

«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

Inhaltsverzeichnis

| | | |
|----------|--|----------|
| 1 | Abstract | 1 |
| 2 | Einleitung | 2 |
| 3 | Software | 3 |
| 3.1 | Klassendiagramm | 3 |
| 3.2 | Ersatzschaltbilder | 3 |
| 3.3 | GUI | 4 |
| 3.3.1 | Menu | 4 |
| 3.4 | Datenverarbeitung | 4 |
| 3.5 | Datenpräsentation | 4 |
| 3.6 | Speicherverwaltung | 4 |
| 4 | Testkonzept | 5 |
| 4.1 | Prinzip | 5 |
| 4.2 | Erwartungen | 5 |
| 4.3 | Validierung | 5 |
| 5 | Schluss | 6 |
| 6 | Anhang | 7 |
| 6.1 | Elektrotechnik | 7 |
| 6.1.1 | Schaltungsaufbau | 7 |
| 6.1.2 | Parasitäre Paramter | 10 |
| 6.1.3 | Gleichtakt | 10 |
| 6.1.4 | Gegentakt | 10 |
| 6.1.5 | Insertion loss/ Streuparameter | 10 |
| 6.1.6 | Kettenmatrix | 12 |
| 6.2 | Programmieren | 14 |
| 6.2.1 | MVC-Struktur | 14 |
| 6.3 | Testkonzept | 14 |

1 Abstract

2 Einleitung

Unser Auftrag in diesem Projekt war es eine Software zu programmieren, die das Verhalten von so genannten EMI-Filtern simuliert. Diese werden üblicherweise in Schaltnetzteile verbaut, um zu verhindern, dass Störungen zurück ins Netz gespeist werden. Diese Netzgeräte können nämlich 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 sind. Der Auftraggeber ist Dr. Luca Dalessandro von der Firma Schaffner Unser Team besteht aus 5 Teammitgliedern. Der Projektleiter ist Niklaus Schwegler. Für die Software ist der Hauptverantwortliche Pascal Puschmann, dieser wurde je nach Bedarf vom gesamten Team unterstützt. Der Elektroteil wurde von Lukas von Däniken und Marco Binder realisiert. Die Organisatorischen Bereiche wurden von Niklaus Schwegler und Simon Rohrer abgedeckt.

Das Tool soll in der Lage sein den Filter in den Schaltungen Differential Mode und Common Mode zu berechnen. Ausserdem soll das User Interface sehr intuitiv und Bedienungsfreundlich sein, damit man schnell zum Resultat kommt. Des Weiteren kann man seine eingegeben Parameter Speichern und ein Filterprofil anlegen, was man später wieder aufrufen kann. Ein weiterer wichtiger Punkt ist, dass das Programm stabil laufen sollte, ohne dass es zu Ausfällen kommt. Dies wird unter anderem durch ausführliche Tests erreicht, was in diesem Bericht noch detailliert beschrieben wird. Um das Programm leicht erweiterbar zu halten haben wir uns für die MVC Struktur entschieden, dies sorgt für einen Modularen Aufbau des Codes. Zudem ist es Plattformunabhängig, weil es mit Java geschrieben wurde. Die Elektrotechnische Herausforderung bestand darin, das Verhalten der beiden gegebenen Schaltungen zu analysieren und durch Berechnungen zu validieren. Die Berechnungen von DM und CM mit allen realen Komponenten wurden in MATLAB durchgeführt. Auf die Resultate von unseren Berechnungen wird innerhalb dieses Berichts Genauer eingegangen. Dieser Bericht soll eine Dokumentation über unsere geleistete Arbeit und unser neu angeeignetes Wissen sein. Die Software steht ganz klar im Fokus des Geschehens, jedoch wird im Kapitel Grundlagen auch ausführlich auf die Elektrotechnik eingegangen, damit auch Fachfremde Leser die Chance haben etwas zu verstehen. Ein zentraler Abschnitt wird der Aufbau und die Funktionsweise der GUI sein. In dem unter anderem auf das Klassendiagramm eingegangen wird und das grafische Aussehen erklärt werden soll. Bis schlussendlich ein umfassendes Fazit gezogen werden kann.

3 Software

3.1 Klassendiagramm

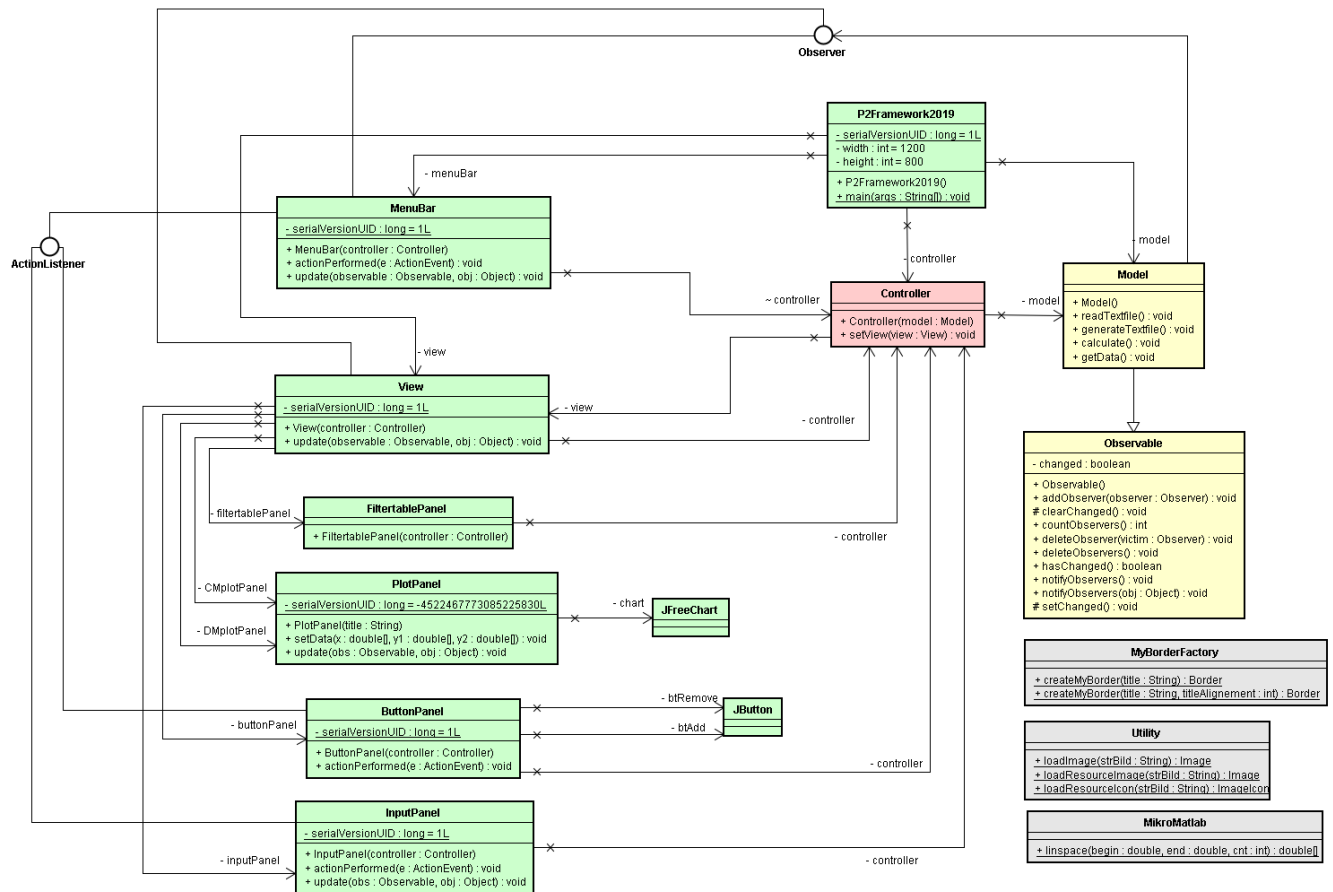


Abbildung 3.1: Klassendiagramm wtf

3.2 Ersatzschaltbilder

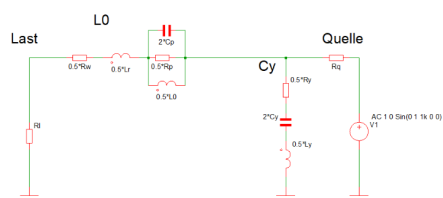


Abbildung 3.2: Vereinfachte CM_Schaltung

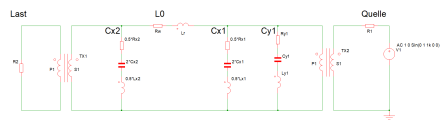


Abbildung 3.3: Vereinfachte DM-Schaltung

3.3 GUI

3.3.1 Menu

3.4 Datenverarbeitung

3.5 Datenpräsentation

3.6 Speicherverwaltung

Programmablauf beschreiben?

4 Testkonzept

Wieso, weshalb und Warum

4.1 Prinzip

Unser Testkonzept soll eine einwandfreie Funktion unserer Software garantieren und eine einfache Bedienung ermöglichen. Dies wird durch verschiedene Tests realisiert.

Grundsätzlich wird alles, was implementiert wird, noch einmal sorgfältig durchgelesen und im Fall der Software, wird noch mal kontrolliert, ob keine Fehlermeldungen im Code angezeigt werden. Zudem werden die Java Methoden einzeln ausgeführt, um sich über die Funktion zu vergewissern. Dies erfordert ein hohes Mass an Selbstverantwortung. Ausserdem wird so oft wie möglich neu eingefügter Code vom ganzen Team angeschaut, damit alle Teammitglieder die Funktionsweise der Software verstehen. Im Elektrotechnik-Teil ist es sehr wichtig, die ausgerechneten Werte zu hinterfragen und auf ihre Richtigkeit zu überprüfen. Dafür wird das Rechenprogramm MATLAB verwendet, weil es eines der umfangreichsten Mathematikprogramme auf dem Markt ist. Je nach Bedarf werden die Fachcoaches kontaktiert, falls noch Unklarheiten bestehen. Diese kontinuierliche Tests werden von Anfang an durchgeführt, um Folgefehler zu vermeiden, was uns mehr Zeit für Verbesserungen und neue Features gibt. Nach der Projektwoche ist die Version 0.9.5 fertig. Diese wird nach Testprotokoll gründlich getestet. Zunächst wird besprochen, ob man die sich gesteckten Ziele zufriedenstellend erreicht hat. Die Version wird auch dem Auftraggeber Dr. Luca Dalessandro zur Verfügung gestellt, damit er die Möglichkeit hat, seine Meinung und Ideen noch einmal einzubringen. Anschliessend beginnt die eigentliche Testphase. Die Software wird zum einen mit einem Kompatibilitätstest gefordert. Bei diesem Test werden Eingaben getätigt, die unsere Software an die Grenzen bringen dürfte. Es werden auch Fehleingaben gemacht, um zu sehen, wie die Software darauf reagiert. Ein anderer Aspekt, um die Kompatibilität zu prüfen, ist die Verwendung von verschiedenen Betriebssystemen (Mac OS, Windows). Um eine saubere Darstellung auf allen Displays zu gewährleisten, werden Bildschirme mit unterschiedlichen Auflösungen verwendet (Full HD, 4K). Danach werden die Tests durch dritte Personen durchgeführt. Dabei werden Experten und auch Fachfremde Tester gesucht. Die Fachfremden erhalten eine grundlegende Einführung über EMI-Filter, damit sie verstehen, wofür diese Software entwickelt wurde. Über die Funktionsweise der GUI an sich erhalten die Testpersonen jedoch keine Einführung. Damit kann man sehr gut prüfen, wie einfach und intuitiv die Softwarebedienung ist. Dafür wird das bereits erwähnte Testprotokoll verwendet, in das die Testpersonen ihre Meinung und allfällige Anregungen hineinschreiben können. Die Fragebogen werden ausgewertet und anschliessend wird sich im Team über allfällige Änderungen ausgetauscht. Vor der Abgabe wird die Software noch einmal getestet und ein Abnahmeprotokoll erstellt, bis die Software den Weg zum Auftraggeber findet.

4.2 Erwartungen

Was erwarten wir für Resultate?

4.3 Validierung

Testergebnisse darstellen und Interpretieren

Ebenfalls wird hier beschrieben, welche Werte wir mit der Simulation und mit Matlab erreicht haben.

5 Schluss

6 Anhang

6.1 Elektrotechnik

6.1.1 Schaltungsaufbau

Das vorgegebene EMI-Filter muss bezüglich der Einfügungsverluste (Insertion Loss) untersucht werden. Die Einfügungsverluste hängen vom Gesamtrauschen der Schaltung ab. Es wird ein Ansatz verwendet, der in der Praxis weit verbreitet ist, bei welchem das Gesamtrauschen in zwei Komponenten unterteilt wird. Man spricht vom Gegen-(=Differential Mode=DM) und Gleichtaktrauschen (=Common Mode=CM). Anhand der vorgegebenen CM- und DM-Äquivalenten Schaltungen (Abbildungen 6.4, 6.5) werden die Einfügungsverluste in Funktion der Frequenz berechnet. Die Berechnungen decken einen Bereich von 0 bis 30MHz ab.

Die Schaltung 6.1 Original Schaltung **aufgabenstellung** zeigt den Filteraufbau, wie er der Aufgabenstellung zu entnehmen ist. Um das Gegentaktrauschen und das Gleichtaktrauschen bestimmen zu können, werden die beiden Schaltungsäquivalente gebildet.

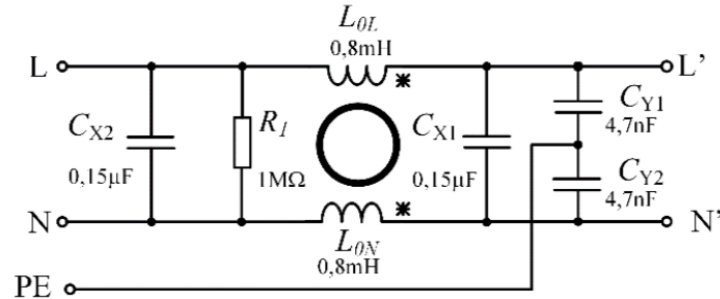


Abbildung 6.1: Original Schaltung **aufgabenstellung**

Hierbei müssen die elektrischen Bauelemente, wie Spule und Kondensator mit den passenden parasitären Parameter ergänzt werden. In Abbildung ?? und ?? werden die parasitären Parameter von Spule und Kondensator gezeigt.

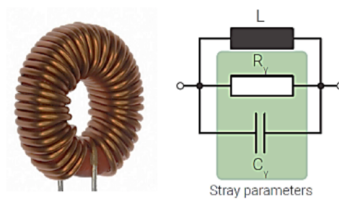


Abbildung 6.2: Parasiäre Elemente einer Induktivität **aufgabenstellung**

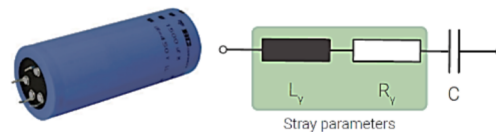


Abbildung 6.3: Parasiäre Elemente einer Kapazität **aufgabenstellung**

Folgende Schaltungen stellen die CM- und DM-Äquivalenten Schaltungen. Da die Berechnungen in einem Bereich von bis zu 30 MHz gemacht werden, ist es notwendig die parasitären Parameter von Spule und Kondensator miteinzubeziehen.

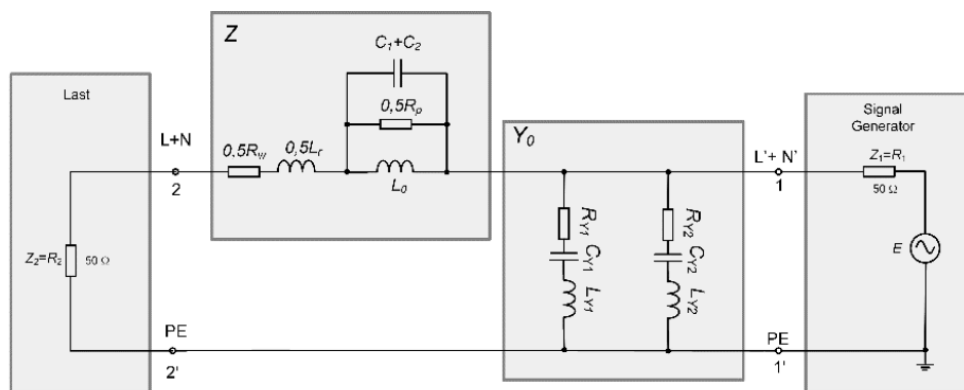
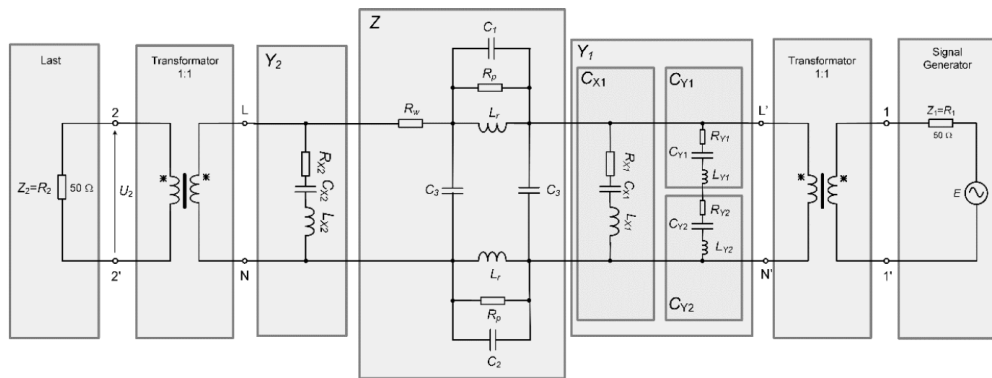


Abbildung 6.4: CM-Schaltungsäquivalent **aufgabenstellung**

Abbildung 6.5: DM-Schaltungsäquivalent **aufgabenstellung**

6.1.2 Parasitäre Paramter

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

6.1.3 Gleichtakt

Beschreibung Common-Mode und wie man ihn ausrechnet

6.1.4 Gegentakt

Beschreibung Differential-Mode und wie man ihn ausrechnet

6.1.5 Insertion loss/ Streuparameter

Folgende Theorieabschnitte wurden überwiegend anhand folgender Quellen zusammengestellt: **hftech**. Die Einfügungsverluste werden analytisch ermittelt. Im ersten Schritt werden die Berechnungen in MATLAB gemacht. Somit können die Funktionen geplottet werden. Diese Plots werden dann mit Simulationen in MPLAB Mindi verglichen um festzustellen ob diese korrekt sind. Die vollständigen und korrekten Berechnungen können somit in Java implementiert werden. Um die Einfügungsverluste bestimmen zu können, wird das Model der 2-Tore verwendet. Einzelne Schaltungsteile werden in ABCD-Matrixen ?? abgebildet, welche dann durch Kaskadierung der einzelnen ABCD-Matrixen zusammengeführt werden. Die Einfügungsverluste werden aus den Streuparameter?? abgeleitet, welche direkt aus der ABCD-Matrix berechnet werden. Der 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.

$$IL = |H(j\omega)| = 20 * \log\left(\frac{|U_{20}|}{|U_2|}\right) \quad (6.1)$$

In der Definition kann das Spannungsverhältnis durch den Streuparameter 6.1.5 (S-Parameter) S_{21} ersetzt werden 6.2.

$$IL = -20 * \log(|S_{21}|) \quad (6.2)$$

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 6.6 2-Tor Wellengrößen und Anschlussleitungen **hftech** zeigt die Streuparameter an einem 2-Tor.

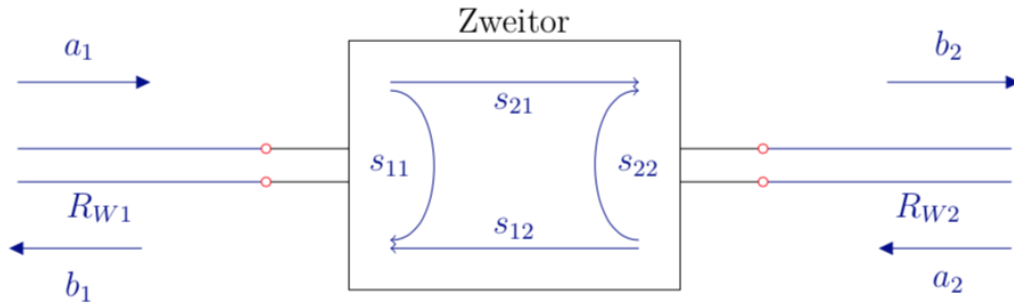


Abbildung 6.6: 2-Tor Wellengrößen und Anschlussleitungen **hftech**

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 6.3 und 6.4 zeigen wie die Wellengrößen a_i sowie b_i definiert sind.

$$a_i = \frac{U_i + R_{Wi} I_i}{2 * \sqrt{R_{Wi}}} \quad (6.3)$$

$$b_i = \frac{U_i - R_{Wi} I_i}{2 * \sqrt{R_{Wi}}} \quad (6.4)$$

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 6.5):

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} * \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \quad (6.5)$$

Die Elemente der S-Matrix sind:

$$s_{11} = b_1/a_1 \text{ Eingangsreflexionsfaktor bei angepasstem Ausgang (} a_2=0 \text{)} \quad (6.6)$$

$$s_{12} = b_1/a_2 \text{ Rückwärtstransmissionsfaktor bei angepasstem Eingang (} a_1=0 \text{)} \quad (6.7)$$

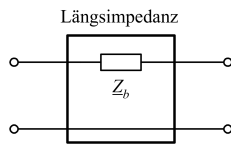
$$s_{21} = b_2/a_1 \text{ Vorwärtstransmissionsfaktor bei angepasstem Ausgang (} a_2=0 \text{)} \quad (6.8)$$

$$s_{22} = b_2/a_2 \text{ Ausgangsreflexionsfaktor bei angepasstem Eingang (} a_1=0 \text{)} \quad (6.9)$$

6.1.6 Kettenmatrix

Die ABCD-Matrix ist eine weitere gängige Variante, um das Verhalten von 2-Toren zu beschreiben. Diese Variante hat den Vorteil, dass man in Serie geschaltene 2-Tore ohne grossen Aufwand zusammen rechnen kann. Sobald man die einzelnen ABCD-Matrixen gebildet hat und die Schaltung soweit vereinfacht ist, dass nur noch in Serie geschaltene ABCD-Matrixen vorzufinden sind, können diese miteinander multipliziert werden. Das Matrix-Produkt stellt dann die ABCD-Matrix der Gesamtschaltung dar. Folgende gängigen Schaltungen helfen die ABCD-Matrixen der einzelnen Schaltungsteilen zu bilden.

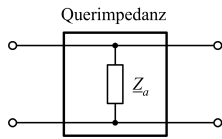
Die Längsimpedanz lässt sich anhand der ABCD-Matrix A_L (Formel 6.10) darstellen



$$A_L = \begin{bmatrix} 1 & Z_b \\ 0 & 1 \end{bmatrix} \quad (6.10)$$

Abbildung 6.7: Längsimpedanz
2torTabelle

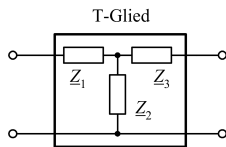
Die Querimpedanz lässt sich anhand der ABCD-Matrix A_Q (Formel 6.11) darstellen



$$A_Q = \begin{bmatrix} 1 & 0 \\ \frac{1}{Z_a} & 1 \end{bmatrix} \quad (6.11)$$

Abbildung 6.8: Querimpedanz
2torTabelle

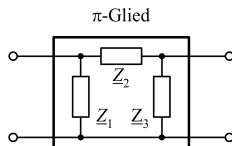
Die Impedanz eines T-Glieds lässt sich anhand der ABCD-Matrix A_T (Formel 6.12) darstellen



$$A_T = \begin{bmatrix} 1 + \frac{Z_2}{Z_3} & Z_1 + Z_3 + \frac{Z_1 Z_3}{Z_2} \\ \frac{1}{Z_2} & 1 + \frac{Z_3}{Z_2} \end{bmatrix} \quad (6.12)$$

Abbildung 6.9: T-Glied **2torTabelle**

Die Impedanz eines π -Glieds lässt sich anhand der ABCD-Matrix A_π (Formel 6.13) darstellen



$$A_\pi = \begin{bmatrix} 1 + \frac{Z_2}{Z_3} & Z_2 \\ \frac{1}{Z_1} + \frac{1}{Z_3} + \frac{Z_2}{Z_1 Z_3} & 1 + \frac{Z_2}{Z_1} \end{bmatrix} \quad (6.13)$$

Abbildung 6.10: Pi-Glied **2torTabelle**

Wenn die ABCD-Matrix einer Schaltung gebildet wurde, kann diese direkt in die Streuparameter umgewandelt werden. Der s_{21} Parameter kann wie in Formel 6.14 beschrieben, durch

einsetzen der ABCD-Matrix bestimmt werden. Für den Widerstand R_w muss die verwendete Bezugsimpedanz eingesetzt werden.

$$s_{21} = \frac{2}{A_{11} + \frac{A_{12}}{R_w} + A_{21} * R_w + A_{22}} \quad (6.14)$$

Die Indexierung der ABCD-Matrix wird in Abbildung 6.15 gezeigt

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad (6.15)$$

6.2 Programmieren

6.2.1 MVC-Struktur

Hier kommen noch weitere Programmiertechnische Grundlagen, falls diese Nötig sind

6.3 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 | | | |