	Gespeicherte Dateien können geladen werden (Save-Load-Option).
3	Programm	Bei Parameteränderungen mittels Slider wird eine zusätzliche Kurve der Verluständerung dargestellt.
		Die Sensibilitätsanalyse kann dargestellt werden.
		Es können beliebige Schaltungen erstellt und simuliert werden.

Tabelle 2: Wunschziele

1.4 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. Diese Nicht-Ziele werden in Tabelle 3 aufgelistet.

Nr.	Bereich	Beschreibung
1	Darstellung	Es wird eine 3D Darstellung realisiert.
2	Eingabefelder	Es wird eine Funktion für einen Bauelementvorschlag-rechner zur Erreichung einer Verlustkurve nach Wunsch realisiert.
3	Webapplikation	Es wird eine App (für Smart-Devices) erstellt.
4	Simulation	Es wird eine Funktion zur Berechnung der Permeabilität der Spulen in Abhängigkeit der Frequenz zur Dämpfung erstellt.

Tabelle 3: Nicht-Ziele

1.5 Lieferobjekte

Die Lieferobjekte sind in der Tabelle 4 festgelegt. Die Berichte werden per Mail an die Herren Dalessandro, Niklaus, Gaulocher und Buchschacher geschickt.

Datum:	Lieferobjekte
21.03.2019	Pflichtenheft Version 1
07.04.2019	Pflichtenheft Endversion
11.04.2019	Zwischenpräsentation
05.05.2019	Mock-Up Software
06.05.2019	Berechnungen mit Matlab
13.05.2019	Betaversion der Software
13.06.2019	Fachbericht und Präsentation

Tabelle 4: Lieferobjekte

2 Softwarekonzept

2.1 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 und Kosten.

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

2.2 Software

Die Software ist in der Programmiersprache Java realisiert, damit die Aufgabe objektorientiert gelöst werden kann. Dank der mitgelieferten Java Laufzeitumgebung (JRE), muss das Programm nur einmal geschrieben werden und läuft auf allen aktuellen Betriebssystemen wie zum Beispiel MacOS (ab Version 10.11) und Windows (ab Version 7). Ein weiterer Vorteil der Sprache Java ist, dass sie weit verbreitet ist und der Auftraggeber somit flexibel für die Vergabe von Wartungs-/ Änderungs- oder Verbesserungsarbeiten bleibt.

Das Programm wird in der MVC-Architektur geschrieben, so dass nachträgliche Änderungen und Erweiterungen ohne grossen Aufwand möglich sind und die Struktur nicht unnötig kompliziert wird. In dieser Architektur wird die Software in drei Bereiche unterteilt: Model, View und Controller. Das Model enthält die Berechnungen für Impedanzen, Kettenmatrizen und Einfügedämpfungen, die View die Elemente der Benutzeroberfläche und der Controller verknüpft die View mit dem Model. Die View ist dabei noch weiter unterteilt.

2.3 Mock-Up

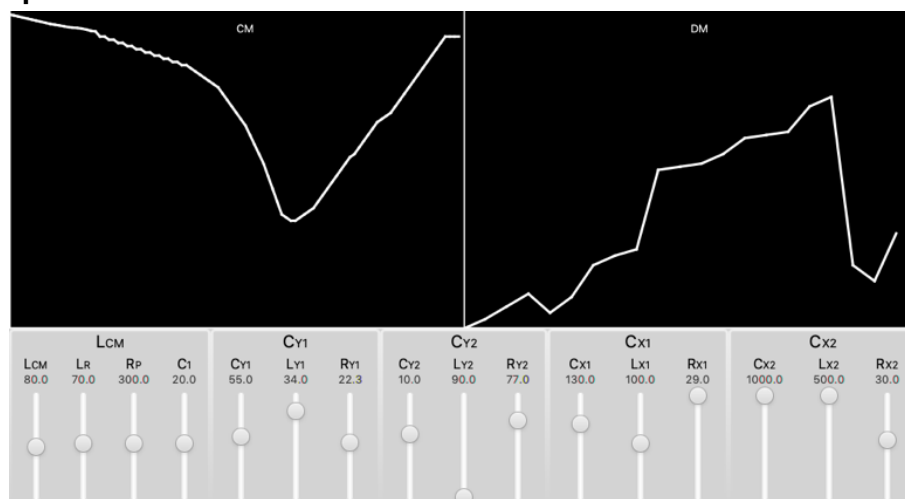


Abbildung 1: GUI Übersicht

Die grafische Bedienoberfläche, die in der Abbildung 1 zu sehen ist, besteht voraussichtlich aus einer Menüleiste (noch nicht realisiert), einem Anzeigefenster für jeweils DM und CM, sowie Haupt- und parasitären Parameter. Die Bibliothek «JavaFX» ermöglicht es 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. Damit die

Bedienoberfläche selbsterklärend und benutzerfreundlicher wird, wird die «Tool Tip API» verwendet: Sobald jemand mit dem Mauspfel länger als zwei Sekunden über einer Oberfläche oder Eingabefeld verweilt, wird eine Beschriftung mit der entsprechenden Funktion sichtbar.

2.4 Menüleiste (noch nicht realisiert)

Von hier aus haben Nutzende Zugriff auf verschiedene gröbere Einstellungen und Funktionen. Wunschziel-Funktionen wie Speichern und Laden von Daten, Ändern der Darstellung und andere könnten per Mausklick aufgerufen werden.

2.5 Anzeigefenster DM/CM

Die Einfügungsverluste erscheinen in zwei voneinander getrennten Diagrammen (CM und DM). Die Frequenzachse ist logarithmisch skaliert, damit ein Frequenzbereich von 0 bis 30MHz abgedeckt werden kann. Für die Darstellung der Kurven verwenden wir das Objekt Canvas. Somit haben wir volle Kontrolle über das «Look and Feel» der Diagramme und können besser auf diesbezügliche Wünsche des Auftraggebers eingehen.

Für eine genauere Ansicht und Inspektion einer Kurve kann das entsprechende Kurvendia-gramm per Doppelklick vergrößert werden. Das andere verschwindet dabei, erscheint aber nach erneutem Doppelklick auf das soeben vergrößerte Diagramm wieder an seiner Stelle.

2.6 Haupt- und parasitäre Parameter

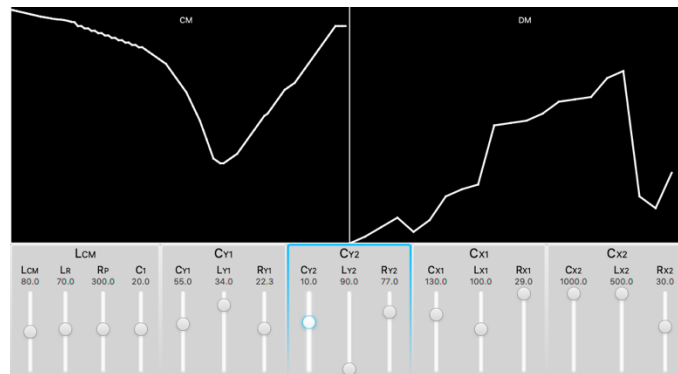


Abbildung 2: Parametereinstellungen

Jeder Hauptparameter wird durch eine gewisse Anzahl parasitärer Parameter dargestellt. Die Werte der letzteren können via Eingabefeld gesetzt und später noch via Schieberegler, die in der Abbildung 2 zu sehen sind, um maximal $\pm 30\%$ verändert werden. Jede Veränderung eines Schiebereglers löst eine erneute Berechnung im Model aus, worauf eine erneute Darstellung der Kurvendia-gramme folgt. Oberhalb der Regler werden die momentanen Werte der parasitären Parameter in einem Label aufgezeigt.

2.7 Beschreibung Programmablauf

Beim Starten des Programmes werden, sofern die Wunschziele erreicht werden können, die Daten einer zuvor beendeten Session wieder geladen. Nach erfolgreichem Laden erfolgt automatisch das Zeichnen der Kurve und die Schieberegler werden auf ihren entsprechen-

den Wert gesetzt. Nutzende können via Menüleiste gespeicherte Sessionen laden oder mit einer Neuen beginnen.

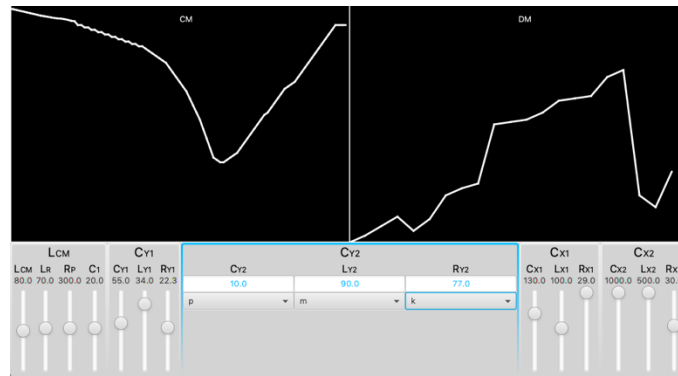


Abbildung 3: Anpassung der Parameter

Abbildung 3 zeigt, dass bei einem Doppelklick auf einen Parameter dessen Darstellung wechselt. Nun können die ursprünglichen Werte der parasitären Parameter via Textfelder und anderen Kontroll-Einheiten eingestellt werden. Bei erneutem Doppelklick geht die Ansicht wieder zurück zu den Schiebereglern.

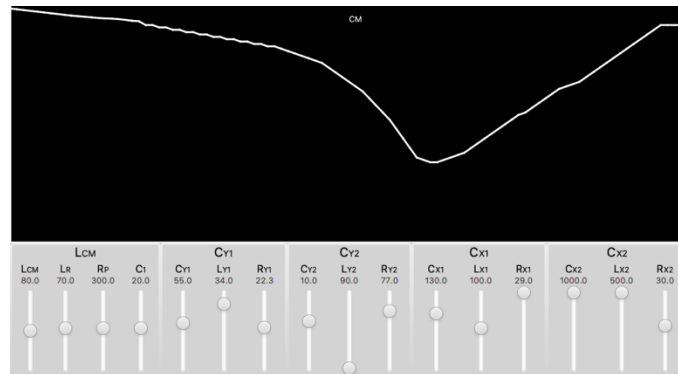


Abbildung 4: Kurvenansicht

Bei Doppelklick auf ein Diagramm, übernimmt dieses die ganze Breite des Fensters und kann so genauer angesehen werden, wie es in Abbildung 4 dargestellt wird. Bei erneutem Doppelklick wird der Vorgang rückgängig gemacht, so dass wieder beide Diagramme sichtbar sind.

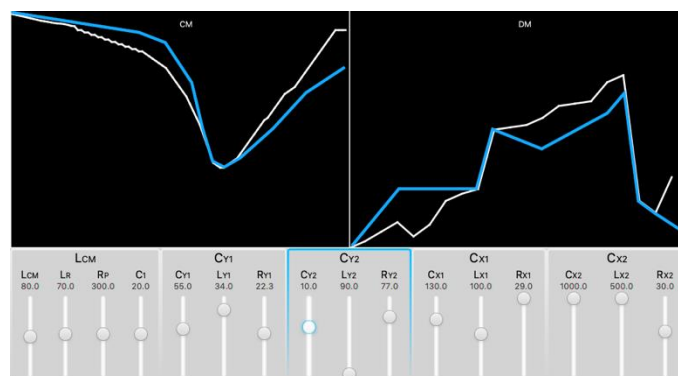


Abbildung 5: Parameterauswirkung

Wenn Nutzende den Mauszeiger über einen Schieberegler bewegen, soll laut Wunschziel 3 in den Diagrammen gezeigt werden, welche Auswirkungen dieser Schieberegler bei voller Auslenkung ($\pm 30\%$) auf die Kurve hätte. Diese Auswirkungen würden, wie in Abbildung 5 gezeigt, als andersfarbig gezeichnete Kurven realisiert werden. So könnten die unterschied-

lichen Auswirkungen aller parasitären Parameter schnell eingesehen werden, ohne dass die Nutzenden die Schieberegler einzeln verschieben müssen.

3 Softwarestruktur

3.1 Klassendiagramm

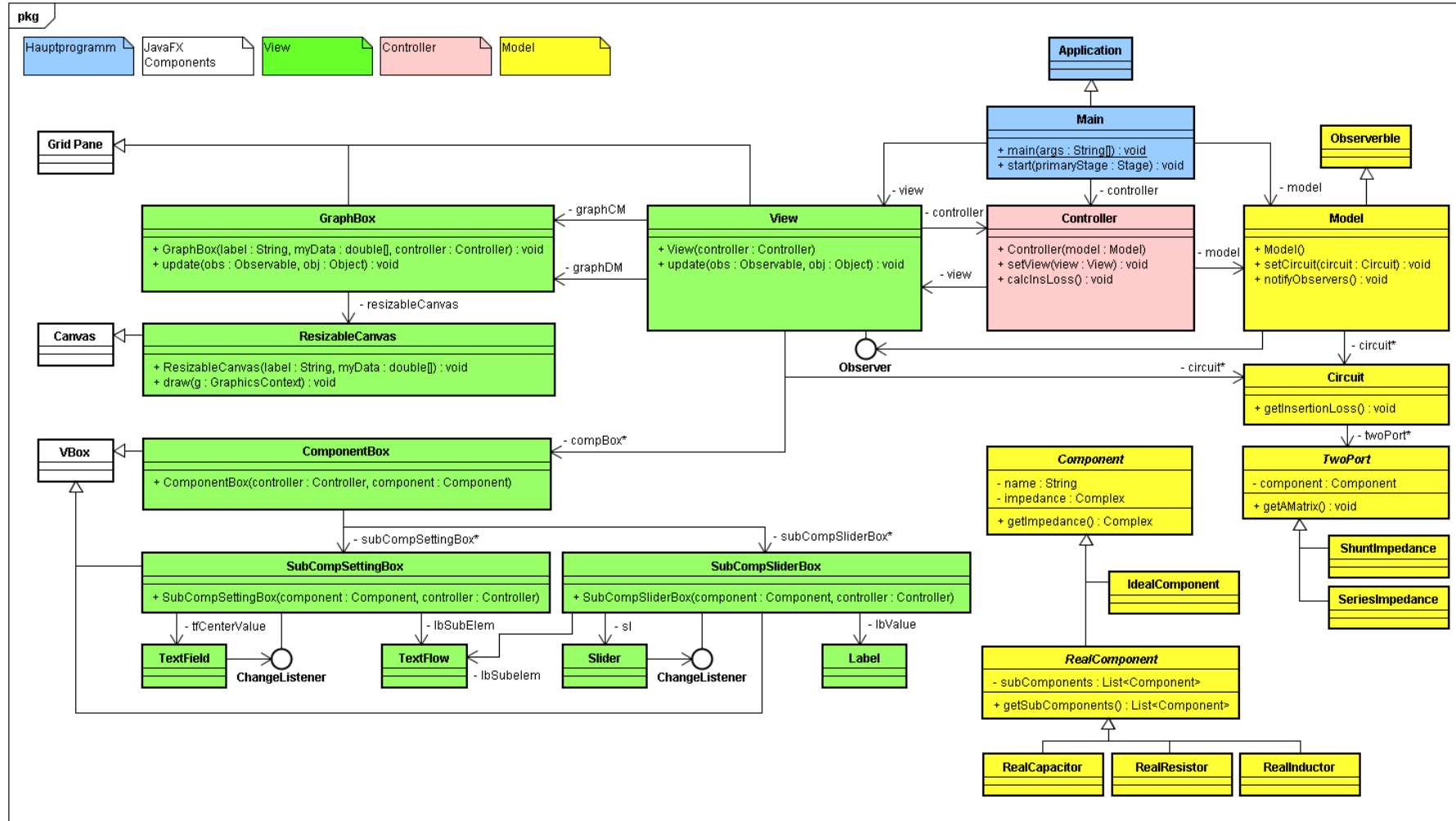


Abbildung 6: Klassendiagramm

3.2 Beschreibung

In

der

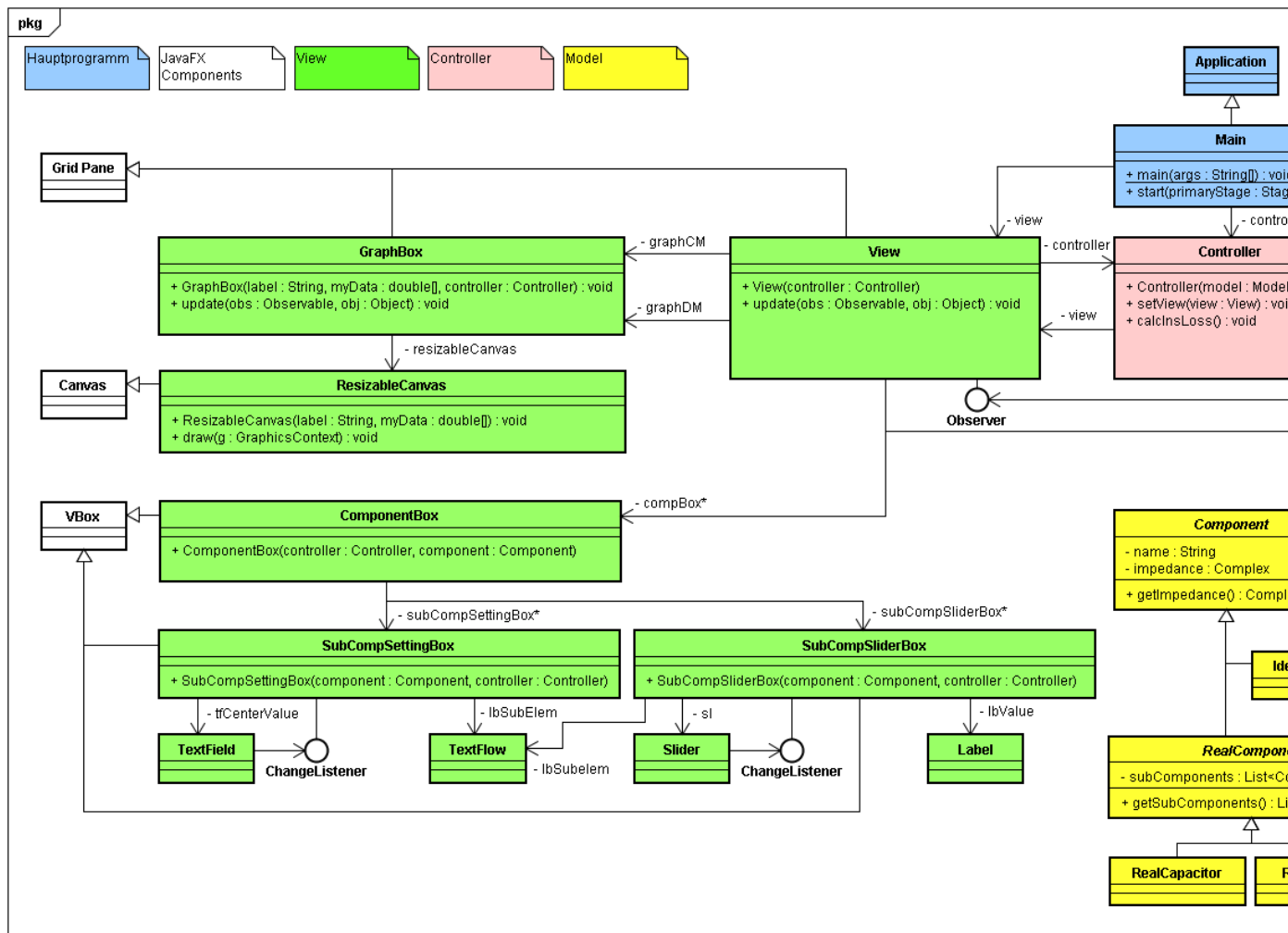


Abbildung 6 ist ersichtlich, dass die Aufgabe objektorientiert gelöst wird.

Von besonderem Interesse ist die abstrakte Klasse Component, mit der alle Bauteile des Filters modelliert werden können. Ob ideale- oder reale Bauteile, ob parallel oder in Serie geschaltet, von jedem Bauteil kann die Impedanz berechnet werden. Daraus wird die Matrix des Zweitors und dann die Einfügedämpfungen der entsprechenden Filterschaltung berechnet.

Sobald Nutzende mit der Bedienoberfläche interagieren, sorgt der ChangeListener dafür, dass die entsprechenden Funktionen im Controller ausgelöst werden. Bei einer Änderung eines Parameterwerts wird zum Beispiel via Controller im Model eine Funktion aufgerufen, welche die Einfügedämpfung neu berechnet. Nach der Berechnung erfolgt automatisch das Neuzeichnen der Kurven: Das Model bewirkt via notifyObservers(), dass die Funktion update() in der View aufgerufen wird und aktualisiert so die betroffenen Komponenten.

4 Theoretische Grundlagen

4.1 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. Damit die Schaltung des Geräts und keine anderen Geräte gestört werden. Damit dieses eingesetzt werden darf, muss jedes Schaltnetzteil sich an bestimmte Normen im Bereich EMV halten.

4.2 Vorgehensweise Berechnung Einfügevverluste

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 C_X -Kondensatoren weg, da diese bei dieser Schaltung trivial sind. Die vereinfachte Schaltung ist in der Abbildung 7 sichtbar.

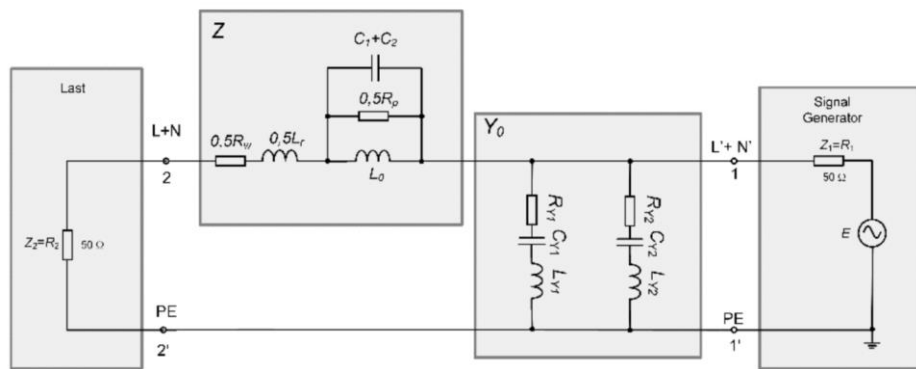


Abbildung 7: Vereinfachte CM-Schaltung

Mit der vereinfachten Schaltung bestimmen wir für alle Bauteile die Impedanzen für eine bestimmte Frequenz. Danach berechnen wir die Impedanz (Z), sowie die Admittanz (Y_0). Daraus ergeben sich Zweitoren. Demzufolge ergeben sich die Kettenmatrix ($A1$) für die Längsimpedanz (Z) sowie ($A2$) für die Queradmittanz (Y_0).

Für den S_{21} -Parameter benötigen wir die Gesamtmatrix, diese berechnet man mit Hilfe der Kettenschaltung, indem man die beiden Matrizen miteinander multipliziert.

Nun können wir die Einfügedämpfung a_I in dB berechnen: [1]

$$a_I = -20 \log |S_{21}| \text{ [dB]}$$

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

Für die DM-Schaltung geht man genau gleich vor. Diese Schaltung hat aber noch eine weitere Querimpedanz in der Schaltung. Die Gesamtmatrix für den S_{21} Parameter berechnet sich dann aus drei Zweitoren. Durch die objektorientierte Programmierung kann diese aber problemlos implementiert und falls nötig, jederzeit angepasst werden.

5 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.

5.1 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.

5.2 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.

5.3 Integrationstest

Die verschiedenen Klassen und Methoden werden isoliert geprüft. Somit 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.

5.4 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.

6 Literaturverzeichnis

- [1] A. Maier und S. Rupp, «Hochfrequenztechnik, Teil 2 - Anwendungen,» Duale Hochschule Baden-Württemberg , Stuttgart, 2016.