



HSB

Hochschule Bremen
City University of Applied Sciences

HOCHSCHULE BREMEN
FAKULTÄT 4: ELEKTROTECHNIK UND INFORMATIK

Analoge Schaltungen

Biquadratische IIR (SOS) Filter

Daniel Albinger (5183249)

Phillip Dorsch (5186000)

Nils Renner (5197659)

Ansprechpersonen

Prof. Dr. Mirco Meiners

Abgabe: 21.07.2025

Inhaltsverzeichnis

1	Einleitung	1
1.1	Zielsetzung	1
2	Theorie	2
2.1	Mathematische Herleitung der Übertragungsfunktionen	3
2.2	Herleitung der Übertragungsfunktion über die systemtheoretische Analyse .	8
2.3	Einfluss der Bauteilgrößen C und R auf das Filterverhalten	9
2.4	Einfluss von Güte- und Verstärkungsfaktor auf das Filterverhalten	10
2.4.1	Einfluss von Q und H_0 auf das Frequenzverhalten	11
2.5	Herleitung einer Blockstruktur für ein analoges Biquad-Filter	13
2.6	Vergleich von analogen und digitalen Filter	14
3	Design eines digitalen Biquads	16
3.1	Bilineare Transformation	16
3.2	Prewarping	17
3.3	Filter Strukturen	17
3.3.1	Direktform 1 und 2	17
3.3.2	Kaskadierte Form und SOS-Strukturen	19
3.4	Implementieren eines digitalen Biquad in Python	20
4	Durchführung und Aufnahme der Messwerte	25
4.1	Softwaretools	25
4.2	Simulation	25
4.3	Messmittel	26
4.4	Messaufbau und verwendete Einstellungen	27
4.5	ASLK-Board	29
4.6	PCB	29
4.7	Aufnahme der Messwerte über die Weboberfläche	32
4.8	Automatisierung der Messwertaufnahme und Auswertung	33
4.9	Probleme bei der Aufnahme und Auswertung der Messwerte	40
5	Auswertung der Ergebnisse	41
5.1	Tiefpass	41
5.2	Hochpass	44
5.3	Bandpass	46
5.4	Bandsperre	51
6	Zusammenfassung	56
7	Fazit	58
8	Anhang	59

1 Einleitung

In einer zunehmend digitalisierten Welt liegt der Fokus vieler technologischer Entwicklungen auf digitalen Schaltungen, Signalverarbeitung und softwaredefinierten Systemen. Dennoch ist in nahezu jeder Signalverarbeitungskette auch analoge Schaltungstechnik implementiert. Das liegt unter anderem daran, dass Übertragungen, wie zum Beispiel bei einer Funkstrecke, durch physikalische Signale aus analogen, kontinuierlichen Größen realisiert werden. Dabei werden die Elektromagnetische Wellen als analoge Spannungen oder Ströme von der Antenne empfangen. Die Verarbeitung durch digitale Systeme erfolgt erst im späteren Verlauf.

Bevor ein Signal durch einen Analog-Digital-Wandler (ADC) in den digitalen Bereich überführt werden kann, ist eine analoge Vorverarbeitung notwendig. Diese soll den Signal-zu-Rausch-Abstand (SNR) optimieren und eine hohe Informationsqualität sicherstellen. Ein verrauschtes oder verzerrtes Signal kann im ADC zu Informationsverlusten und damit zu einer deutlich eingeschränkten Übertragungsqualität führen. Im Rahmen der analogen Vorverarbeitung sorgen Verstärker für eine Anpassung des Signalpegels, damit das empfangene Signal durch den ADC korrekt erkannt und Verarbeitet werden kann. Zusätzlich sind Filterbausteine für eine Abtrennung des Nutzsignals vom umgebenden Rauschen und störenden Nebenfrequenzen verantwortlich.

Insbesondere die analoge Filterung stellt einen zentralen Schritt dar. Nur durch die gezielte Unterdrückung unerwünschter Frequenzanteile lässt sich eine zuverlässige digitale Weiterverarbeitung sicherstellen und beispielsweise das Nyquist-Kriterium erfüllen. Als praktisches Beispiel soll in diesem Modul der sogenannte biquadratische Filter, kurz Biquad, genauer untersucht werden.

Allgemein ist der Biquad-Filter ein analoger oder digitaler Filter zweiter Ordnung, der sich durch seine kompakte Realisierbarkeit und vielseitige Einsatzmöglichkeiten auszeichnet. Er lässt sich sehr einfach durch 3 bzw. 4 miteinander verschaltete Operationsverstärker (OPs) realisieren. Die vielseitigen Einsatzmöglichkeiten ergeben sich dadurch, dass an den Ausgängen der einzelnen OPs unterschiedliche Filtertypen bereit gestellt werden können. Für Filter höherer Ordnung werden mehrere Biquad-Stufen in Reihe kaskadiert.

1.1 Zielsetzung

Ziel dieses Moduls ist es, den Aufbau, das Verhalten und die Einsatzmöglichkeiten eines analogen Biquads zu untersuchen. Dafür soll zunächst die theoretische Grundlage erarbeitet und die Übertragungsfunktionen der verschiedenen Filtertypen hergeleitet werden. Zudem soll die Bedeutung von Gütefaktor Q und Verstärkungsfaktor H_0 erarbeitet werden.

Darauf aufbauend folgt die erste Realisierung des Biquads auf dem ASLK-PRO Bord um Verständnis für die gewählten Parameter der Schaltung, wie beispielsweise Größe der Widerstände und Kondensatoren zu gewinnen. Anschließend soll aus den Erkenntnissen auf dem ASLK-Bord eine eigene Leiterplatte (PCB) entwickelt werden.

2 Theorie

Die bisher in diesem Studium behandelten Filterschaltungen wurden meist durch Kapazitäten und/oder Induktivitäten realisiert. Das hatte den großen Vorteil, dass die Schaltungen dadurch sehr verständlich und einfach zu identifizieren waren. Wenn eine Schaltung jedoch auf physikalischen Platz, den sie einnimmt, optimiert werden soll, führt das zu Problemen. Während Kondensatoren sehr kompakt aufgebaut und problemlos in integrierte Schaltungen integriert werden können, stellen Induktivitäten eine Herausforderung dar. Sie sind groß, schwer und lassen sich nur schwer miniaturisieren. Daher ist ihr Einsatz in modernen elektronischen Systemen, insbesondere bei hoher Integration, unpraktisch.

Um dennoch die gewünschten Filterfunktionen ohne echte Spulen zu realisieren, müssen Induktivitäten durch andere Komponenten simuliert werden. Eine Möglichkeit dafür ist der Gyrator, der mithilfe aktiver Bauelemente das Verhalten einer Induktivität nachbildet. Dieser benötigt jedoch eine Bezugserdung und ist deshalb nur eingeschränkt als flexibles Filterelement einsetzbar. Eine Alternative stellt der Floating Gyrator dar, der ohne festen Massebezug auskommt, jedoch deutlich komplexer und instabiler ist.

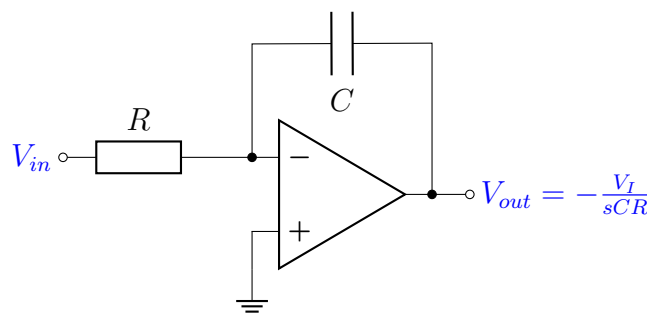


Abbildung 1: Operationsverstärker als Invertierender Integrator

Eine bessere Lösung bietet der Operationsverstärker (OpAmp), der beispielsweise als Integrator betrieben werden kann. Durch die geschickte Kombination von OpAmps mit Widerständen und Kondensatoren lassen sich vielfältige Filterstrukturen auf kleinem Raum realisieren. Diese Eigenschaft macht Operationsverstärker zu einem zentralen Baustein in der modernen analogen Filtertechnik.

Ein besonders flexibler Filter ist der sogenannte Biquad, ein aktiver Filter zweiter Ordnung. In dieser Arbeit kommt der Multiple-Feedback-Biquad (MFB) zum Einsatz.

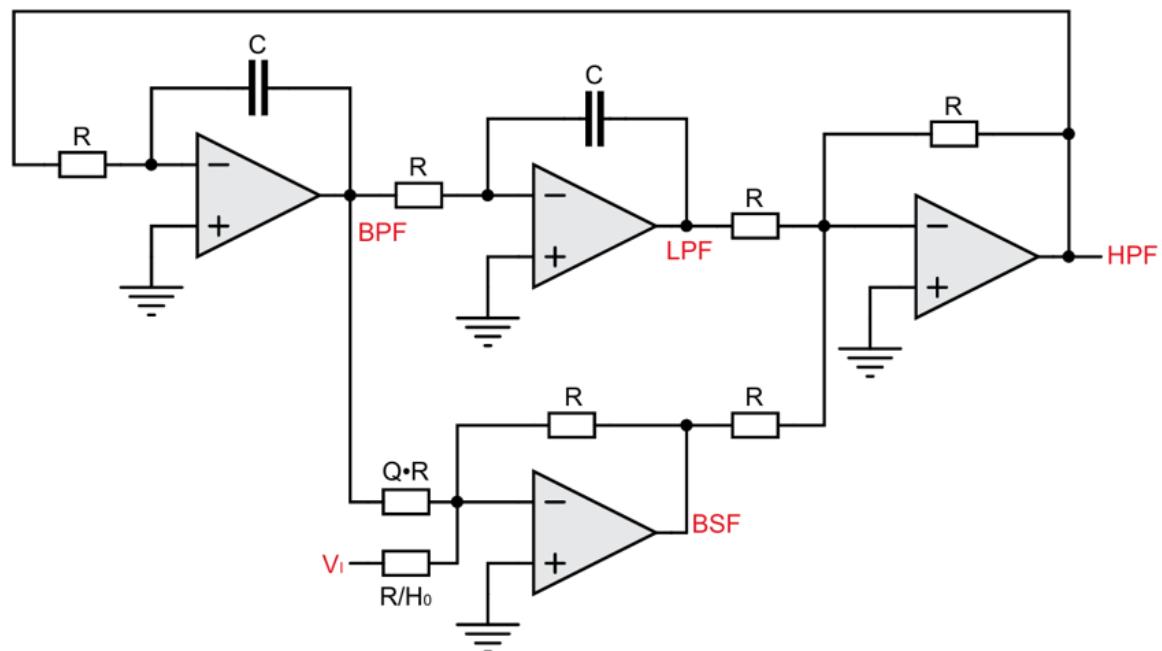


Abbildung 2: Multiple-Feedback-Biquad [1]

Diese Schaltung basiert auf addierenden und invertierenden Operationsverstärkern, bei der sowohl Rückkopplungen als auch Vorschaltpfade mit mehreren Bauelementen realisiert werden. Der MFB-Biquad zeichnet sich durch eine hohe Stabilität und einfache Abstimmung der Filterparameter aus und eignet sich besonders gut für die Realisierung von allen Arten von Filtertypen.

Neben dem MFB-Biquad existieren weitere Topologien, etwa der Tow-Thomas-Biquad oder der KHN-Biquad (Kerwin-Huelsman-Newcomb). Diese Varianten unterscheiden sich in ihrer inneren Struktur, der Art der bereitgestellten Ausgangssignale sowie ihrer Empfindlichkeit gegenüber Bauteiltoleranzen. So gibt der Tow-Thomas-Biquad nur eine Tiefpass- und Bandpassfilterung an seinen Outputs aus. Auch der KHN-Biquad liefert an seinen Ausgängen nur Hochpass-, Tiefpass- und Bandpass-Signale. Die Wahl der geeigneten Topologie hängt daher stark vom jeweiligen Anwendungsfall ab. Für dieses Modul ist der MFB-Biquad die am besten geeignete Wahl, da dieser alle Filtertypen bereitstellt und sich Mittenfrequenz, Gütefaktor und Verstärkung weitgehend unabhängig voneinander einstellen lassen.[2]

2.1 Mathematische Herleitung der Übertragungsfunktionen

Um die Übertragungsfunktionen der einzelnen Filterausgänge zu bestimmen, mussten zunächst die Formeln der in Abbildung 2 dargestellten Operationsverstärkerstufen hergeleitet werden.

Dazu wurde das idealisierte Operationsverstärker-Modell verwendet. Es wird angenommen, dass kein Strom in die Eingänge des Operationsverstärkers fließt und die Spannungsdifferenz zwischen dem invertierenden und dem nichtinvertierenden Eingang bei bestehen-

der negativer Rückkopplung stets null ist. Diese sogenannte virtuelle Masse vereinfacht die Analyse erheblich.

Die Herleitung der Ausgangsgleichungen erfolgte für jede einzelne OPV-Stufe durch Anwendung des Knotenstromsatzes am invertierenden Eingang. Die Ströme, die über die Eingangswiderstände zugeführt werden, müssen über den Rückkopplungszweig wieder abgeführt werden, da in den OPV selbst kein Strom fließt. Daraus ergibt sich eine Gleichung, die die Ausgangsspannung in Abhängigkeit der Eingangsspannungen beschreibt.

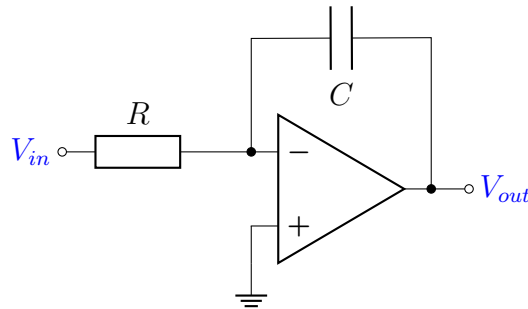


Abbildung 3: Invertierender Integrator

Im Fall eines invertierenden Integrators (Abbildung 3) wird die Eingangsspannung über einen Widerstand R zugeführt, während die Rückkopplung über einen Kondensator C erfolgt. Der Strom durch den Widerstand beträgt $I = V_{in}/R$ und muss vollständig durch den Kondensator abfließen. Über die Strom-Spannungs-Beziehung des Kondensators $I = C \cdot \frac{dV_{out}}{dt}$ ergibt sich die bekannte Ausgangsgleichung:

$$V_{out}(t) = -\frac{1}{RC} \int V_{in}(t) dt \quad (1)$$

Im Laplace-Bereich ergibt sich entsprechend:

$$V_{out}(s) = -\frac{V_{in}(s)}{sRC} \quad (2)$$

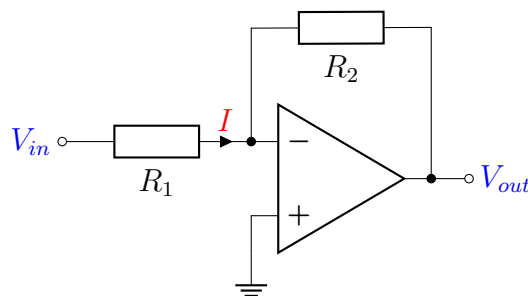


Abbildung 4: invertierender Verstärker

Bei der nächsten Stufe handelt es sich um einen invertierenden Verstärker mit zwei Widerständen R_1 (Eingang) und R_2 (Rückkopplung), wie in Abbildung 4 dargestellt. Der gesamte Strom $I = V_{in}/R_1$ fließt durch den Rückkopplungswiderstand R_2 , sodass sich diese Ausgangsspannung ergibt. [3]

$$V_{out} = -\frac{R_2}{R_1} \cdot V_{in} \quad (3)$$

Darauf aufbauend folgt eine **Addierschaltung** mit mehreren Eingangssignalen (siehe Abbildung 5). Jeder Eingang besitzt einen eigenen Widerstand, über den ein Strom zum virtuellen Massepunkt fließt. Diese Ströme summieren sich und fließen gemeinsam durch den Rückkopplungswiderstand R_2 . [3] Die Ausgangsspannung ergibt sich somit zu:

$$V_{out} = -R_2 \left(\frac{V_1}{R_{11}} + \frac{V_2}{R_{12}} + \frac{V_3}{R_{13}} \right) \quad (4)$$

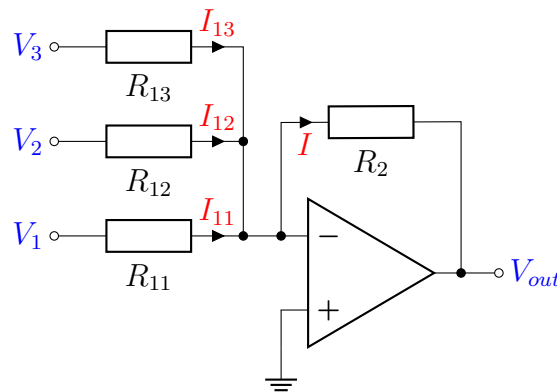


Abbildung 5: invertierender Addierer

Durch die Kombination der zuvor beschriebenen Teilschaltungen ergibt sich die vollständige Biquad-Filterschaltung. Die resultierende Schaltung besteht aus mehreren gekoppelten Verstärker-Stufen, deren Ausgangsspannungen jeweils die Eingangsgrößen der nachfolgenden Stufen beeinflussen. Im Folgenden sind die Gleichungen aufgelistet, die die Spannungsverläufe an den relevanten Knotenpunkten der realisierten MFB-Biquad-Struktur beschreiben:

$$V_1 = -(V_3 + V_4) \quad (5)$$

$$V_2 = -\left(\frac{1}{s}\omega_0 \cdot V_1\right) \quad (6)$$

$$V_3 = -\left(\frac{1}{s}\omega_0 \cdot V_2\right) \quad (7)$$

$$V_4 = -\left(\frac{V_2}{Q} + H_0 \cdot V_i\right) \quad (8)$$

Im Folgenden werden nun die Übertragungsfunktionen der einzelnen Filtertypen berechnet.

Berechnung von $\frac{V_2}{V_i}$:

Durch Einsetzen von Gleichung (5) in (6) ergibt sich:

$$\begin{aligned} V_2 &= -\frac{\omega_0}{s} \cdot V_1 = -\frac{\omega_0}{s} \cdot (-V_3 - V_4) \\ &= -\frac{\omega_0}{s} \cdot \left(\frac{\omega_0}{s} \cdot V_2 + \frac{1}{Q} \cdot V_2 + H_0 \cdot V_i \right) \end{aligned}$$

Durch Umformung und Zusammenfassen der Terme ergibt sich:

$$V_2 = -\frac{\omega_0^2}{s^2} \cdot V_2 - \frac{\omega_0}{sQ} \cdot V_2 - \frac{\omega_0}{s} H_0 \cdot V_i$$

$$\Rightarrow \frac{\omega_0}{s} H_0 \cdot V_i = -\left(\frac{\omega_0^2}{s^2} + \frac{\omega_0}{sQ} + 1\right) V_2$$

Nach Umstellung der Gleichung ergibt sich die Übertragungsfunktion $\frac{V_2}{V_i}$:

$$\frac{V_2}{V_i} = \frac{\frac{\omega_0}{s} H_0}{-\left(\frac{\omega_0^2}{s^2} + \frac{\omega_0}{sQ} + 1\right)} = -\frac{\frac{s}{\omega_0} H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} \quad (9)$$

Diese Übertragungsfunktion ergibt sich für die Bandpassfilterung.

Übertragungsfunktion $\frac{V_3}{V_i}$:

Durch Einsetzen der zuvor hergeleiteten Übertragungsfunktion (9) in Gleichung (7) ergibt sich:

$$V_3 = -\frac{\omega_0}{s} \cdot V_2 = -\frac{\omega_0}{s} \cdot \left(-\frac{\frac{s}{\omega_0} H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} \cdot V_i\right)$$

$$= \left(\frac{H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}}\right) V_i$$

$$\Rightarrow \frac{V_3}{V_i} = \frac{H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} \quad (10)$$

Die Übertragungsfunktion beschreibt die Tiefpasscharakteristik des Filters. Bei tiefen Frequenzen nähert sich der Betrag der Verstärkung dem Wert H_0 .

Übertragungsfunktion $\frac{V_4}{V_i}$:

Durch Einsetzen der zuvor bestimmten Funktion $V_2(V_i)$ aus Gleichung (9) in die Definition von V_4 gemäß Gleichung (8) ergibt sich:

$$V_4 = -\left(\frac{1}{Q} \cdot V_2 + H_0 \cdot V_i\right) = -\left(\frac{1}{Q} \cdot \left(-\frac{\frac{s}{\omega_0} H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} \cdot V_i\right) + H_0 \cdot V_i\right)$$

$$= \left(\frac{\frac{s}{\omega_0 Q} H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} - H_0\right) V_i$$

$$= \left(\frac{\frac{s}{\omega_0 Q} - \left(1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}\right)}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} \cdot H_0\right) V_i$$

$$= -\frac{\left(1 + \frac{s^2}{\omega_0^2}\right) H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} \cdot V_i$$

$$\Rightarrow \frac{V_4}{V_i} = -\frac{\left(1 + \frac{s^2}{\omega_0^2}\right) H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} \quad (11)$$

Hinweis: Im Aufgabenheft wird hier

$$\frac{V_4}{V_i} = \frac{\left(1 + \frac{s^2}{\omega_0^2}\right) H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}}$$

angegeben, was einer Negierung meines Ergebnisses entspricht.

Berechnung der Übertragungsfunktion $\frac{V_1}{V_i}$:

Ausgehend von Gleichung (5):

$$V_1 = -(V_3 + V_4)$$

Zunächst wird die zuvor bestimmte Übertragungsfunktion (10) für V_3 verwendet:

$$\frac{V_3}{V_i} = \frac{H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}}$$

Fall 1: Verwendung der negierten Form der Übertragungsfunktion von V_4 aus der eigenen Herleitung:

$$\frac{V_4}{V_i} = - \frac{\left(1 + \frac{s^2}{\omega_0^2}\right) \cdot H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} \quad (12)$$

Durch Einsetzen von (10) und (12) in Gleichung (5) ergibt sich:

$$\begin{aligned} V_1 &= -(V_3 + V_4) = - \left(\frac{H_0}{D} + \left(- \frac{\left(1 + \frac{s^2}{\omega_0^2}\right) H_0}{D} \right) \right) V_i \\ &= \left(\frac{\left(\frac{s^2}{\omega_0^2}\right) H_0}{D} \right) V_i \quad \text{mit } D = 1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2} \\ \Rightarrow \frac{V_1}{V_i} &= \frac{\left(\frac{s^2}{\omega_0^2}\right) H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} \end{aligned} \quad (13)$$

Fall 2: Verwendung der positiven Form der Übertragungsfunktion von V_4 wie im Aufgabenheft angegeben:

$$\frac{V_4}{V_i} = \frac{\left(1 + \frac{s^2}{\omega_0^2}\right) \cdot H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} \quad (14)$$

Auch hier wird wieder in Gleichung (5) eingesetzt:

$$\begin{aligned} V_1 &= -(V_3 + V_4) = - \left(\frac{H_0}{D} + \frac{\left(1 + \frac{s^2}{\omega_0^2}\right) H_0}{D} \right) V_i \\ &= - \left(\frac{\left(2 + \frac{s^2}{\omega_0^2}\right) H_0}{D} \right) V_i \\ \Rightarrow \frac{V_1}{V_i} &= - \frac{\left(2 + \frac{s^2}{\omega_0^2}\right) H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} \end{aligned} \quad (15)$$

Wiederholte Berechnung der \ddot{U} -Funktion $\frac{V_2}{V_i}$ über V_1 :

Ausgehend von der zuvor hergeleiteten korrekten Übertragungsfunktion für V_1 (Gleichung (13)):

$$\frac{V_1}{V_i} = \frac{\frac{s^2}{\omega_0^2} \cdot H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}}$$

Verwenden der bekannten Beziehung $V_2 = -\frac{\omega_0}{s} \cdot V_1$:

$$\begin{aligned} V_2 &= -\frac{\omega_0}{s} \cdot V_1 \\ &= -\frac{\omega_0}{s} \cdot \frac{\frac{s^2}{\omega_0^2} \cdot H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} \cdot V_i \\ &= -\frac{\frac{s}{\omega_0} \cdot H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} \cdot V_i \\ \Rightarrow \frac{V_2}{V_i} &= -\frac{\frac{s}{\omega_0} H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} \end{aligned} \quad (16)$$

Ein Vergleich der beiden Übertragungsfunktionen $\frac{V_2}{V_i}$ (Gleichung (9), (16)) zeigt, dass der entstandene Fehler auf die Gleichung $\frac{V_4}{V_1}$ zurückzuführen ist, da die Herleitung von $\frac{V_1}{V_i}$ korrekt bleibt.

Fazit: Nur die Verwendung der negierten Form von $\frac{V_4}{V_i}$ aus Gleichung (12) führt zu einem konsistenten Ergebnis für $\frac{V_1}{V_i}$ gemäß Gleichung (13). Die Version aus dem Aufgabenheft (Gleichung (14)) scheint daher fehlerhaft zu sein.

2.2 Herleitung der Übertragungsfunktion über die systemtheoretische Analyse

Die Übertragungsfunktionen eines Systems können nicht nur rechnerisch bestimmt werden, sondern lassen sich auch mithilfe systemtheoretischer Methoden herleiten. Im Folgenden wird dies exemplarisch anhand der Übertragungsfunktion $\frac{V_2}{V_i}$ demonstriert.

Dazu werden zunächst die für die einzelnen Operationsverstärker aufgestellten Gleichungen (5),(6),(7) und (8) betrachtet und anschließend systemtheoretisch, das heißt in Form eines Blockschaltbilds, modelliert. Dieses Blockschaltbild veranschaulicht den signaltheoretischen Zusammenhang der Teilsysteme und ermöglicht es, die Gesamtübertragungsfunktion durch die Verknüpfung der einzelnen Blöcke zu ermitteln. Als Ergebnis ergibt sich für die betrachtete Übertragungsfunktion das folgende Blockschaltbild:

Im Blockschaltbild des Biquad-Filters erkennt man anhand der beiden grau eingefärbten Integratoren deutlich, dass es sich um ein Filter zweiter Ordnung handelt. Der Begriff Biquad steht für biquadratisch und bezieht sich auf die mathematische Struktur der Übertragungsfunktion: Im Z- oder Laplace-Bereich ist diese das Verhältnis zweier quadratischer Polynome. Das spiegelt sich darin wider, dass die Übertragungsfunktion des Biquad-Filters zwei Pole und (meist auch) zwei Nullstellen aufweist.

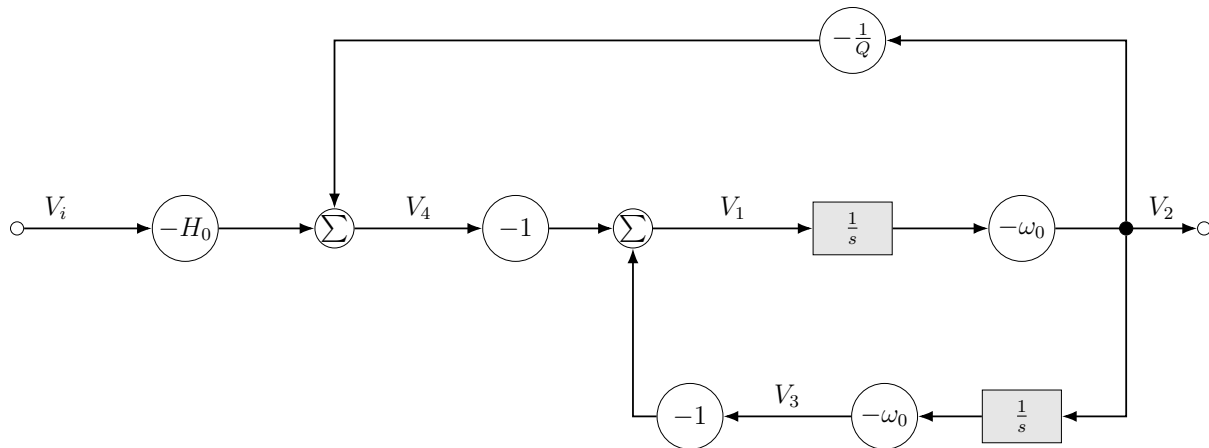


Abbildung 6: Blockschaltbild des Systems

Die Übertragungsfunktion $\frac{V_2}{V_i}$ lässt sich nun unmittelbar aus dem Blockschaltbild ablesen, indem die Verknüpfungsregeln für Reihen- und Parallelverschaltungen angewendet werden. So erhält man eine transparente Herleitung, die sowohl für die Analyse als auch für die Synthese analoger Systeme von großem Nutzen ist.

2.3 Einfluss der Bauteilgrößen C und R auf das Filterverhalten

Die Werte der im MF-Biquad eingebauten Widerstände und Kondensatoren bestimmen maßgeblich die Charakteristik des Filters. Insbesondere beeinflussen sie die Grenzfrequenz ω_0 , die Filtergüte und die maximale Verstärkung. Durch gezielte Wahl der Bauteilgrößen lässt sich das Frequenzverhalten des Biquad-Filters auf die jeweilige Anwendung abstimmen.

Die Grenzfrequenz ω_0 ergibt sich bei idealisierten Schaltungsvarianten nach folgender Formel:

$$\omega_0 = \frac{1}{RC} \quad (17)$$

Dabei beschreibt R den verwendeten Widerstandswert für alle (bis auf zwei) Widerstände und C den Kapazitätswert der zwei Kondensatoren.

Für die praktische Auslegung wird jedoch häufig die Kreisfrequenz $\omega_0 = 2\pi f_0$ mit der gewünschten Grenzfrequenz f_0 verwendet. Entsprechend lässt sich die Kapazität für die gewünschte Grenzfrequenz und den gegebenen Widerstandswert wie folgt berechnen:

$$C = \frac{1}{\omega_0 R} = \frac{1}{2\pi f_0 R} \quad (18)$$

Soll so beispielsweise ein Filter mit einer Kreisfrequenz von $\omega_0 = 1 \text{ kHz } (s^{-1})$ bei einem Widerstand von $R = 1 \text{ k}\Omega$ aufgebaut werden, ergibt sich die benötigte Kapazität wie folgt:

$$\begin{aligned} C &= \frac{1}{10^3 \text{ s}^{-1} \cdot 10^3 \Omega} \\ &= \frac{1}{10^6} = 1 \mu\text{F} \end{aligned}$$

Wird stattdessen ein Standardkapazitätswert von $C = 100 \text{ nF}$ verwendet, verschiebt sich die Grenzfrequenz nach:

$$f_0 = \frac{1}{2\pi RC} = \frac{1}{2\pi \cdot 10^3 \Omega \cdot 100 \times 10^{-9} \text{ F}} = 1,59 \text{ kHz}$$

Durch diesen Zusammenhang kann geschlussfolgert werden dass eine Verkleinerung der Kapazität um den Faktor 10 zu einer zehnfach höheren Grenzfrequenz führt. Das gilt entsprechend auch umgekehrt oder bei Anpassung des Widerstandswerts. Somit bestimmen die gewählten Bauteilwerte die tatsächlich erreichbare Frequenz des Filters. In der Praxis wird meist der Kondensatorwert angepasst, da hierbei nur zwei Bauteile geändert werden müssen, anstatt acht.

2.4 Einfluss von Güte- und Verstärkungsfaktor auf das Filterverhalten

Zur genaueren Analyse des Frequenzverhaltens von analogen Biquad-Filtern bietet sich die Betrachtung der Pol- und Nullstellen in der komplexen s -Ebene an. Die Lage der Punkte liefert direktes Verständnis über Filtergüte, Resonanz- und Dämpfungsverhalten sowie Stabilität des Systems. Im Laplace-Bereich gilt: $s = \sigma + j\omega$.

Als Beispiel wird die Übertragungsfunktion $\frac{V_2}{V_i}$ des Bandpassfilters betrachtet:

$$\frac{V_2}{V_i} = -\frac{\frac{s}{\omega_0} H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}}$$

Bei der Nullstellenbestimmung ergibt sich aus dem Zählerpolynom:

$$\frac{s}{\omega_0} H_0 = 0 \Rightarrow s_0 = 0 \quad (19)$$

Dies bedeutet, dass das System eine einfache Nullstelle im Ursprung besitzt. Gleichanteile im Eingangssignal ($s = 0$, also $\omega = 0$) werden somit vollständig unterdrückt.

Die Polstellen ergeben sich durch Nullsetzen des Nenners:

$$\begin{aligned} 1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2} &= 0 \\ s^2 + \frac{\omega_0}{Q} s + \omega_0^2 &= 0 \end{aligned}$$

Anwendung der pq-Formel:

$$s_{1,2} = -\frac{\omega_0}{2Q} \pm \sqrt{\left(\frac{\omega_0}{2Q}\right)^2 - \omega_0^2} \quad (20)$$

Vereinfachung des Radikanden:

$$\sqrt{\omega_0^2 \left(\frac{1}{4Q^2} - 1\right)} = \omega_0 \sqrt{\frac{1}{4Q^2} - 1}$$

Wie gut zu erkennen ist, hängt die Lage der Pole direkt vom Gütefaktor Q ab. Solange $Q > 0$ ist befinden sich die Pole in der linken Halbebene der s -Ebene, was die Stabilität des Systems garantiert. Je nach Wert von Q lassen sich drei charakteristische Fälle anhand der Impulsantwort unterscheiden:

- **Schwingfall** ($Q > \frac{1}{2}$): Komplex-konjugierte Pole, gedämpftes Schwingungsverhalten:

$$s_{1,2} = -\frac{\omega_0}{2Q} \pm j\omega_0 \sqrt{1 - \frac{1}{4Q^2}}$$

- **Aperiodischer Grenzfall** ($Q = \frac{1}{2}$): Doppelter reeller Pol, kritische Dämpfung:

$$s_{1,2} = -\omega_0$$

- **Kriechfall** ($Q < \frac{1}{2}$): Zwei reelle Pole, träges (überdämpftes) Verhalten:

$$s_{1,2} = -\frac{\omega_0}{2Q} \pm \omega_0 \sqrt{\frac{1}{4Q^2} - 1}$$

2.4.1 Einfluss von Q und H_0 auf das Frequenzverhalten

Das Frequenzverhalten von Biquad-Filtern zweiter Ordnung wird maßgeblich von der Güte Q und dem Verstärkungsfaktor H_0 bestimmt. Dabei unterscheidet sich der Zusammenhang zwischen Güte und Bandbreite je nach Filtertyp.

Bei Hoch- und Tiefpassfiltern stellt die Güte Q einen wichtigen Parameter zur Charakterisierung des Flanken- und Resonanzverhaltens dar. Eine Erhöhung des Gütefaktors führt zu einer steileren Filterflanke und ab $Q > 0,707$ führt dies zu einer Resonanzüberhöhung nahe der Grenzfrequenz.

Im Gegensatz dazu verfügen Bandpass- und Bandsperrfilter über eine deutlich ausgeprägte Mittenfrequenz ω_0 , welche das Zentrum des Durchlass- bzw. Sperrbereichs markiert. Die Bandbreite $\Delta\omega$ beschreibt den Abstand zwischen den beiden -3 dB-Grenzfrequenzen. Die folgende Gleichung [4] beschreibt den Zusammenhang zwischen Güte, Bandbreite und Mittenfrequenz.

$$\Delta\omega = \frac{\omega_0}{Q} \quad (21)$$

Ein größerer Gütefaktor Q führt also zu einer schmaleren Bandbreite und einer stärker ausgeprägten, spitzen Resonanz. In der s -Ebene spiegeln sich hohe Q -Werte darin wider, dass die Pole näher an die imaginäre Achse rücken. Dieser Zusammenhang gilt für alle Filtertypen, die auf der Biquad-Struktur basieren, da das Nennerpolynom bei allen Typen identisch aufgebaut ist.

Der Verstärkungsfaktor H_0 wirkt sich hingegen lediglich auf die Amplitudenhöhe des Frequenzgangs aus. Die Form der Übertragungsfunktion sowie die Lage der Pole bleiben von einer Veränderung von H_0 unberührt; die Resonanzfrequenz und die Bandbreite bleiben somit konstant.

Für den Bandpass-Filter zeigt sich folgende Pol-Nullstellen-Verteilung:

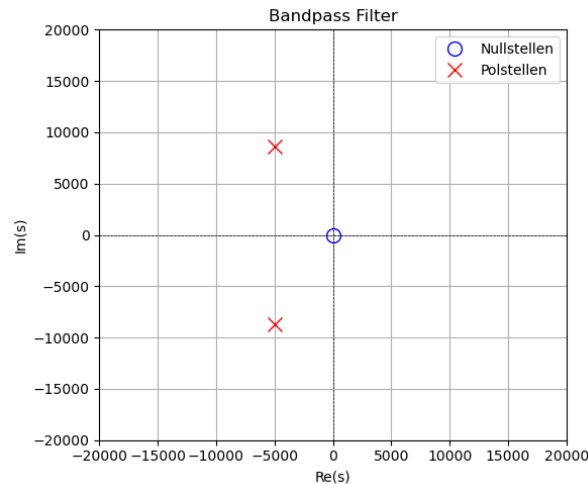


Abbildung 7: Pol- und Nullstellenansicht des Bandpass mit $Q = 1$

Angewandt auf die anderen Filtertypen ergibt sich dieses Pol-Nullstellendiagramm:

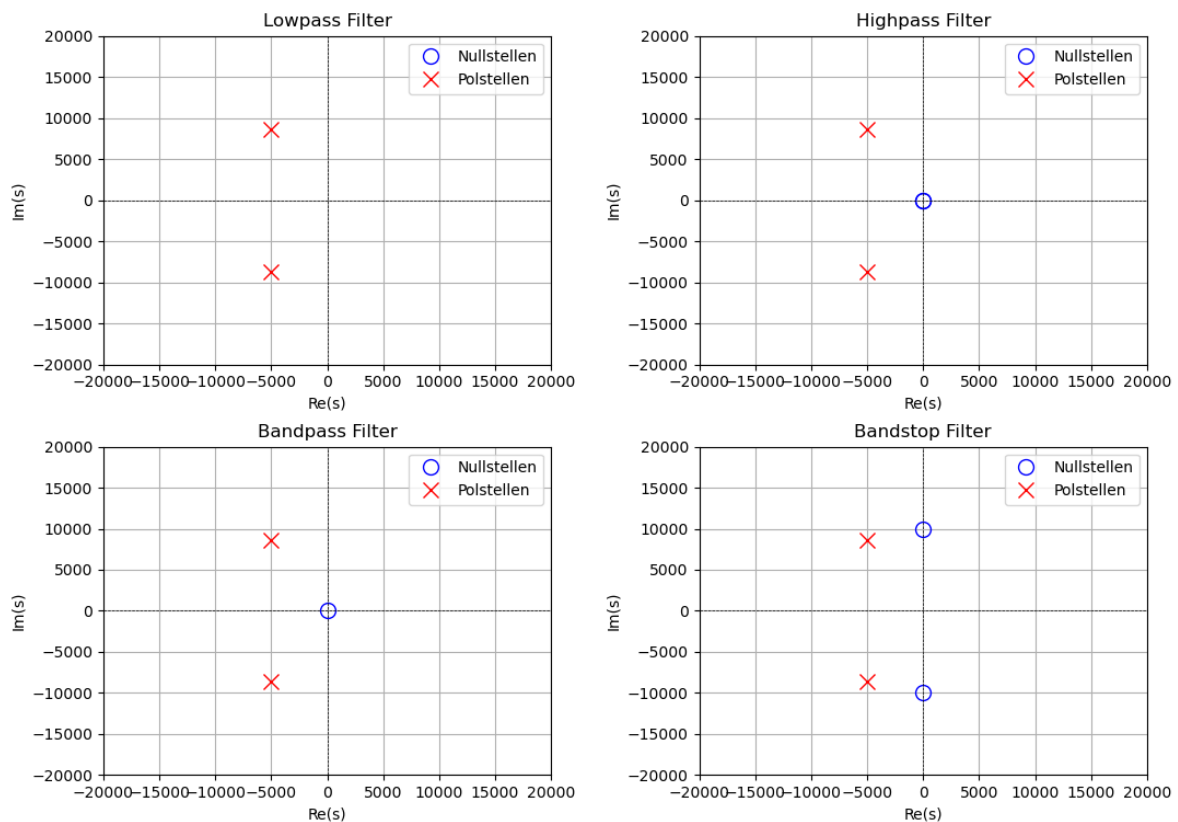


Abbildung 8: Pol- und Nullstellen in der S-Ebene

Hier ist auch nochmal gut zu erkennen, dass alle filtertypen die polstellen an der gleichen koordinaten ahben und dass das nennerpolynom überall gleich sein muss. Dabei ist deutlich zu erkennen, dass bei allen Filtertypen die Pole an denselben Koordinaten in der s -Ebene liegen. Dies verdeutlicht, dass das Nennerpolynom der Übertragungsfunktion für sämtliche Biquad-basierten Filter identisch ist. Der tatsächliche Filtertyp

(Tiefpass, Hochpass, Bandpass oder Bandsperre) wird lediglich durch die Auswahl und Anordnung der Nullstellen bzw durch den Zähler des Systems festgelegt. Die Polstellen und damit das Dämpfungs- und Resonanzverhalten wird durch die Parameter im Nenner bestimmt.

2.5 Herleitung einer Blockstruktur für ein analoges Biquad-Filter

Ein analoger Bandpass-Filter zweiter Ordnung lässt sich im Allgemeinen durch eine rationale Übertragungsfunktion der Form

$$H(s) = \frac{V_2(s)}{V_i(s)} = -\frac{\frac{s}{\omega_0} H_0}{1 + \frac{s}{\omega_0 Q} + \frac{s^2}{\omega_0^2}} \quad (22)$$

beschreiben. Diese entspricht einem sogenannten Biquad-Filter mit linearer Nullstelle im Zähler und zwei komplex konjugierten Polstellen im Nenner.

Zur systematischen Umsetzung dieser Übertragungsfunktion wurde das System zunächst in die sogenannte kanonische Reihen- bzw. Zustandsraumdarstellung (Second-Order Section, SOS) gebracht. Ausgangspunkt hierfür ist die Beobachtung, dass der Nenner auf eine Differenzialgleichung zweiter Ordnung hinweist, die sich durch Kaskadierung von zwei Integratoren als Kernbestandteile realisieren lässt. Durch geeignete Rückführung und Gewichtung der Zustandsgrößen sowie die Auswahl des Verstärkungsfaktors im Vorwärtszweig entsteht die hier dargestellte Blockstruktur.

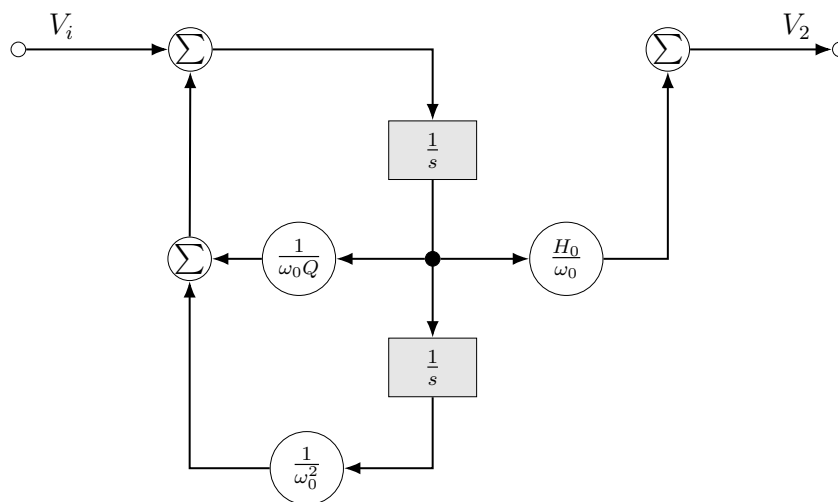


Abbildung 9: SOS der Bandpass Übertragungsfunktion

Die Blockstruktur in Abbildung 9 entspricht direkt der mathematischen Übertragungsfunktion des Biquad-Filters und bietet durch ihre klare Trennung von Vorwärts- und Rückführpfaden eine ideale Grundlage für Analyse und Simulation. Besonders vorteilhaft ist, dass diese kanonische Darstellung eine unkomplizierte Anwendung der Bilineartransformation zur Digitalisierung des Filters ermöglicht. Zudem sorgt die bereits normierte Struktur für numerische Stabilität und erleichtert den modularen Aufbau komplexerer Filtersysteme.

2.6 Vergleich von analogen und digitalen Filter

Filter lassen sich auf unterschiedliche Weise realisieren. Sie können sowohl analog in Hardware, als auch digital als Algorithmus auf einem Prozessor realisiert werden. Beide Arten der Implementierung haben ihre Vor- und Nachteile. Die Wahl der geeigneten Implementierung hängt von den technischen Anforderungen an den Filter ab.

Der erste Unterschied zwischen den beiden Filtern ist die Art, wie die Filter arbeiten. Analoge Filter arbeiten, wie der Name schon erahnen lässt, zeit- und wertekontinuierlich. Digitale Filter hingegen sind zeit- und wertediskret. Digitale Filter erfordern vor der Filterung eine Abtastung und nach der Filterung eine Rekonstruktion des Signals. Das zieht mehrere Nachteile mit sich. Zu einem muss das Abtasttheorem erfüllt sein.

$$f_S = 2 \cdot f_{max} \quad (23)$$

Um ein f_{max} begrenztes Signal aus den Abtastwerten fehlerfrei rekonstruieren zu können, muss die Abtastfrequenz f_S mindestens doppelt so groß sein wie f_{max} . Das limitiert auch den Frequenzbereich, in welchen der Filter eingesetzt werden kann. Ist die Abtastfrequenz $f_S < 2 \cdot f_{max}$ kommt es zum Aliasing-Effekt, wo höhere Frequenzen als niedrigere interpretiert werden. Aus diesem Grund wird meist noch ein analoger Tiefpass vor den Eingang des ADC geschaltet, um den Aliasing-Effekt zu verhindern.

Der Dynamikumfang des digitalen Filters wird durch das Quantisierungsrauschen des ADC und des DAC negativ beeinflusst. Der Dynamikumfang ist das Verhältnis zwischen der kleinsten vom Rauschen unterscheidbare Amplitude zu der größtmöglich unverzerrten Amplitude.[5]

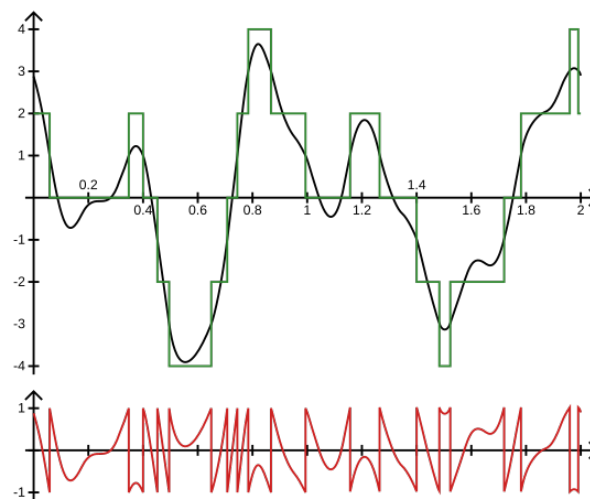


Abbildung 10: Beispiel für die Quantisierung. Schwarz: Ursprungssignal, Grün: Quantisiertes Signal, Rot: Abweichung der Quantisierung[6]

Als Quantisierungsrauschen wird der Fehler, welcher bei der Quantisierung der Amplituden auftritt, bezeichnet. Bei der Quantisierung wird ein wertkontinuierliches Signal in ein wertdiskretes Signal umgewandelt. Der dabei auftretenden Rundungsfehler kann als Rauschen interpretiert werden[7]

Die mittlere Leistung des Rundungsfehlers σ_{E_Q} ergibt sich aus

$$\sigma_{E_Q} = \frac{1}{\Delta} \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} e_Q de_Q \quad (24)$$

Wobei e_Q der Rauschanteil des diskretisierten Signals ist. Δ ist die Höhe der einzelnen Quantisierungsstufe.

$$\Delta = \frac{2}{Q} = \frac{2}{2^q} \quad (25)$$

Der Exponent q ist die Wortgröße oder Bittiefe des ADC und bestimmt die Auflösung und damit auch den Quantisierungsfehler. Je größer q wird, desto kleiner wird das Quantisierungsrauschen. Das Signal-Rausch-Verhältnis kann dann mit

$$SNR|_{dB} = 10 \cdot \log_{10} \left(\frac{P}{\sigma_{E_Q}^2} \right) \approx 1,76 + 6q \quad (26)$$

bestimmt werden.

Neben dem eingeschränkten Frequenzbereich und Dynamikumfang bringen digitale Filter auch eine höhere Latenz mit sich, als analoge Filter. Neben den Verzögerungen durch die Abtastung, Quantisierung und anschließende Rekonstruktion kommt auch noch die Rechenzeit hinzu, welcher der Prozessor benötigt um das gefilterte Signal zu berechnen. Mit steigender Filterordnung steigt auch die benötigte Rechenzeit an.

Allerdings haben digitale Filter auch Vorteile gegenüber analogen Filtern. Digitale Filter können einfacher angepasst, entwickelt und auch implementiert werden, da diese nur Anpassungen im Code erfordern. Bei analogen Filtern hingegen müssen die Werte der Bauteile verändert werden oder es muss ggf. eine komplett neue Schaltung entworfen werden.

Analoge Filter sind abhängig von ihren Komponenten. Eine hohe Toleranz bei den Bauteilen macht eine reproduzierbare Performance eines Filterdesigns über mehrere Schaltungen schwierig. Auch können Komponenten im Laufe der Zeit altern, was auch die Performance negativ beeinflussen kann. Ebenfalls sind die analogen Filter empfindlicher gegenüber Störungen im Arbeitsumfeld. Zum Beispiel können Temperaturschwankung ungewollte Frequenzverschiebung mit sich bringen.

Ein weiterer Vorteil von digitalen Filtern ist der physikalische Platzbedarf. Mit wachsender Filterordnung benötigt ein analoger Filter immer mehr Platz. Ein digitaler Filter hat immer den gleichen physikalischen Platzbedarf, unabhängig von der Filterordnung.

3 Design eines digitalen Biquads

Die Entwicklung eines digitalen Filters ist ähnlich zu der Entwicklung eines analogen Filters. In beiden Fällen wird die Übertragungsfunktion des Filters so angepasst, bis diese den gewünschten Anforderungen entspricht. Der Unterschied ist, dass die Übertragungsfunktion der analogen Filter auf der Laplace-Ebene ist und die der digitalen Filter auf der Z-Ebene.

3.1 Bilineare Transformation

Ein analoger Biquad-Filter ist IIR-Filter. Ein IIR-Filter besitzt eine unendliche Impulsantwort. Digitale IIR-Filter werden aus den analogen IIR-Filter entworfen. Dafür wird die analoge Übertragungsfunktion $H(s)$ mithilfe der bilinearen Transformation in die digitale Übertragungsfunktion umgewandelt.

Die bilineare Transformation ist in 27 angegeben.

$$s = \frac{2}{T_d} \cdot \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right) \quad (27)$$

Das Abtastintervall ist dabei durch T_d gegeben. Die bilineare Transformation begrenzt die Frequenzantwort des Filters so, dass diese innerhalb der Niquist-Frequenz liegt, welche durch mit $\frac{2}{T_d}$ gegeben ist. So wird der Aliasing Effekt verhindert.

Die bilineare Transformation bildet die imaginäre Achse der s-Ebene auf den Einheitskreis in der z-Ebene ab. Die Beziehung zwischen dem kontinuierlichen Frequenzbereich $-\infty \leq \Omega \leq \infty$ und des diskret Frequenzbereiches $-\pi \leq \omega \leq \pi$ ist stark nicht linear. Die Beziehung kann mit der Gleichung 28 beschrieben werden.

$$\omega = 2 \cdot \arctan \left(\frac{\Omega \cdot T_d}{2} \right), \Omega = \frac{2}{T_d} \cdot \tan \left(\frac{\omega}{2} \right) \quad (28)$$

Die nichtlineare Beziehung zwischen der s-Ebene und der z-Ebene führt zu einer Frequenzverzerrung, welche Frequency Warping genannt wird.

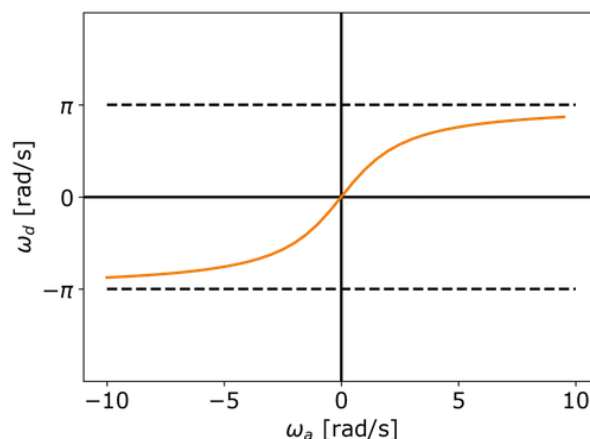


Abbildung 11: Nicht lineare Beziehung zwischen der s-Ebene und der z-Ebene[8]

Das Frequency Warping beeinflusst die Frequenzantwort von digitalen Filtern. Bei niedrigen Frequenzen, nahe $\omega = 0$ bzw. 0 Hz, ist die Frequenzverzerrung minimal, wie auch in

Abbildung 11 zu erkennen. Das bedeutet, dass sich analoge und digitale Filter in diesem Bereich nahezu identisch verhalten. Die Verzerrung nimmt mit steigender Frequenz weiter zu und erreicht ihr maximum nahe der Niquist-Frequenz $\frac{f_s}{2}$, bzw. π im zeitdiskreten Bereich.[9]

3.2 Prewarping

Um die Frequenzverzerrung zu kompensieren wird Prewarping eingesetzt. Durch das Prewarping wird die analoge Übertragungsfunktion $H(s)$ vor verzerrt, sodass die digitale Übertragungsfunktion $H(z)$ nach der bilinearen Transformation den gewünschten Vorgaben entspricht.[9]

$$\Omega_s = \frac{2}{Td} \cdot \tan\left(\frac{\omega_{eck}}{2}\right) = 2 \cdot f_s \cdot \tan\left(\frac{\omega_{eck}}{2}\right) \quad (29)$$

Typischerweise wird das in Gleichung 29 beschriebene Prewarping auf die Eckfrequenzen angewendet. Zu den Eckfrequenzen gehören die Pass- und Stopbandfrequenzen von Tief- und Hochpässen, sowie Mittenfrequenzen von Bandpässen und Bandsperren.[9]

Allerdings ist Prewarping nicht zwingend notwendig. Wird die Abtastfrequenz $f_s \gg f_{eck}$ ist Prewarping nicht erforderlich und kann sogar $H(s)$ zu stark verzerren.[9]

Diese Gegebenheit resultiert aus der vorherigen Erkenntnis, dass die Frequenzverzerrung bei niedriger Frequenz minimal ist. Die Abtastfrequenz f_s darf aber auch nicht zu hoch gewählt werden, da sonst die Eckfrequenzen verschoben werden.

3.3 Filter Strukturen

Es gibt viele verschiedene Filterstrukturen, um eine gegebene Übertragungsfunktion zu implementieren. Die Übertragungsfunktion eines IIR-Filters ist durch 2N-1 Koeffizienten beschrieben und mit mehreren Multiplizieren und Addierer implementiert werden.

In der Theorie macht es keinen Unterschied, welche Art an Filterstruktur gewählt wird. Alle Filterstrukturen bilden dieselbe Übertragungsfunktion ab. Das gilt allerdings nur, wenn die Filterstrukturen mit unendlicher Genauigkeit implementiert werden können. In der Praxis jedoch ist nur eine Implementierung mit endlicher Genauigkeit möglich. Das führt dazu, dass sich die verschiedenen Filterstrukturen unterschiedlich verhalten.

Die endliche Genauigkeit kann zu einer Verschiebung von Pol- und Nullstellen, Überläufern oder zu möglicherweise instabilen Filtern führen. Ebenfalls können sich Filterstrukturen in ihrer Komplexität, Durchsatzraten oder Phasen- und Gruppenlaufzeit unterscheiden.

3.3.1 Direktform 1 und 2

Filterstrukturen, bei denen die Koeffizienten der Multiplizierer, denen der Übertragungsfunktion $H(z)$ entsprechen, werden als Direktform Filterstrukturen bezeichnet.

Gegeben in 30 ist die Übertragungsfunktion $H(z)$ eines IIR-Filters zweiter Ordnung.

$$H(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}} \quad (30)$$

Die Übertragungsfunktion ist auf a_0 normiert. Diese Übertragungsfunktion lässt sich als zwei Filterteile implementieren. Dafür sollte die dazugehörige Differenzengleichung 31, welche sich aus der Übertragungsfunktion $H(z)$ in 30 ergibt, betrachtet werden.

$$y[n] = b_0 \cdot x[n] + b_1 \cdot x[n-1] + b_2 \cdot x[n-2] - a_1 \cdot y[n-1] - a_2 \cdot y[n-2] \quad (31)$$

Der erste Filterteil bilden die verzögerten Eingangswerte, welche durch entsprechende Koeffizienten gewichtet und zusammen addiert werden.

Die rückgekoppelten Ausgangswerte bilden den zweiten Filterteil. Auch diese werden verzögert und entsprechend gewichtet. Im Gegensatz zu den Eingangswerten sind diese allerdings negativ gewichtet. Anschließend werden die beiden Filterteil miteinander addiert.

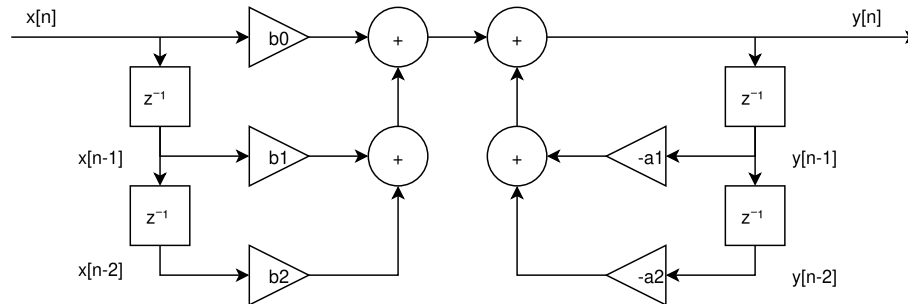


Abbildung 12: Umgesetzte Differenzengleichung in Direkt Form 1

In Abbildung 12 ist die resultierende Direkt Form 1 dargestellt. Die Direkt Form 1 benötigt für die gegebene Differenzengleichung 31 vier Verzögerungsglieder, zwei für die Eingangswerte und zwei für die rückgekoppelten Ausgangswerte. Damit ist diese Art der Implementierung nicht kanonisch. Nicht kanonisch bedeutet, dass mehr Verzögerungsglieder benötigt werden, als von der Ordnung des Filters vorgeben wird.

Die Direkt Form 1 ist einfach umzusetzen, da diese eine direkte Umsetzung der Differenzengleichung darstellt. Ihr großer Vorteil liegt in ihrer Robustheit gegenüber Quantisierungsfehler bei Koeffizienten. Die Pfade für Feedforward und Feedback sind voneinander getrennt. So beeinflussen Quantisierungsfehler in einem Pfad nicht direkt den anderen Pfad. Dies führt zu einer verbesserten numerischen Stabilität.[9]

Der Nachteil der nicht kanonische Aufbau der Direkt Form 1. Es werden mehr Verzögerungsglieder benötigt, als die Ordnung vorgibt

Dieses Problem wird durch die Direkt Form 2 gelöst. Mit der Direkt Form 2 wird die Anzahl der benötigten Verzögerungsglieder auf das Minimum reduziert. Die Anzahl entspricht dann der Filterordnung. Damit ist diese Filterstruktur kanonisch.

Die Direkt Form 2 macht sich die Kommutivität von LTI-Systemen zunutze. Sie vertauscht die Reihenfolge der Verarbeitung von Ein- und Ausgang.[9]

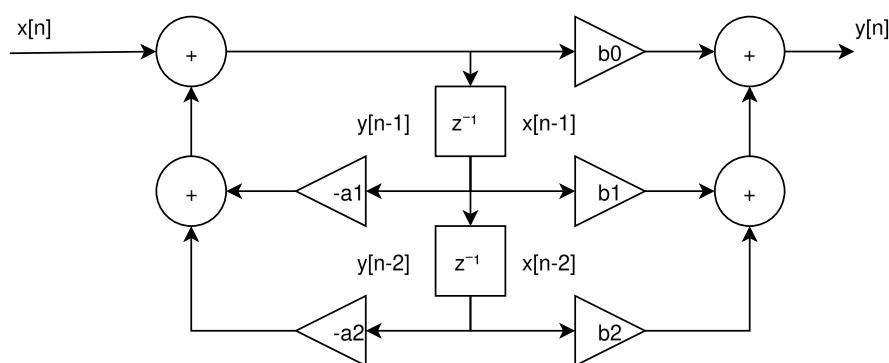


Abbildung 13: Umgesetzte Differenzengleichung in Direkt Form 2

In Abbildung 13 ist die Differenzengleichung 31 in Direkt Form 2 umgesetzt. Durch das Vertauschen der Reihenfolge können immer zwei Verzögerungsglieder zusammengefasst werden. Dadurch wird die Anzahl der benötigten Verzögerungsglieder halbiert.

Diese Form der Implementierung weist jedoch eine höhere Anfälligkeit gegenüber Quantisierungsfehlern auf, als die Direkt Form 1. Das kann das SNR am Ausgang zusätzlich negativ beeinflussen. Der Vorteil der Direkt Form 2 ist ihr kanonischer Aufbau und damit kleinerer Bedarf an Verzögerungsgliedern.[9]

3.3.2 Kaskadierte Form und SOS-Strukturen

Bei der kaskadierten Form werden mehrere einzelne Filterstrukturen seriell miteinander verschaltet.

Die einzelnen Filterstrukturen sind sogenannte Second Order Sections oder auch SOS-Strukturen. Zur Implementierung werden die Polynome der Übertragungsfunktion $H(z)$ in mehrere Polynome ersten und zweiten Grades faktorisiert.

$$H(z) = \rho_0 \prod_k \left(\frac{1 + \beta_{1k} \cdot z^{-1} + \beta_{2k} \cdot z^{-2}}{1 + \alpha_{1k} \cdot z^{-1} + \alpha_{2k} \cdot z^{-2}} \right) \quad (32)$$

Im Falle einer Übertragungsfunktion erster Ordnung sind $\alpha_{2k} = 0$ und $\beta_{2k} = 0$. Somit entsteht nur ein Polynom ersten Grades. Für eine Übertragungsfunktion zweiten Grades entsteht nur ein SOS-Struktur.

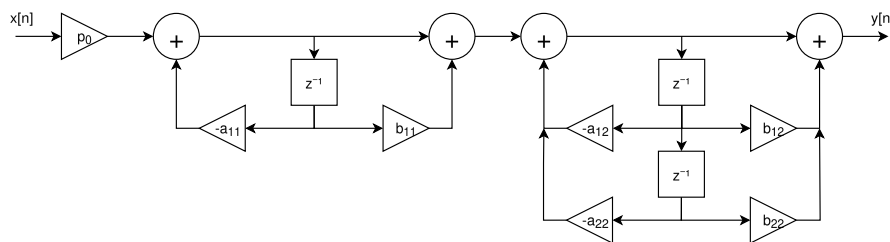


Abbildung 14: Umgesetzte Differenzengleichung in Direkt Form 2

Die Abbildung 14 zeigt eine Kaskadierte Filterstruktur mit einzelnen SOS-Strukturen. Zum besseren Verständnis wird eine IIR-Filter 3-ten Grades gewählt. So entsteht eine First Order Section gefolgt von einer Second Order Section. Die dazugehörige Übertragungsfunktion ist in 33 gegeben.

$$H(z) = \rho_0 \left(\frac{1 + b_{11} \cdot z^{-1}}{1 + a_{11} \cdot z^{-1}} \right) \left(\frac{1 + b_{12} \cdot z^{-1} + b_{22} \cdot z^{-2}}{1 + a_{12} \cdot z^{-1} + a_{22} \cdot z^{-2}} \right) \quad (33)$$

Es ist auch möglich andere Filterstrukturen als SOS-Strukturen zu kaskadieren.

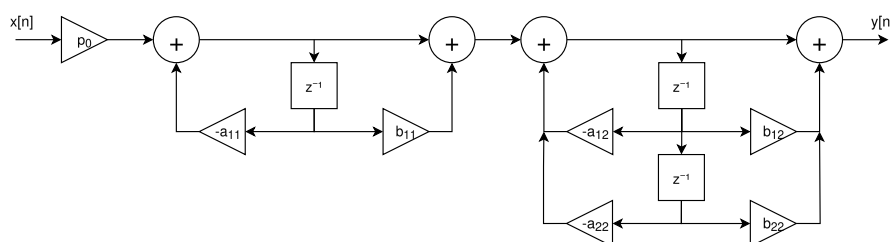


Abbildung 15: Umgesetzte Differenzengleichung in Direkt Form 2

Die Abbildung 15 zeigt die Direkt Form 2 in Kaskadierte Struktur. Auch hier wird zum besseren Verständnis eine Übertragungsfunktion 3-ten Grades gewählt. Diese ist in 34 gegeben.

$$H(z) = \left(\frac{b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + b_3 \cdot z^{-3}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + a_3 \cdot z^{-3}} \right) = \left(\frac{z^{-1}}{1 - a_1 \cdot z^{-1}} \right) \left(\frac{b_1 + b_2 \cdot z^{-1} + b_3 \cdot z^{-2}}{1 + a_2 \cdot z^{-1} + a_3 \cdot z^{-2}} \right) \quad (34)$$

Der Vorteil von SOS-Strukturen ist, dass diese numerisch stabil sind. Bei Filtern mit hoher Ordnung können kleine Rundungsfehler bei der Koeffizienten zu Instabilitäten oder unerwünschten Verhalten führen. Durch die Kaskadierung in mehrere SOS-Strukturen kann dieses Problem reduziert und dennoch ein Filter mit hoher Ordnung realisiert werden.[10] Weiterhin sind kaskadierte Filterstrukturen sehr gut skalierbar. Somit können Filter mit hoher Ordnung schnell und einfach realisiert werden.

3.4 Implementieren eines digitalen Biquad in Python

Designt wird eine digitale Version des zuvor designten analogen Biquad Filter. Der digitale Biquad soll dabei die Charakteristiken aufweisen, wie der analoge Biquad Filter. Ziel dabei soll sein, zuvor angeeignetes Wissen in den Modulen DSP und ANS anzuwenden und zu vertiefen, sowie die theoretische Frequenzantwort des analogen und digitalen Biquad-Filter miteinander zu vergleichen.

Eine große Hilfe dabei stellt das Github Repository von Herr Funkner und Herr Schwarz[9]. Beide haben bei der Entwicklung eines Testskripts und bei theoretischem Hintergrundwissen weitergeholfen.

Zur Umsetzung und Implementierung der Filter wird von Scipy das Signal Modul verwendet. Ebenfalls werden numpy und matplotlib.pyplot zum Speichern, auswerten und Plotten verwendet. Zum Testen der designten Filter wird zusätzlich noch pyAudio verwendet. PyAudio ermöglicht es in Python Audio einfach wiederzugeben, aufzunehmen und live Audio Streams einzubinden. Letzters wird zum Testen des digitalen Biquad-Filters verwendet.

Als Grundlage dienen die einzelnen Übertragungsfunktionen für den Tiefpass (3.4), Hochpass (3.4), Bandsperre (3.4) und Bandpass (3.4) des analogen Biquad-Filter.

$$H_{LP}(s) = \frac{H_0}{1 + \frac{s}{\omega_0 \cdot Q} + \frac{s}{\omega_0}} \quad (35)$$

$$H_{HP}(s) = \frac{H_0 \cdot \frac{s}{\omega_0}}{1 + \frac{s}{\omega_0 \cdot Q} + \frac{s}{\omega_0}} \quad (36)$$

$$H_{BS}(s) = \frac{-H_0 \cdot \frac{s}{\omega_0}}{1 + \frac{s}{\omega_0 \cdot Q} + \frac{s}{\omega_0}} \quad (37)$$

$$H_{BP}(s) = \frac{\left(1 + \frac{s}{\omega_0}\right) \cdot (-H_0)}{1 + \frac{s}{\omega_0 \cdot Q} + \frac{s}{\omega_0}} \quad (38)$$

Die Grenzkreisfrequenz ω_0 wird zunächst mit 39 bestimmt.

$$\omega_0 = \frac{1}{R \cdot C} \quad (39)$$

Gewählt wird für den Widerstand $R = 1000\Omega$ und für den Kondensator $C = 1\mu F$. Als Verstärkung wird $H = 1$ und als Güte $Q = 1$ gewählt. Daraus ergeben sich dann für die Übertragungsfunktionen:

$$H_{LP}(s) = \frac{1}{1 + \frac{s}{1000\frac{1}{s}} + \frac{s^2}{1 \cdot 10^6 \frac{1}{s^2}}}$$

$$H_{HP}(s) = \frac{1 \cdot \frac{s^2}{1 \cdot 10^6 \frac{1}{s^2}}}{1 + \frac{s}{1000\frac{1}{s}} + \frac{s^2}{1 \cdot 10^6 \frac{1}{s^2}}}$$

$$H_{BS}(s) = \frac{-1 \cdot \frac{s}{1000\frac{1}{s}}}{1 + \frac{s}{1000\frac{1}{s}} + \frac{s^2}{1 \cdot 10^6 \frac{1}{s^2}}}$$

$$H_{BP}(s) = \frac{\left(1 + \frac{s^2}{1 \cdot 10^6 \frac{1}{s^2}}\right) \cdot (-1)}{1 + \frac{s}{1000\frac{1}{s}} + \frac{s^2}{1 \cdot 10^6 \frac{1}{s^2}}}$$

Diese Übertragungsfunktionen werden dann mithilfe der Funktion *bilinear(num,den,fs)* aus dem Signal Modul von Scipy von der s-Ebene in die z-Ebene transformiert. Als Abtastfrequenz wird $f_s = 44100$ Hz gewählt. Da die Abtastfrequenz $f_s \gg f_g$ als die 3dB-Grenzfrequenz gewählt wird, ist das Prewarping nicht erforderlich.

Ebenfalls werden von Signal die Funktion *freqs(num,den)* und die Funktion *freqz(num,den,fs)* verwendet. Mit *freqs(num,den)* lässt sich die Amplitudenantwort des analogen Filters und mit *freqz(num,den,fs)* die Amplitudenantwort des digitalen Filters ermitteln.

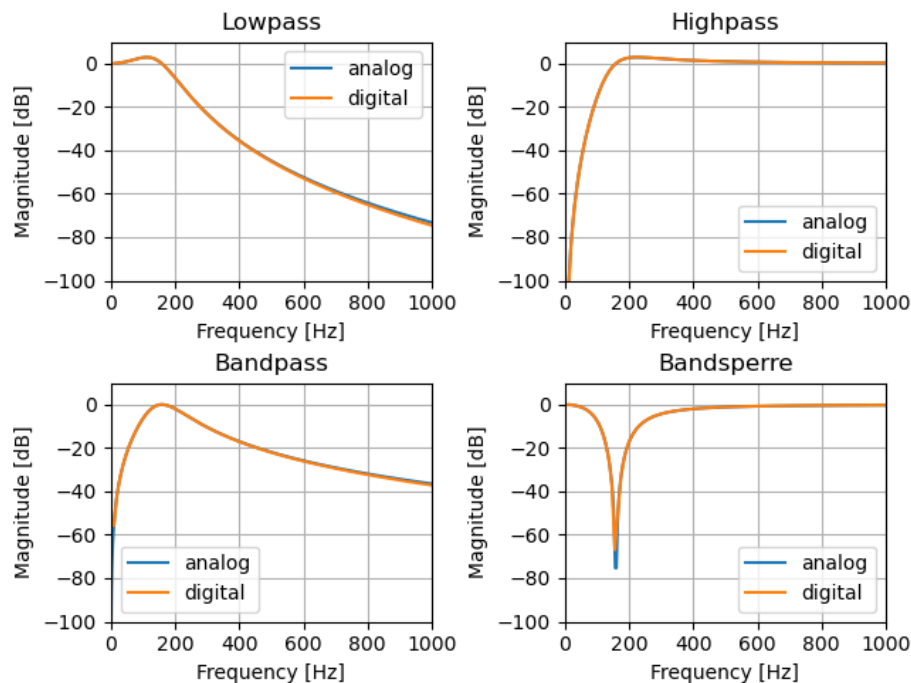


Abbildung 16: Vergleich zwischen der analogen und digitalen Amplitudenantwort

Der Vergleich zwischen der Amplitudenantwort des analogen Biquad-Filter und der des digitalen Biquad-Filters ist in Abbildung 16 für alle Filtertypen dargestellt. Zu sehen ist, dass die digitale Amplitudenantwort bei allen Filtertypen nahezu identisch mit der Amplitudenantwort der analogen Biquad-Filter ist. Nur bei der Bandsperre weist die

digitale Version mit $-74.9dB$ eine bessere Dämpfung an der Mittenfrequenz auf, als die analoge Version mit $-65.7dB$. Erwartet wird beim Tief- und Hochpass eine 3dB-Grenzfrequenz von $f_g = 159.15Hz$ und bei Bandpass und Bandsperre wird eine Mittenfrequenz $f_M = f_g = 159.15Hz$ erwartet. Sowohl die analogen, als auch die digitalen Biquad-Filter weisen die erwarteten Grenz- und Mittenfrequenzen auf. Das bedeutet, dass keine oder nur eine sehr minimale Frequenzverzerrung (Frequency Warping) bei der bilinearen Transformation aufgetreten ist.

Um den digitalen Biquad-Filter auf Stabilität zu überprüfen, müssen die Polstellen der einzelnen Filtertypen kontrolliert werden. Mit der Funktion $tf2zpk(num, den)$ aus dem Signal Modul von Scipy können die Pol- und Nullstellen der Übertragungsfunktion bestimmt werden.

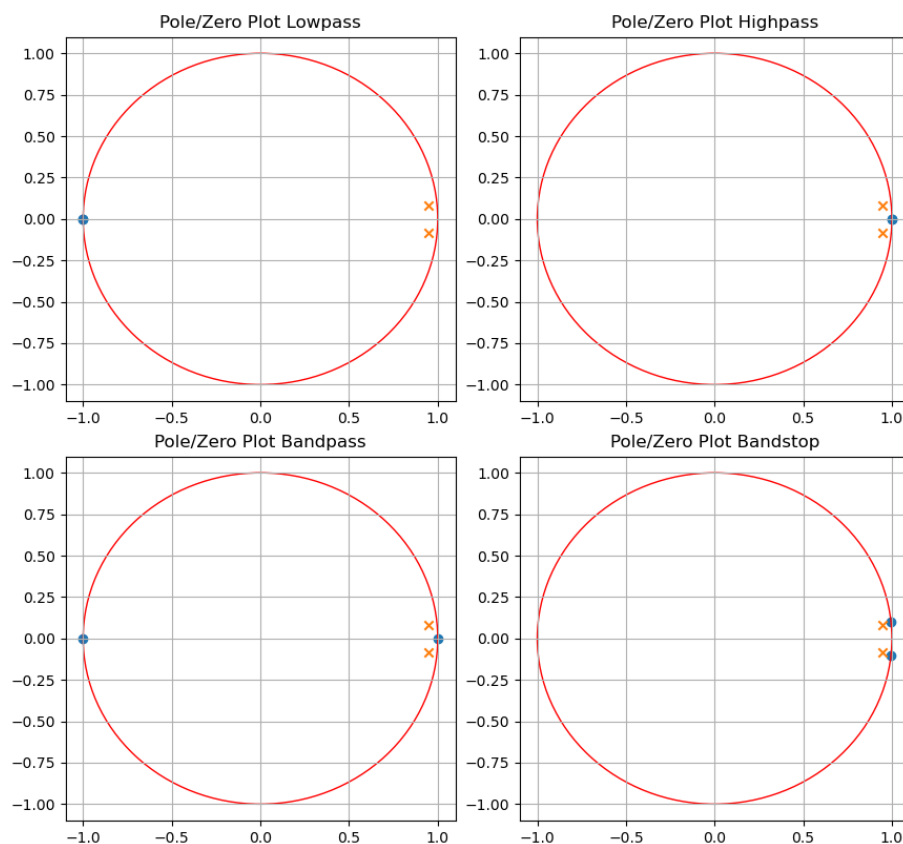


Abbildung 17: Pol- und Nullstellen aller Filtertypen des digitalen Biquad-Filter

Die Pol- und Nullstellen aller Filtertypen des digitalen Biquad-Filters sind in Abbildung 17 dargestellt. Die orangefarbene Kreuze stellen die Polstellen und die blauen Punkte die Nullstellen dar. Der rote Kreis repräsentiert den Einheitskreis.

Für einen stabilen analogen Filter müssen alle Polstellen auf der linken Seite der s -Ebene liegen. Da die linke Seite der s -Ebene durch die bilineare Transformation in den Einheitskreis abgebildet wird, müssen somit für digitale Filter alle Polstellen im Einheitskreis liegen. Das ist bei allen Filtertypen des digitalen Biquad-Filters der Fall.

Je näher ein Pol dem Einheitskreis kommt, desto niedriger wird die Dämpfung der Schwingungen im Zeitbereich. Liegt ein Pol auf dem Einheitskreis würde dies bedeuten, dass die Schwingung im Zeitbereich weder gedämpft noch verstärkt wird. Ein Pol außerhalb des Einheitskreises sorgt dafür, dass sich die Schwingung im Zeitbereich immer weiter aufschwingt. Das hätte einen instabilen Filter zur Folge.

Um die Filter zu testen zu können, muss eine geeignete Filterstruktur gewählt werden. Da es sich bei Biquad-Filtern bereits um SOS-Strukturen handelt, wird eine Implementierung als SOS-Struktur gewählt. Das Modul Signal stellt dafür die Funktionen $tf2sos(num, den)$ und $sosfiltfilt(sos, data)$ bereit. Mit der Funktion $tf2sos(num, den)$ können alle Parameter für die SOS-Struktur aus der jeweiligen Übertragungsfunktion ermittelt werden. Mit pyAudio wird ein beliebiges Audiosignal in Python eingebunden. Bei dem Audiosignal handelt es sich um ein Playback des internen Audiostreams des Rechners. So kann z.B Musik in Echtzeit gefiltert werden.

Mit $sosfiltfilt(sos, data)$ kann die ermittelt SOS-Struktur verwendet werden, um Daten im Objekt *data* zu filtern. Als Testskript wird das `plotting_audio.ipynb` aus dem Github Repository von Herr Funkner und Herr Schwarz [9] entsprechend erweitert.

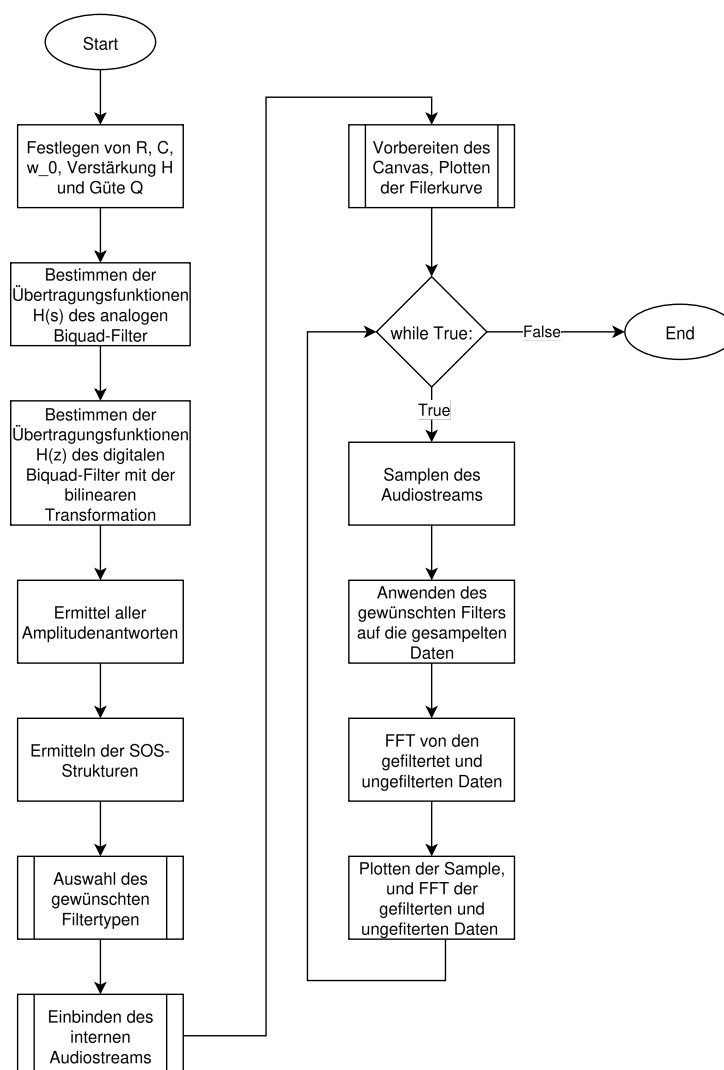


Abbildung 18: Ablaufplan des Testskripts für den digitalen Biquad-Filter

Der Ablaufplan für des modifizierten Testskript ist in Abbildung 18 dargestellt. Wie bereits erwähnt, stammt das ursprüngliche Programm aus dem Github Repository von Herr Funkner und Herr Schwarz[9]. Übernommen aus dem Programm werden das Vorbereiten des Canvas und plotten der Filterkurve im Canvas, Teile vom Einbinden des Audiostreams, Samplen des Audiostreams und das Plotten der Sample und die FFT der ungefilterten

Daten.

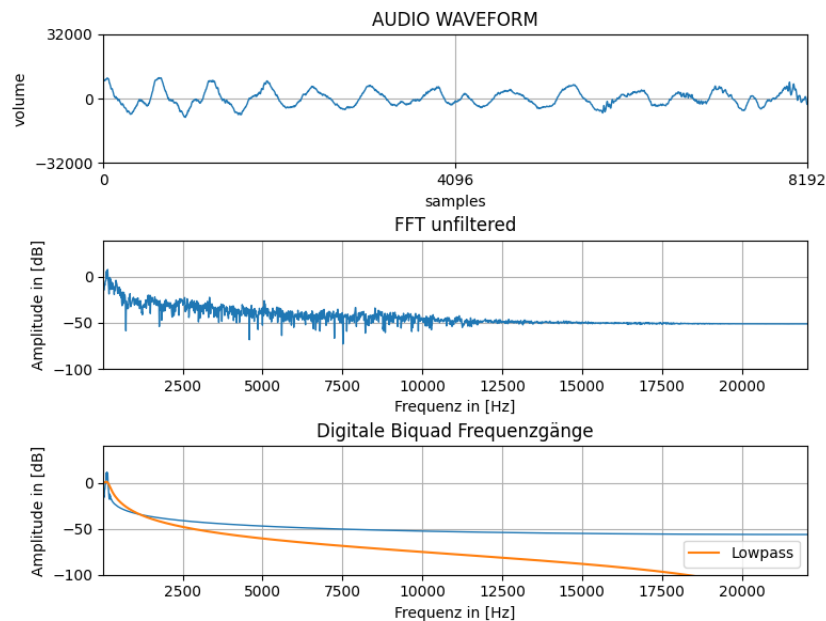


Abbildung 19: Ausgabe Fenster des Testprogramms

Als Ausgabe erzeugt das Programm ein Fenster, was die aktuellen Samples des Audiostreams, sowie eine FFT der gefilterten und unfilterten Samples. Das Fenster ist in Abbildung 19 zu sehen. Der obere Plot sind die aktuellen Samples. In der Mitte ist die FFT für die unfilterten Samples und die untere FFT sind die Samples nach dem Durchlaufen eines Tiefpasses mit der Grenzfrequenz $f_g = 159.15 \text{ Hz}$.

Es ist ein deutlicher Unterschied zwischen den gefilterten und unfilterten Samples zu sehen. Auch kann die Tiefpasscharakteristik klar erkannt werden. Jedoch folgt der angewandte Filter nicht ganz der eigentlich erwünschte Filterkurve, welche in Orange dargestellt ist. Im Stopband wird die gewünschte minimale Stopbanddämpfung nicht erreicht und im Passband wird die maximale Passbanddämpfung überschritten. Eine genaue Ursache dafür konnte noch nicht ermittelt werden.

4 Durchführung und Aufnahme der Messwerte

4.1 Softwaretools

Im Rahmen dieses Moduls kamen viele verschiedene Softwarewerkzeuge für Messung, Design oder Darstellung zum Einsatz.

Für die Erstellung des Schaltplans, des PCB-Layout und der Simulation von der Schaltung wurde überwiegend mit KiCad 9 gearbeitet. Das ist eine offene EDA-Software, die unter anderem vom CERN und einer internationalen Entwicklergemeinschaft aktiv weiterentwickelt wird. KiCad beinhaltet unter anderem eine sehr umfassende Komponentenbibliothek, einen integrierten Schaltplan- und PCB-Editor, 3D-Visualisierung, zahlreiche Exportformate sowie einen eingebetteten SPICE-Simulator für die Analyse analoger Schaltungen. So wurde KiCad in diesem Modul sowohl für die Simulation des Biquad-Filters als auch zur Erstellung der Schaltpläne und den Entwurf des Platinenlayouts verwendet.

Die tatsächliche Messwertaufnahme erfolgt neben der CSV-Ausgabe des RedPitayas über Python in Kombination mit der Entwicklungsumgebung Spyder. So konnten die Daten semiautomatisch über den RedPitaya sowie KiCad aufgenommen, weiterverarbeitet und gespeichert werden. Auch die Darstellung der Ergebnisse erfolgt in Python.

4.2 Simulation

Zu Beginn der Durchführung wurde die Simulation der Schaltung in KiCad mit dem integrierten SPICE-Simulator durchgeführt, da diese für das Verständnis der Biquad-Schaltung von großem Wert ist. Innerhalb der Simulationsoberfläche kam hauptsächlich die SPICE-Implementierung LTspice zum Einsatz.

In KiCad wurde zunächst der Schaltplan wie in Aufgabe 4 im ASLK-PRO-Manual [1] zusehen aufgebaut. Zusätzlich musste bei der Simulation auf die Versorgungsspannung der Operationsverstärker geachtet werden. Die Werte für Widerstände und Kondensatoren wurden mithilfe des `.param`-Befehls zentral definiert, sodass sie bei Änderungen nur an einer Stelle angepasst werden müssen und nicht bei jedem Bauteil einzeln. Auch die Parameter für den Gütefaktor Q und die Verstärkung H_0 wurden auf diese Weise implementiert.

Anschließend mussten den Bauteilen passende SPICE-Modelle zugewiesen werden. Das hier verwendete OpAmp-Modell TL082 lag in Form einer `.cir`-Datei vor. Die Einbindung der Modelle erfolgt über das Untermenü „Simulationsmodell“, in dem die Pinbelegung zwischen Schaltplanbauteil und SPICE-Modell zugeordnet werden kann.

Anfangs funktionierte diese Zuordnung nicht korrekt, da bei der Verwendung zweier Operationsverstärker (jeweils in einem DIP-8-Gehäuse) die Pinzuweisungen für beide Bauteile richtig vorgenommen werden mussten. Da beide Bauteile auf dasselbe SPICE-Modell verwiesen, wurde die Pinbelegung jeweils vom anderen überschrieben bzw. zurückgesetzt. Als Lösung wurden zwei Instanzen des Modells in einer gemeinsamen Datei (`TL082-dual.lib`) zusammengeführt. Danach trat dieses Problem nicht mehr auf.

Nach erfolgreicher Zuweisung der Modelle konnte die Simulation durchgeführt werden. Die Kleinsignalanalyse (AC-Analyse) wurde genutzt, um den Frequenzgang als Bode-Diagramm darzustellen.

Zur direkten Weiterverarbeitung der Rohdaten wurde eine `.RAW`-Datei erzeugt, die alle Ergebnisse der aktuellen Simulation speichert und bei jeder neuen Simulation überschrieben wird. Diese Datei konnte anschließend in Python eingelesen und zur grafischen Darstellung der Simulationswerte verwendet werden.

Im Folgenden ist der Verlauf der Amplituden der verschiedenen simulierten Filtertypen dargestellt. Dabei wurden sowohl der Gütefaktor Q als auch der Verstärkungsfaktor H_0 auf 1 gesetzt. Die Widerstandswerte betragen jeweils $1\text{k}\Omega$, und als Kapazitäten wurden $1\mu\text{F}$ angenommen.

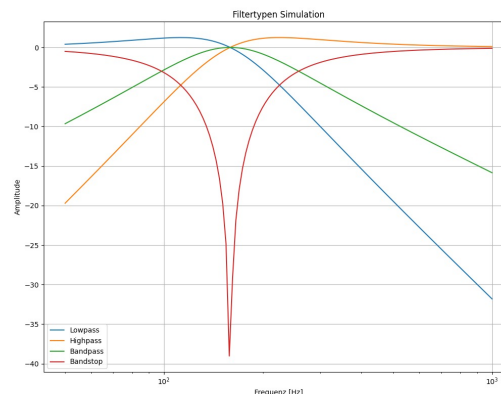


Abbildung 20: Darstellung des Frequenzverhaltens aller Ausgänge des Biquads

4.3 Messmittel

Die Messwerte werden mithilfe eines Red Pitaya STEMLab 125-14 v.1.1 aufgenommen. Dieser verfügt über zwei 125 MS/s 14-Bit Eingänge und über zwei 14-Bit Ausgänge. Der Red Pitaya STEMLab 125-14 v.1.1 ist ein vielfältiges Messgerät. Er kann als Oszilloskop, Signalgenerator, Bode Analysator, Logik Analysator, über ein externes Modul als LCR Meter, Software Defined Radio (SDR) oder als Vektor Network Analyser (VNA) eingesetzt werden.[11]



Abbildung 21: Red Pitaya STEMLab 125-14 v.1.1[11]

Der Red Pitaya basiert auf einem Xilinx SoC (Zynq 7010 FPGA mit einem Dual Core ARM Cortex-A9 Prozessor) und bietet zur einfachen Bedienung ein Online Web Interface,

auf welches über LAN oder WLAN zugegriffen werden kann.

Auch kann der Red Pitaya auch über verschiedene Wege ferngesteuert und für automatisierte Messung verwendet werden. Unterstützt werden LabVIEW, MATLAB, Python und Scilab.[11]

4.4 Messaufbau und verwendete Einstellungen

Für alle Messungen wird der gleiche Messaufbau verwendet.

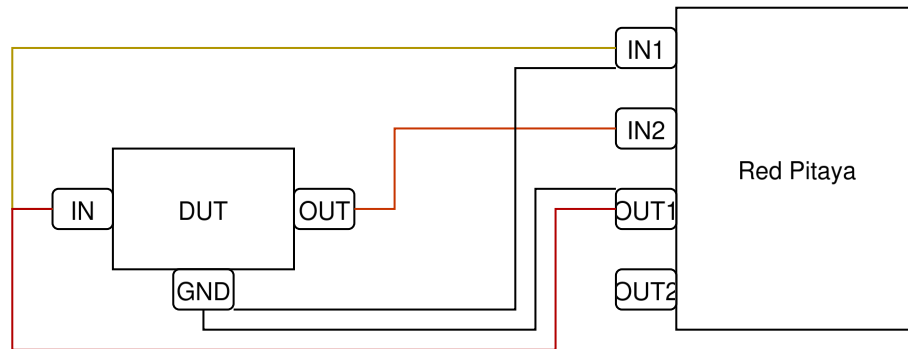


Abbildung 22: Gewählter Messaufbau

Die Abbildung 22 zeigt den verwendeten Messaufbau. Das Device Under Test (DUT) stellt einen der Filter des Biquad-Filters da. Der Eingang 1 (IN1) des Red Pitayas wird zusammen mit Ausgang 1 (OUT1) des internen Funktionsgenerators an den Eingang des DUT angeschlossen. Der Eingang 2 (IN2) des Red Pitayas wird an den Ausgang des DUT angeschlossen. Zusätzlich wird der Masse Anschluss von Eingang 1 und Ausgang 1 des Red Pitayas mit der Masse des DUT verbunden. Bevor mit dem Red Pitaya Messwerte aufgenommen werden, sollte dieser Kalibriert werden. Dafür muss der Ein- und Ausgang des DUT überbrückt werden.

Insgesamt werden 13 Messungen für vier Filtertypen durchgeführt. Für alle Messungen wird

- Funktion: Sinus
- Amplitude: 0.2V
- Offset: 0.0V

verwendet. Die anderen Einstellungen verändern sich für jede Messung und werden im Folgenden aufgelistet.

Für den Tiefpass, Hochpass, Bandsperre und Bandpass bei $f_0 = 159.15\text{Hz}$, Verstärkung $H = 1$ und Güte $Q = 1$ werden zusätzlich eingestellt:

- Startfrequenz: 50Hz
- Stopfrequenz: 1000Hz
- Messpunkte: 100

- Kondensatoren: $C = 1\mu F$
- Widerstand Verstärkung: $R_H = 1000\Omega$
- Widerstand Güte: $R_Q = 1000\Omega$

Bei der höheren Mittenfrequenz $f_0 = 1591.5Hz$ wird für Tiefpass, Hochpass, Bandsperre und Bandpass ,bei gleichbleibender Verstärkung $H = 1$ und Güte $Q = 1$, zusätzlich eingestellt:

- Startfrequenz: $50Hz$
- Stopfrequenz: $1000Hz$
- Messpunkte: 100
- Kondensatoren: $C = 100nF$
- Widerstand Verstärkung: $R_H = 1000\Omega$
- Widerstand Güte: $R_Q = 1000\Omega$

Bei der Erhöhung der Güte auf $Q = 10$, bei gleichbleibender Verstärkung von $H = 1$ und Mittenfrequenz $f_0 = 159.15Hz$ wird zusätzlich, zur Vermessungen des Tiefpasses, Hochpass, Bandsperre und Bandpass, eingestellt:

- Startfrequenz: $50Hz$
- Stopfrequenz: $1000Hz$
- Messpunkte: 100
- Kondensatoren: $C = 1\mu F$
- Widerstand Verstärkung = $R_H = 1000\Omega$
- Widerstand Güte: $R_Q = 10k\Omega$

Die letzte Messung erfolgt nur am Tiefpass bei einer Mittenfrequenz von $f_0 = 159.15Hz$, Verstärkung $H = 100$ und Güte $Q = 1$:

- Startfrequenz: $50Hz$
- Stopfrequenz: $1000Hz$
- Messpunkte: 100
- Kondensatoren: $C = 1\mu F$
- Widerstand Verstärkung: $R_H = 10\Omega$
- Widerstand Güte: $R_Q = 1000\Omega$

4.5 ASLK-Board

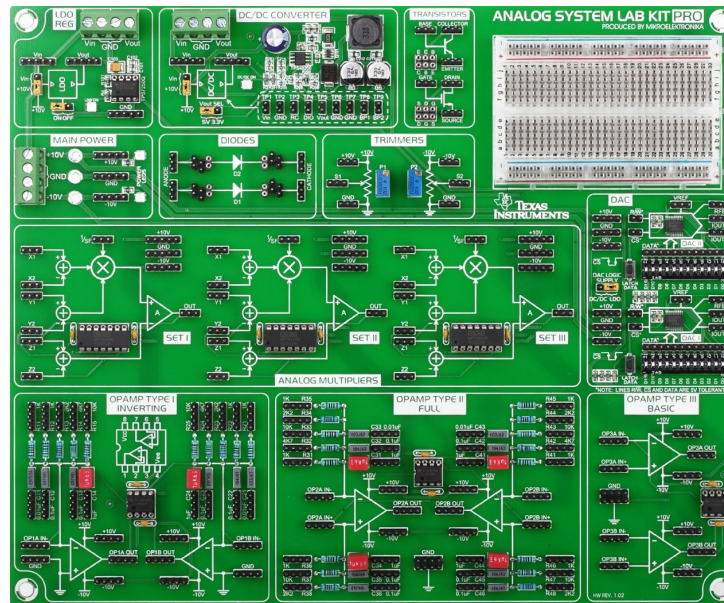


Abbildung 23: ASLK-Board

In Abbildung 23 ist das verwendete ASLK-Board zu sehen. Unten auf dem Board sind die Operationsverstärker zu sehen. Diese sind als Typ I, Typ II und Typ III gekennzeichnet und unterscheiden sich in ihrer Verschaltungsflexibilität. Die Typ-I-Op-Amps sind ausschließlich für invertierende Verstärkerschaltungen vorgesehen, Typ II kann sowohl in invertierender als auch in nichtinvertierender oder komplexeren Konfigurationen wie Addierer oder Differenzierer genutzt werden, während Typ III auf einfache Grundsaltungen beschränkt ist. Ergänzt wird das Board durch analoge Multiplikatoren, Digital-Analog-Wandler, Versorgungseinheiten (DC/DC-Wandler und LDO), Transistor- und Diodensockel, Trimmer, sowie ein Breadboard.

4.6 PCB

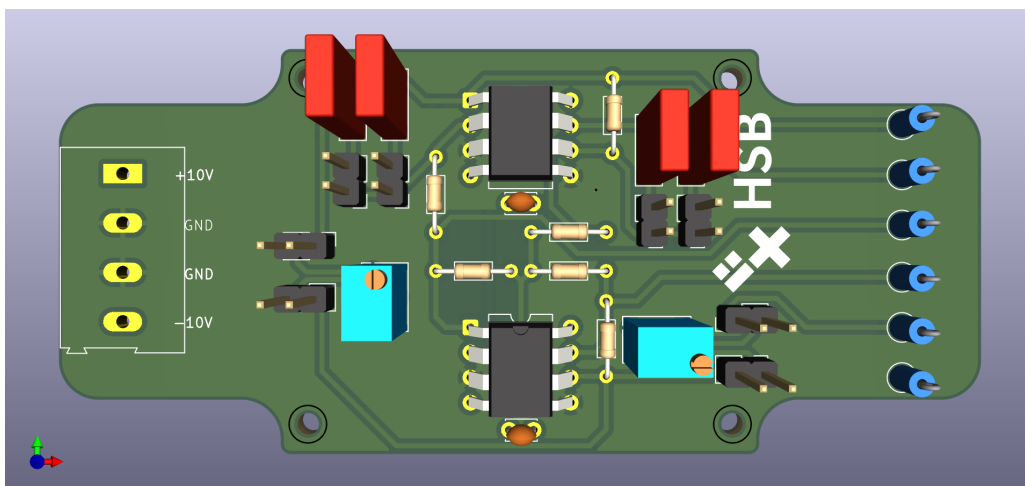


Abbildung 24: PCB-Layout

In Abbildung 24 ist das PCB-Layout zu sehen. Links auf dem Board wird die Betriebsspannung von $\pm 10\text{V}$ bereitgestellt. Diese wird über einen eigenen Layer an die TL082-Operationsverstärker geführt. Die OPs sind mittig auf dem Board platziert. Jeweils darunter befinden sich die Stützkondensatoren. Im oberen Bereich des Boards sind vier Kondensatoren angeordnet, die zur Einstellung der Grenzfrequenz des Filters verwendet werden. Hierfür kommen 100nF und $1\mu\text{F}$ zum Einsatz. Die Auswahl erfolgt über die darunterliegenden Stiftleisten. Weitere Stiftleisten weiter unten dienen der Einstellung von zwei $10\text{k}\Omega$ Potentiometern, mit denen einerseits die Güte Q und andererseits die Verstärkung des Frequenzgangs $|H(j\omega)|$ eingestellt werden.

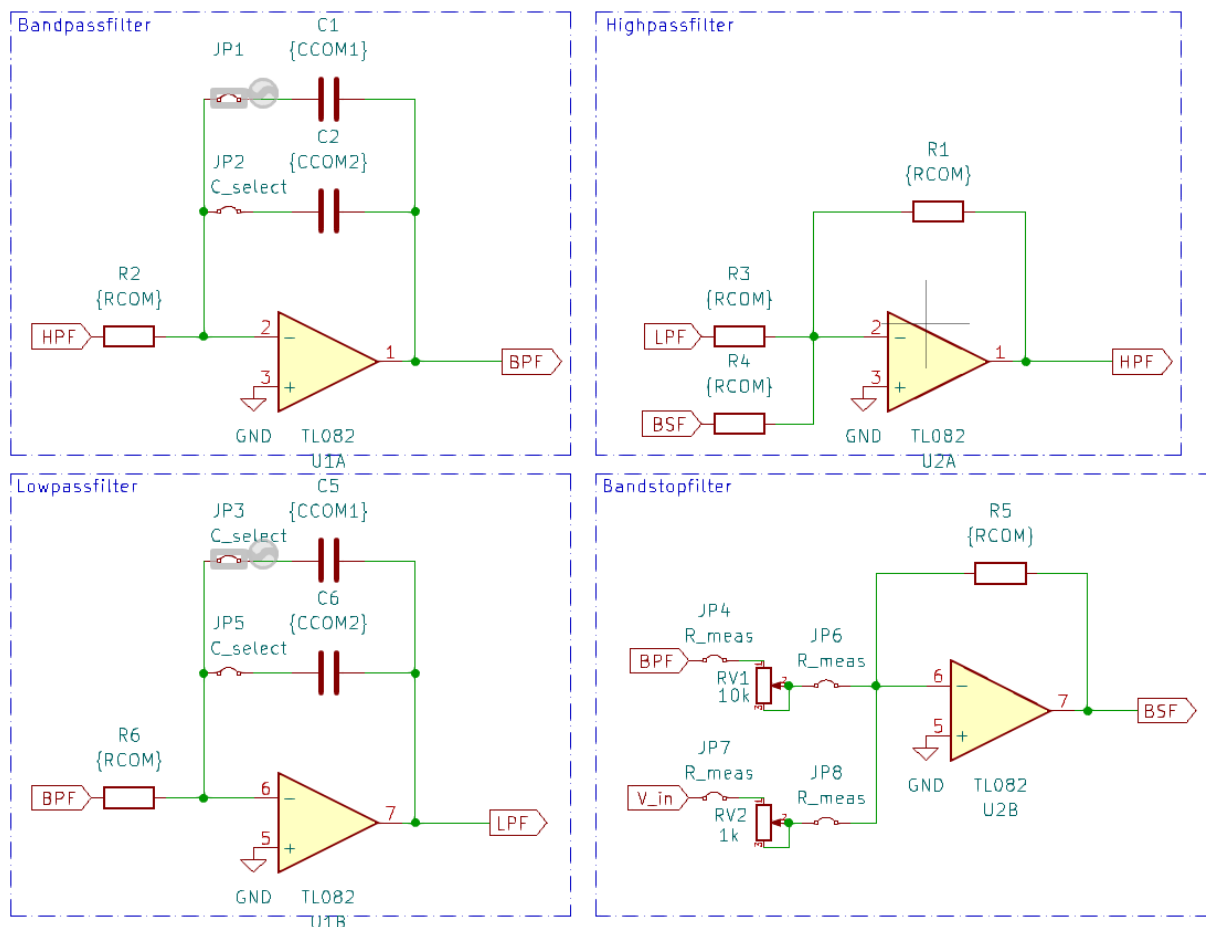


Abbildung 25: PCB-Schematic

Ein Ausschnitt aus dem Schematic der Leiterplatte ist in Abbildung 25 zu sehen. Zu sehen sind die verwendeten vier invertierenden Operationsverstärker. Davon sind zwei Integrerverstärker mit jeweils zwei Kondensatoren, die man über Jumper schalten kann und zwei weitere sind Summierverstärker.

Der erste Integrator speist das Signal des Hochpassfilters und gibt das Signal Bandpassgefiltert wieder. Der zweite Integrator speist dieses Signal und Tiefpassfiltert es. Das Bandpassgefilterte Signal fließt auch in den Bandsperrefilter, der als Summierverstärker verschaltet ist. Dieser addiert das Signal V_{in} dazu und gibt das Bandsperrefilterte Signal aus. Dieses und das Tiefpassgefilterte Signal werden im zweiten Summierverstärker addiert. Dieser gibt wiederum das Hochpassgefilterte Signal wieder.

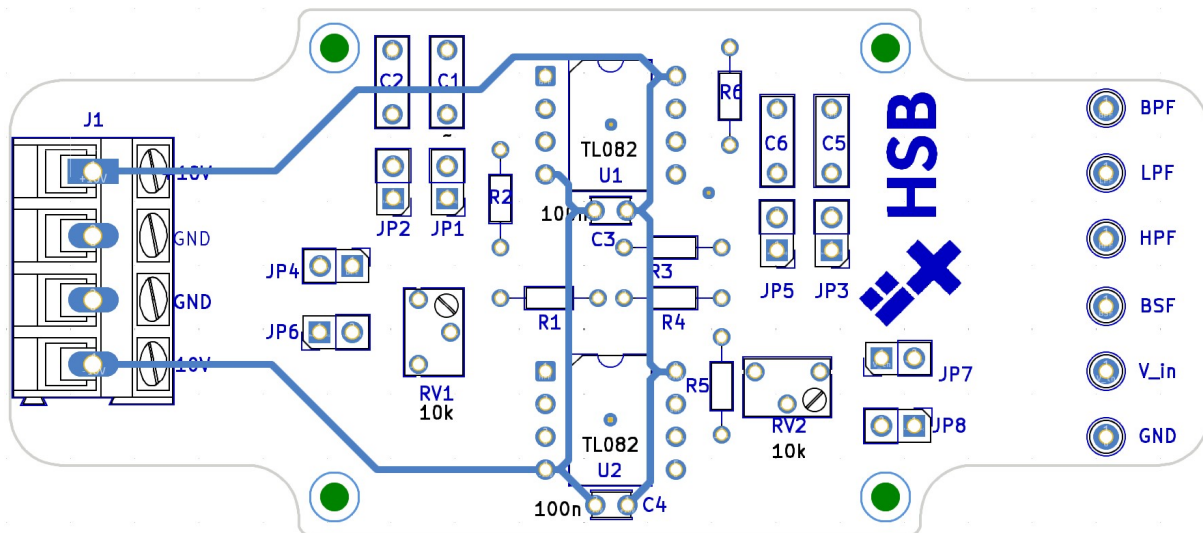


Abbildung 26: PCB-Layer 1

In Abbildung 26 ist der PCB-Layer dargestellt, welches der Spannungsversorgung der OPs liegt.

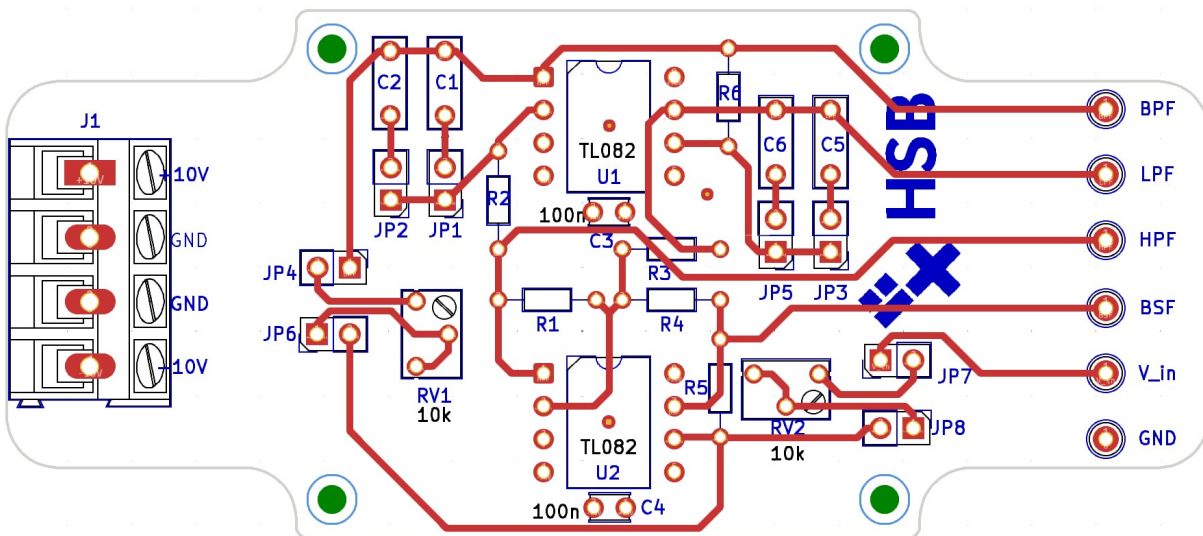


Abbildung 27: PCB-Layer 2

Der zweite Layer ist in Abbildung 27 dargestellt. Wie man sieht, ist es hauptsächlich für die Signalführung verantwortlich. Der Grund für die Trennung von Spannungsversorgung und Signalverläufen dient mehr der Übersichtlichkeit als dem Schutz der Signale. Bei den operierenden Frequenzen ist das noch nicht nötig. Weiterhin sieht man, dass die Bohrlöcher am Rand platziert worden sind, um Stabilität zu schaffen und keine Konflikte zu verursachen.

Des Weiteren wurden für mehr Komfortabilität und eine bessere Erreichbarkeit die seitlichen Ränder der Platine ausgespart, die aus dem Gehäuse herausragen. So ist es unkompliziert, die Stromversorgung zuzuführen und Filterausgänge und Messpunkte zu prüfen. Darüber hinaus ist das Schaltungsdesign komprimiert gehalten, um Signallaufzeiten so klein wie möglich zu halten und Verzerrungen somit entgegenzuwirken.

Im Designprozess wurden auch einige Verbesserungen gemacht. So waren zum Beispiel nicht benötigte Pins der Potentiometer unverbunden, wodurch sich elektrische Störungen einkoppeln könnten. Deshalb wurde der Pin jeweils mit einem anderen Pin des Potentiometers kurzgeschlossen.

Weiterhin wurden die Stützkondensatoren nahe an den OPs angebracht. Dadurch werden kurze Wege für den Strom gewährleistet und so die Versorgungsspannung direkt am Bauteil stabil gehalten. Sie filtern hochfrequente Störungen und kompensieren schnelle Stromspitzen, die beim Schalten im OP-Amp entstehen können. Wird der Kondensator zu weit entfernt platziert, kann die Leitung wie eine Antenne wirken oder störende Induktivität aufweisen, wodurch die Filterwirkung verloren geht und der OP-Amp anfällig für Störungen oder Instabilitäten wird.

Das PCB wurde als 2 Layer umgesetzt, da das für die Ansprüche völlig ausreichend ist und günstiger in der Fertigung.

Um auf der Platine zwischen zwei verschiedenen Kondensatoren wählen zu können, wurde eine Lösung mit Jumpfern und Stiftheisten umgesetzt. Alternativ hätten auch regelbare Kondensatoren, wie Dreh- oder Trimmkondensatoren, eingesetzt werden können, mit denen sich die Kapazität stufenlos einstellen lässt. Allerdings war die Umsetzung mit Jumpfern unproblematischer. Diese Trimmkondensatoren sind oft teurer und mechanisch empfindlicher.

4.7 Aufnahme der Messwerte über die Weboberfläche

Im ersten Schritt werden die Messwerte über die erwähnte Weboberfläche aufgenommen. Dafür wird der integrierte Bode Analyser verwendet.

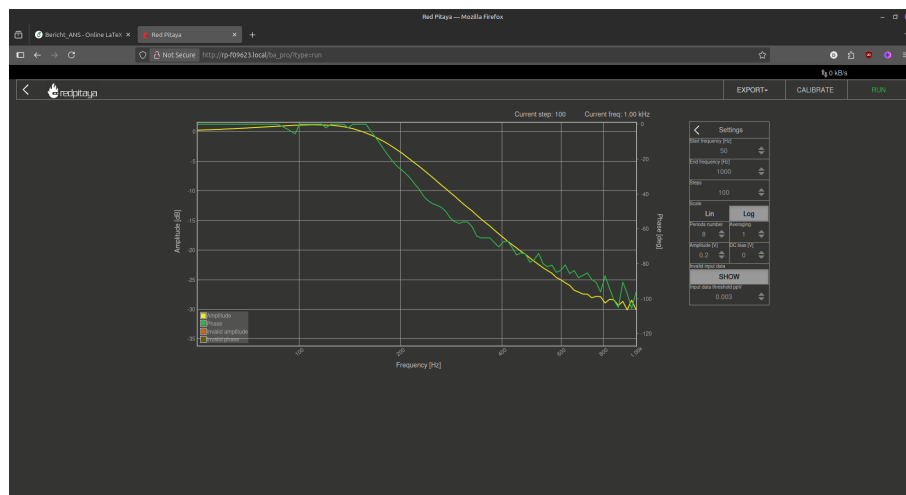


Abbildung 28: Bode Analyser in der Weboberfläche

In Abbildung 28 ist die Bedienoberfläche des Bode Analyser abgebildet. Unter *Settings* können Parameter für die Messung eingestellt werden. Eingestellt werden die Start- und Stopfrequenz, sowie die Anzahl der Messpunkte und die Darstellung in der Anzeige. Das Messergebnis kann entweder logarithmisch oder linear angezeigt werden.

Nach der Messung können die aufgenommenen Messwerte unter dem Punkt *Menu* entweder als .csv oder als Bild exportiert werden. Gewählt wird die Option als .csv, um die Messwerte später noch in Python weiter verarbeiten zu können.

4.8 Automatisierung der Messwertaufnahme und Auswertung

Der nächste Schritt ist die Fernsteuerung des Red Pitaya über ein Python Skript. Mit dem Python Skript soll eine automatische Aufnahme, Auswertung und Abspeichern der Bode Plots möglich sein. Das Pythonskript soll folgende Anforderungen erfüllen:

- Verbindung mit den Red Pitaya über die SCPI-Schnittstelle herstellen
- Konfigurationsmöglichkeit der Messung
- Durchführung der Messung und automatisches abspeichern der Messwerte als .csv
- Berechnung der Magnitude und Phase und anschließendes abspeichern der berechneten Werte
- Plotten berechneten Magnitude und Phase und Vergleich mit passenden Simulationsdaten

Um, den Red Pitaya mit dem Pythonskript fernsteuern zu können, muss der SCPI-Server des Red Pitayas gestartet werden. Zusätzlich muss in der Hochschule selbst noch die VPN verwendet werden, um eine Verbindung zum Red Pitaya herstellen zu können.

Ablauf der Messwertaufnahme und Auswertung

Das erstellte Messprogramm ist eine Erweiterung eines bereits vorhandenen Pythonskripts von Prof. Dr-Ing. M.Meiners. Unter anderem wird das Programm die Möglichkeit, Simulationsdaten aus KiCad oder LtSpice einzuladen, erweitert.

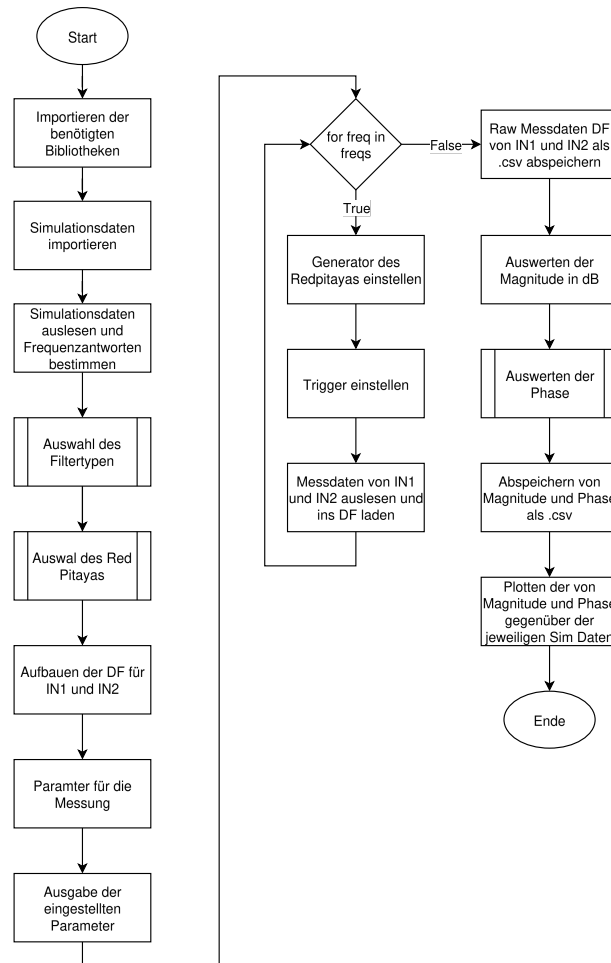


Abbildung 29: Flowchart des erstellten Messprogramms

In Abbildung 29 ist ein Flowchart des erstellten Messprogramms dargestellt. Im ersten Schritt werden die benötigten Bibliotheken eingeladen. Dazu gehören `ltpspice`, `matplotlib.pyplot`, `numpy`, `pandas`, `redpitaya_scp`, von Scipy wird das Modul `Signal` verwendet, sowie `datetime`, `time`, `sys` und `os`.

Importieren und auswählen der Simulationsdaten

Im nächsten Schritt werden die Simulationsdaten eingelesen. Diese werden vorher mithilfe des integrierten Simulators in KiCad erstellt. Aus den Simulationsdaten werden die Magnituden und Phase der einzelnen Filter ausgelesen. Über eine Eingabe können die Simulationsdaten für einen bestimmten Filter ausgewählt werden. Der Teilablaufplan dafür ist in 30 dargestellt.

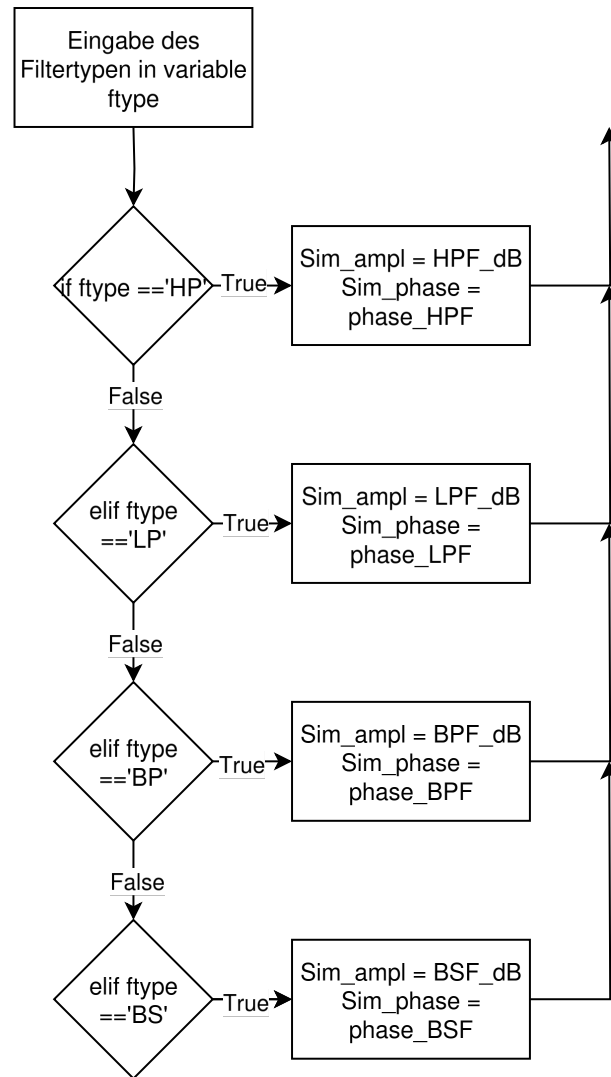


Abbildung 30: Flowchart des erstellten Messprogramms

Am Ende der Messung wird dann der vermessene Filter mit den zuvor ausgewählten Simulationsdaten verglichen. Zur Auswahl stehen Hochpass, Lowpass, Bandpass und Bandsperre, welche jeweils über HP, LP, BP oder BS ausgewählt werden können.

Messparameter und verbinden mit dem Red Pitaya

Im nächsten Schritt erfolgt dann die Auswahl des Arbeitsplatzes. Jedem Arbeitsplatz ist dabei eine andere IP-Adresse zugewiesen. Diese Methode ermöglicht einen schnellen und einfachen Wechsel des verwendeten Arbeitsplatzes und damit Red Pitaya.

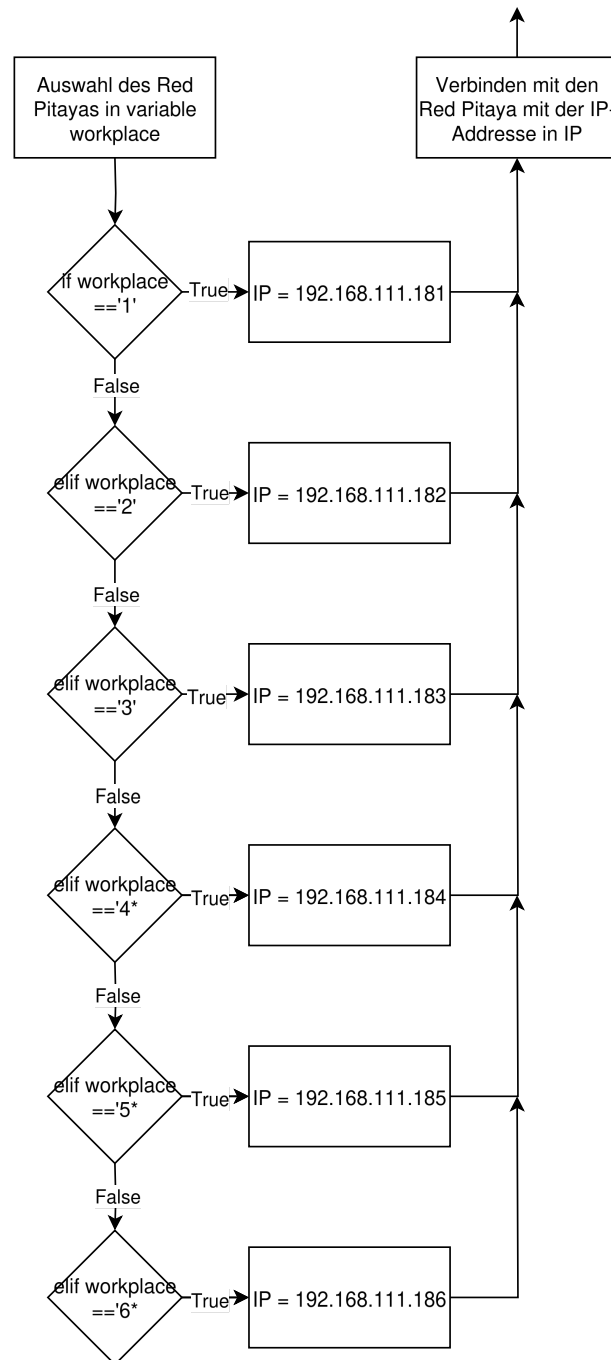


Abbildung 31: Teilauslaufplan zur Auswahl des Red Pitaya

Nach der Auswahl des Red Pitayas werden zwei Data Frames mit Pandas für den Eingang IN1 und den Eingang IN2 aufgebaut. In diese Data Frames werden später die aufgenommen rohen Messwerte während der Messung abgespeichert.

Über mehrere Parameter kann der integrierte Funktionsgenerator des Red Pitaya eingestellt werden. Die eingestellte Parameter bestimmen auch die Messung. Mit dem Parameter *func* kann die Form, wie z.B. Sinus oder Kosinus, der ausgegebenen Spannung eingestellt werden. Der Parameter *ampl* stellt die Amplitude ein und über *offset* der Wechselspannung noch eine gewünschte Gleichspannung hinzugefügt werden, welche den Pegel anhebt oder senkt. Zum Schluss kann über den Parameter *freq* der gewünschte Frequenzbereich und die Anzahl der Frequenzpunkte eingestellt werden. Hierfür wird von numpy die Funk-

tion linspace verwendet. Mit linspace werden zwischen der Start- und Stopfrequenz eine bestimmte Anzahl an Frequenzpunkte in gleichbleibenden Abstand gesetzt. Die eingestellte Parameter werden auch über die Konsole ausgegeben.

Der Messvorgang findet innerhalb der for-Schleife statt. Für jeden vorher in *freq* gewählten Frequenzpunkt wird der Funktionsgenerator passend eingestellt und anschließend die Messwerte an IN1 und IN2 im jeweiligen Data Frame abgespeichert. Die Messwerte werden anschließend auch als .csv abgespeichert.

Ermittlung der Frequenzantwort

Die Frequenzantwort umfasst die Amplituden- und Phasenantwort des Filters. Um die Amplitudenantwort zu ermitteln, wird folgende Gleichung verwendet.

$$MAG_dB = 20 \cdot \log_{10}\left(\text{abs}\left(\frac{V_{IN2}}{V_{IN1}}\right)\right) \quad (40)$$

Für die Amplitudenantwort wird die Ausgangsspannung V_{IN2} durch die Eingangsspannung V_{IN1} geteilt. Es folgt dann noch die Umrechnung in dB. So wird die Amplitudenantwort des Filters logarithmisch dargestellt.

Zur Ermittlung der Phasenantwort wird eine for-Schleife verwendet. Das Ziel dabei ist es, die Phase an jeden Messpunkt zu ermitteln.

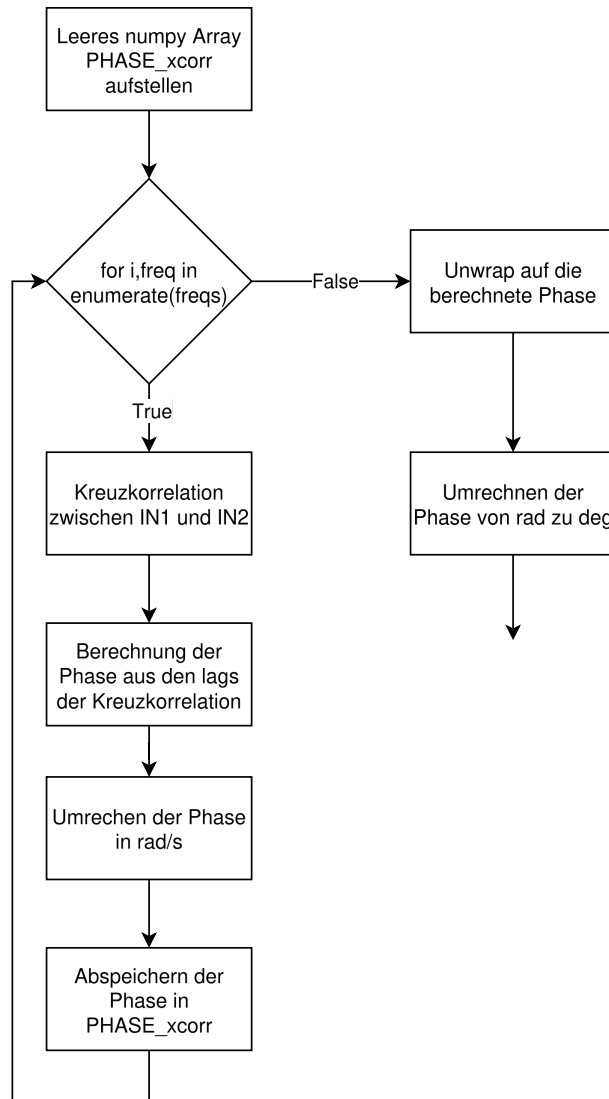


Abbildung 32: Teilablaufplan zur Bestimmung der Phasenantwort

Die Abbildung 32 zeigt den Ablaufplan der for-Schleife zur Bestimmung der Phasenantwort. Bevor die Phase ermittelt wird, wird ein leeres Numpy Array mit einer Länge, welche gleich der Anzahl der Frequenzpunkte entspricht, erstellt. In diesem Numpy Array werden die Phasen an den einzelnen Frequenzpunkten abgespeichert.

Um die Phase zu ermitteln, wird eine Kreuzkorrelation zwischen der Spannung V_{IN1} und der Spannung am Ausgang V_{IN2} durchgeführt.

$$r_{IN1IN2}[\ell] = \lim_{K \rightarrow \infty} \frac{1}{2K+1} \sum_{n=-K}^K IN1[n+\ell] \cdot IN2^*[n] \quad (41)$$

Die Kreuzkorrelation berechnet die Ähnlichkeit zwischen dem Eingangssignal V_{IN1} und dem Ausgangssignal V_{IN2} . Die Kreuzkorrelation gibt einen Lag wieder, welcher gleich der Verschiebung zwischen den beiden Signalen ist.

Mit der Verschiebung kann dann die Phase berechnet werden. Die Kreuzkorrelation und die anschließende Ermittlung der Phase werden mit zwei Funktionen von Scipy.Signal durchgeführt. Für die Kreuzkorrelation wird die Funktion *correlate* und für die anschließende Ermittlung der Phase aus der Verschiebung *correlation_lags* verwendet.

Die Phase wird dann in im zuvor erstellten leeren Numpy Array abgespeichert. Die Einheit der Phase ist rad/s . Zu besseren Darstellung soll die Phase nach der for-Schleife in deg umgewandelt werden. Verwendet wird dafür die Funktion $rad2deg$ von Numpy. Zuvor muss aber noch Unwrap verwendet werden.

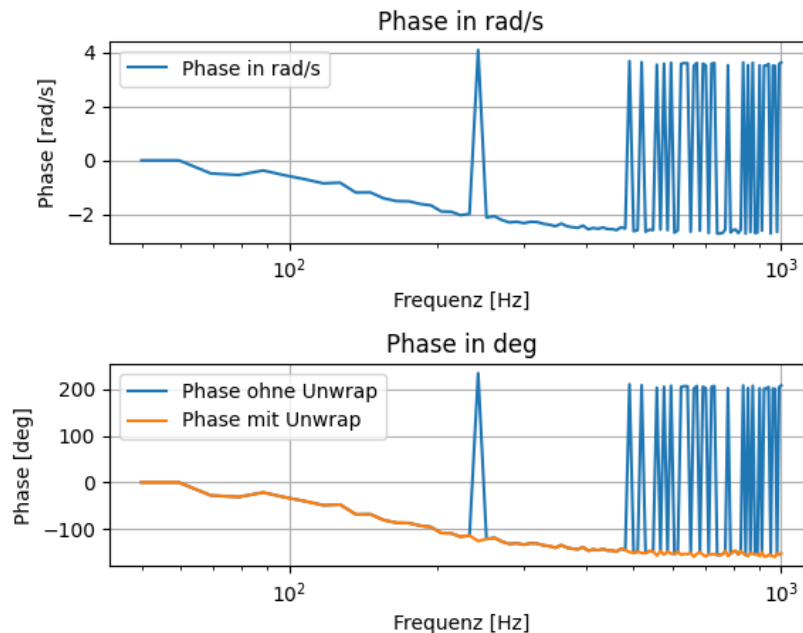


Abbildung 33: Vergleich zwischen der Phasenantwort mit und ohne Unwrap

Mit Unwrap werden plötzliche Sprünge, auch Diskontinuität genannt, im Phasenverlauf beseitigt. Diese Sprünge werden auch Phase Wrapping genannt. Phase Wrapping ist kein physikalischer Effekt, sondern entsteht bei der mathematischen Bestimmung der Phase. Phase Wrapping tritt auf, wenn die Phase den Wertebereich der verwendeten Funktion verlässt oder nach einer Fast Fourier Transformation. In Gleichung 42 ist der Arctan dargestellt. $s(t)$ ist dabei ein realwertiges Signal.

$$\varphi(t) = \arctan\left(\frac{\text{Im}[s(t)]}{\text{Re}[s(t)]}\right), -\pi \leq \varphi \leq +\pi \quad (42)$$

Mit dem Arctan kann die Phase in einem Bereich von 0 und 2π oder $-\pi$ und π eindeutig festgelegt werden. Wenn die Phase $-\pi$ oder π überschreitet, springt diese zurück in definiert Intervall. Die Phase springt dann von $+\pi$ zu $-\pi$ oder umgekehrt. Um die Diskontinuitäten zu verhindern wird die Funktion *unwrap* von Numpy verwendet. Die Funktion überprüft den Unterschied zwischen zwei benachbarten Werten. Ist dieser größer als π rechnet die Funktion ein vielfaches (2π) drauf gerechnet. [12] Damit werden die Diskontinuität beseitigt und die Phasenantwort wird so geglättet. Die Abbildung 33 zeigt den Vergleich zwischen der Phase ohne die Unwrap-Funktion und der Phase mit der Unwrap-Funktion

Die aufgenommen Amplituden- und Phasenantwort werden einzeln ebenfalls als .csv abgespeichert und auch im Messprogramm mit den passenden Simulationsdaten geplottet.

4.9 Probleme bei der Aufnahme und Auswertung der Messwerte

Während der Entwicklung des Messprogramms sind wiederholt unterschiedliche Probleme bei Aufnahme und der Auswertung der Messung aufgetreten.

Die Probleme fangen bei der Aufnahme der Messwerte über die Weboberfläche an. Einige Male lies sich der Bode Analyser des Red Pitayas nicht einstellen. Durch das herstellen einer neuen Verbindung lies sich das Problem lösen. Nach dem Update auf eine neue Firmware trat das Problem nicht mehr auf.

Die nächsten Probleme betrifft die Aufnahme und Auswertung der Messwerte über das erstellte Messskript. Das erste da aufgetretende Problem hängt mit der Anzahl der gewählten Messpunkte zusammen. Sobald mehr als 100 Messpunkte gewählt werden, besteht die Gefahr, dass die Verbindung zum Red Pitaya mitten in der Messung verloren geht. Alle bis dahin aufgenommen Messdaten sind verloren gegangen und die Messung musste neu gestartet werden. Ein Grund für die Verbindungsproblem konnte nicht auswendig gemacht werden. Das Problem ist nach dem Wechsel auf einen anderen Red Pitaya nicht mehr aufgetreten.

Dafür kommt aber ein neues Problem, welches ebenfalls mit den 100 Messpunkten zusammenhängt. Ebenfalls ab 100 gewählten Messpunkten, gibt das Pandas Modul bei der Berechnung der Phase eine Fehlermeldung aus, dass die Data Frames von IN1 und IN2 stark fragmentiert seien. Die Berechnung der Phase erfolgt trotzdem und die fragmentierten Data Frames scheinen keine Auswirkung auf die Phase zu haben.

Ein weiteres Problem hängt ebenfalls mit der Bestimmung der Phase zusammen. Vom Red Pitaya selbst wird standardmäßig nichts gegen Phase Wrapping unternommen. Weder bei der Aufnahme über die Weboberfläche oder bei der Aufnahme über das Python Skript. Aus diesem Grund muss das unwrap eingebunden werden, bevor die Phase von *rad/s* zu *deg* umgewandelt wird.

Ebenfalls ist die Aufnahme über das erstellte Messskript sehr zeitaufwendig, da zwischen jeder Messung eine Sekunde gewartet wird. Der Grund dafür ist, dass der Trigger und der Funktionsgenerator für jeden Messpunkt neu eingestellt wird. Eine Verkürzung dieser Zeit, führt zu einer deutlichen Verschlechterung der Messergebnisse.

Am Tag, wo das PCB vermessen werden sollte, ist ein weiteres Problem bei der Aufnahme über das Messskript aufgetreten. Die aufgenommen Amplitude- und die Phasenantworten wiesen große Schwankungen auf, welche nicht erklärt werden konnten und auch kein Grund dafür im Messskript selbst gefunden wurde. Die Schwankungen traten beim PCB, als auch beim ASLK PRO Board auf. Die Schwankungen traten aber nicht bei der Aufnahme über die Weboberfläche auf, weshalb am Ende alle Messwerte für das PCB und das ASLK PRO Board über die Weboberfläche des Red Pitayas aufgenommen.

5 Auswertung der Ergebnisse

Im folgenden Kapitel werden die Messergebnisse der verschiedenen Filterschaltungen analysiert und ausgewertet. Für jeden der vier Filtertypen – Hochpass, Tiefpass, Bandpass und Bandsperre – wurden Messungen durchgeführt und die jeweiligen Bode-Diagramme aufgenommen. Bei Analyse der Graphen werden klare Unterschiede zwischen Simulation (blau), PCB (grün) und ASLK-PRO-Board (orange) deutlich.

5.1 Tiefpass

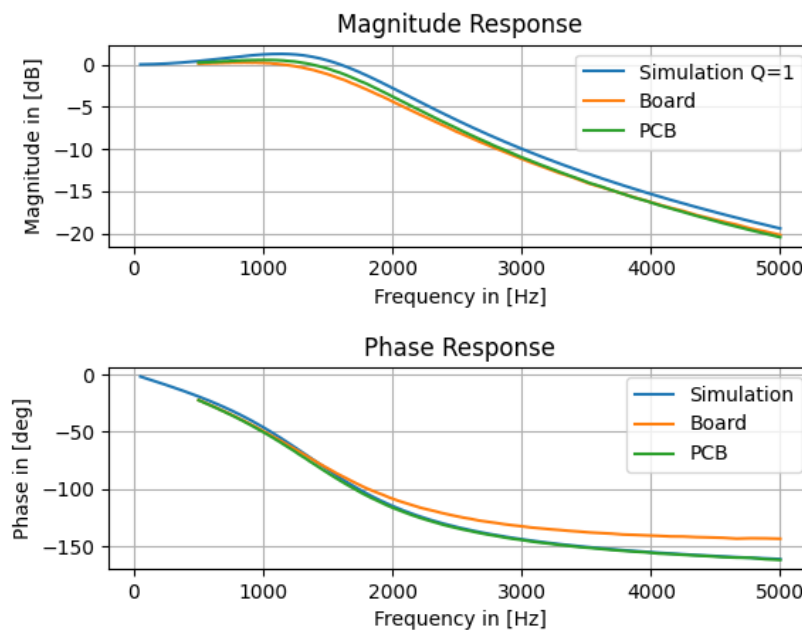
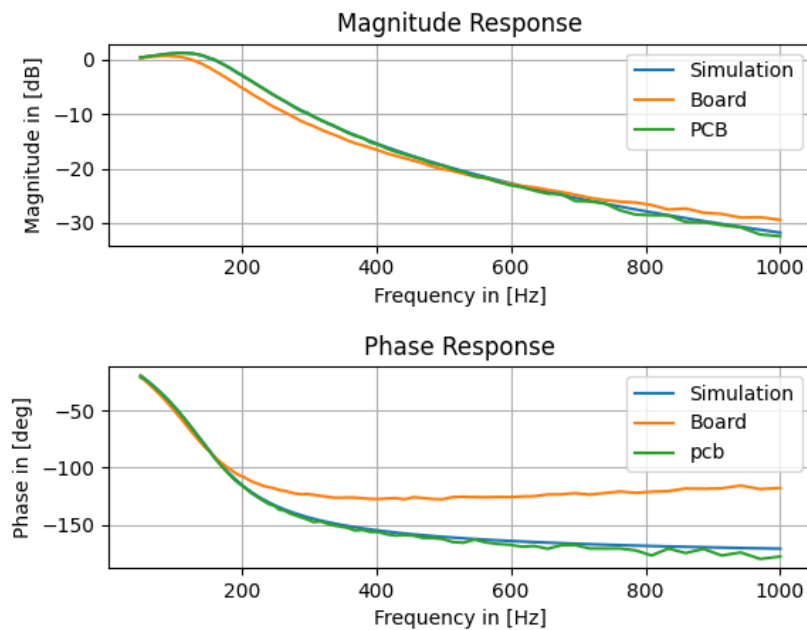


Abbildung 34: Tiefpass mit $Q=1$, $H=1$, $C=100\text{nF}$

Der Frequenzgang zeigt einen klassischen Tiefpass zweiter Ordnung, bei dem die Grenzfrequenz aus der Formel $f_0 = \frac{1}{2\pi RC}$ resultiert (Gleichung 25) und bei den verwendeten Werten bei ca. 1,59 kHz liegt. Der Amplitudenverlauf fällt oberhalb dieser Frequenz erwartungskonform mit -40 dB/Dekade ab. Bei 5 kHz liegen die Amplituden bei etwa -20 dB

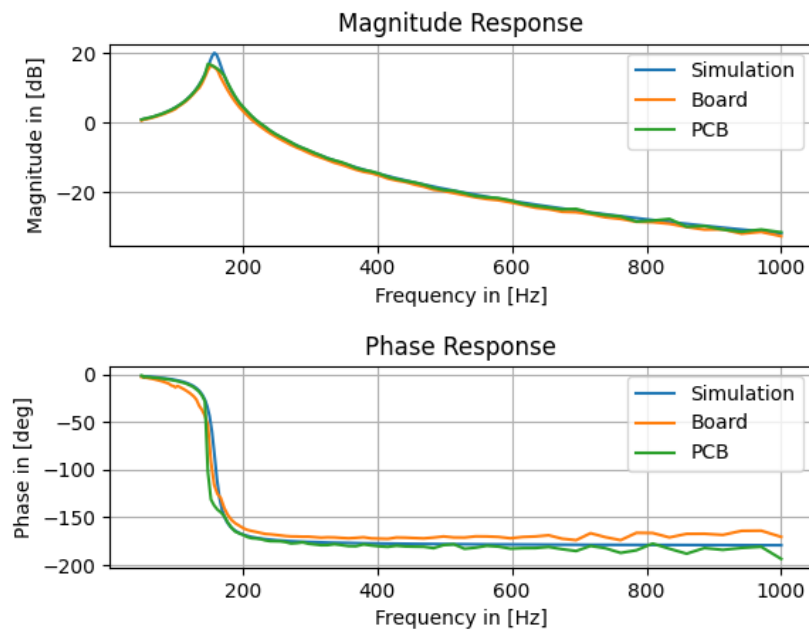
Der Graph des ASLK-Boards und des PCBs sind relativ deckungsgleich, weichen allerdings um einen konstanten Faktor von der Simulation ab. So ist der Simulationsgraph ab etwa 1 kHz um ca. 1 dB konstant über ASLK-Board und PCB.

Im Phasengang verlaufen Simulation und PCB fast identisch und zeigen einen weichen Übergang von 0° auf -160° , wobei das ASLK-Board zwar dieselbe Tendenz besitzt, allerdings mit ca. -140° deutlich höher ausfällt für 5 kHz. Dies deutet auf Streu- und Kontaktkapazitäten im Versuchsaufbau, die das Verhalten real verändern.

Abbildung 35: Tiefpass mit $Q=1$, $H=1$, $C=1\mu\text{F}$

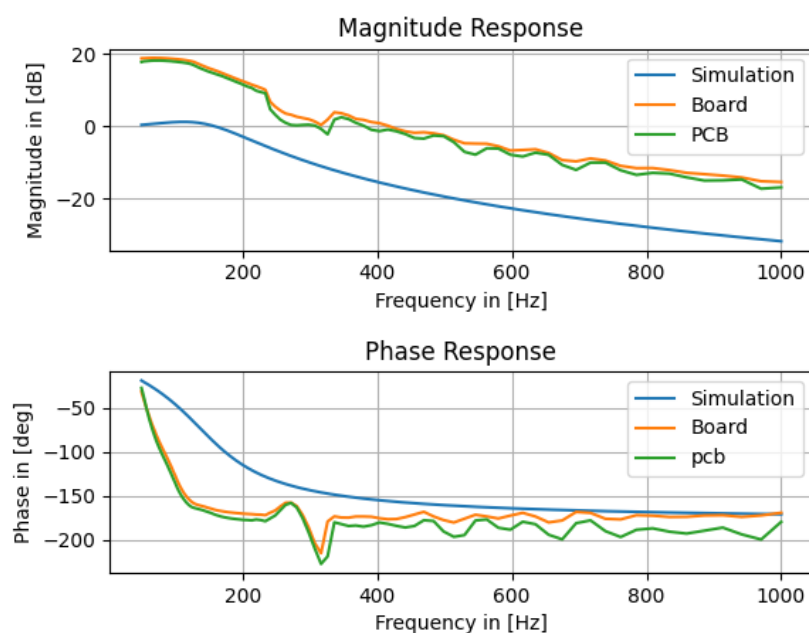
Mit der Anhebung der Kapazität verschiebt sich die theoretische Grenzfrequenz auf rund 159 Hz. Auffällig und im Kontrast zur ersten Variante ist jedoch, dass das ASLK-Board im Phasengang eine starke Abweichung von Simulation und PCB aufweist. Nach der Grenzfrequenz wird die negative Steigung betragsmäßig zunehmend kleiner bis die Steigung positiv wird. Während PCB und Simulation bei 1 kHz auf die theoretisch erwarteten -180° Phasenverschiebung kommen, liegt das ASLK-Board bei ca. -120° . Diese Differenz ist ein klarer Hinweis auf parasitäre Effekte sowie Dämpfungen im Kontaktierungsaufbau des ASLK-Boards und zeigt den deutlichen Vorteil einer gefertigten Leiterplatte gegenüber eines Steckboards.

Betrachtet man den Frequenzgang liegen PCB und Simulation wieder deutlich beieinander, während der Graph des ASLK-Boards zunächst stärker fällt und bei ca. 500 Hz die anderen beiden Graphen schneidet. Bei 1 kHz liegt das ASLK-Board bei etwa -30 dB. PCB und Simulation haben eine um 3 dB stärkere Dämpfung.

Abbildung 36: Tiefpass mit $Q=10$, $H=1$, $C=1\mu\text{F}$

Der hohe Gütefaktor von $Q = 10$ bewirkt eine ausgeprägte Resonanzüberhöhung im Amplitudengang, wie es die Theorie für Pole nahe der imaginären Achse vorhersagt. ASLK-Board und PCB zeigen ein fast deckungsgleiches Maximum, während die Simulation erneut eine höhere Resonanz aufweist.

Beim Phasengang tritt bei Simulation und PCB erwartungsgemäß ein abrupter, sehr steiler Sprung im Bereich der Resonanz auf. Beim ASLK-Board hingegen verläuft der Phasensprung flacher und beginnt mit steigender Frequenz größer zu werden.

Abbildung 37: Tiefpass mit $Q=1$, $H=100$, $C=1\mu\text{F}$

Durch die Erhöhung des Verstärkungsfaktors nehmen sowohl der PCB als auch der ASLK Graph seltsame Verläufe an, die eine große Ähnlichkeit aufweisen.

Auch im Phasendiagramm nehmen PCB und ASLK-Board merkwürdige verläufe an. Bis auf die typische Phasenverschiebung zu Schluss sind beide nahezu deckungsgleich.

Es ist möglich, dass die Verläufe ausgelöst werden durch die Begrenzungen der Operationsverstärker.

5.2 Hochpass

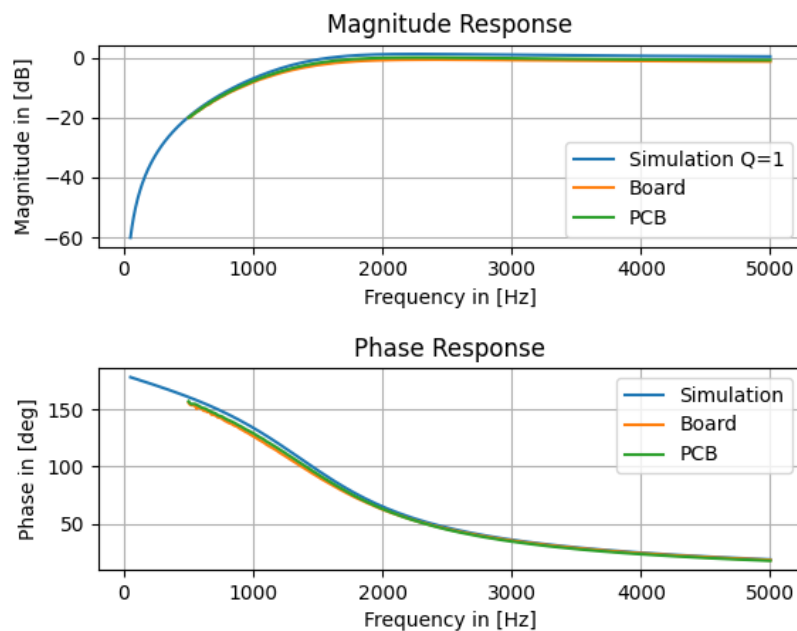


Abbildung 38: Hochpass mit $Q=1, H=1, C=100\text{nF}$

Der Hochpasscharakter zeigt sich in einer starken Dämpfung tiefer Frequenzen. Allerdings fehlt der charakteristische Verlauf von $+40\text{ dB/Dekade}$ (zweite Ordnung). Durch den 100 nF Kondensator, liegt die Grenzfrequenz hier wieder bei $1,59\text{ kHz}$.

Amplituden- und Phasengänge sind zwischen Simulation, ASLK-Board und PCB fast identisch. Beim Phasengang erreicht die Simulation nahezu den typischen Verlauf von -180° bis 0° . ASLK-Board und PCB haben zwar einen stark ähnlichen Verlauf beginnen aber, wie beim Frequenzgang, erst bei ca. 500 Hz , was der Messwertaufnahme geschuldet ist.

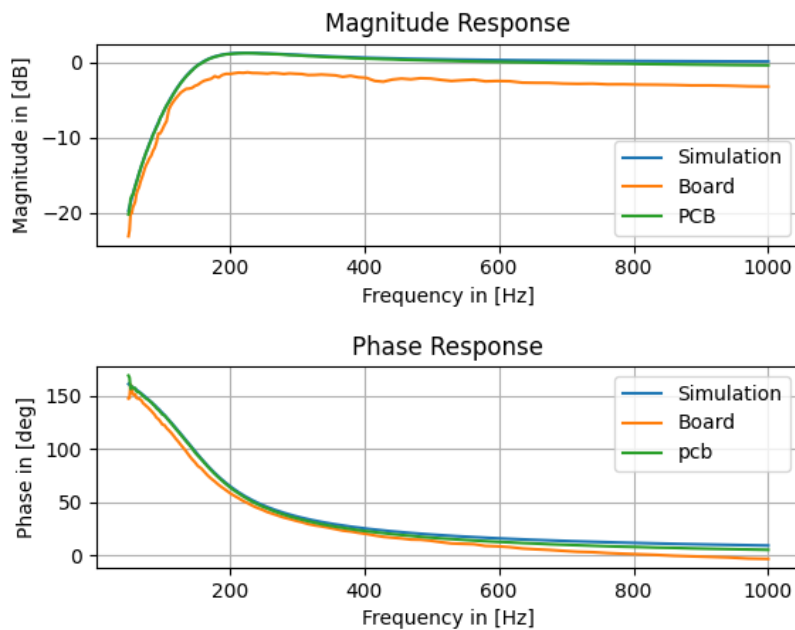


Abbildung 39: Hochpass mit $Q=1$, $H=1$, $C=1\mu\text{F}$

Die Frequenzverschiebung durch vergrößerte Kapazität setzt die Grenzfrequenz auf etwa 159 Hz herab. Betrachten wir zunächst den Frequenzgang. Während der Graph des PCBs die steilen Übergänge und Plateaus der Simulation exakt reproduziert, weicht das ASLK-Board durch eine fast konstant stärkere Dämpfung (rund 3 dB) ab. Der Phasengang wiederum gleicht erneut dem eines Hochpasses. Alle Graphen liegen nahe beieinander, starten jedoch erst bei -160° und verlaufen nach 0°

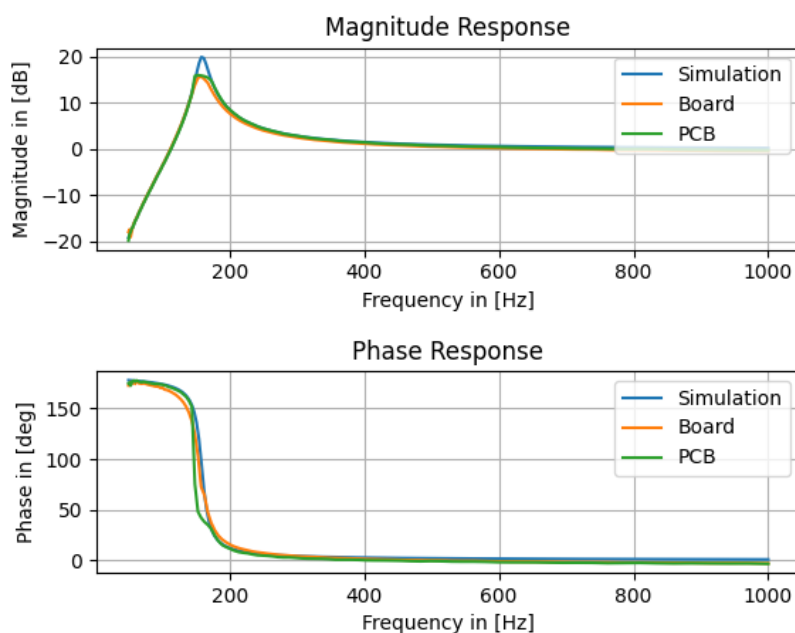


Abbildung 40: Hochpass mit $Q=10$, $H=1$, $C=1\mu\text{F}$

Bei hoher Güte tritt im Hochpass eine starke Resonanz kurz hinter der Grenzfrequenz

auf. Diesen Effekt zeigen wieder ASLK-Board und PCB in fast perfektem Gleichlauf. Die Simulation erreicht eine deutlich höhere Resonanz.

Der Phasengang zeigt für alle Graphen den charakteristischen Verlauf von 180° bis 0° .

Die Messergebnisse des Hoch- und Tiefpasses zeigen, dass die PCB und Simulations-Graphen die theoriebezogenen Erwartungen (z. B. Formel $f_0 = \frac{1}{2\pi RC}$, Flankensteilheit -40 dB/Dekade , Pol-Nullstellen-Konstellation) größtenteils abbilden. Markante Abweichungen betrafen den Tiefpass für eine Verstärkung von $H = 100$ und die Steigung des Hochpasses.

Das ASLK-Board hingegen zeigt klar die Grenzen steckbarer Laboraufbauten. Insbesondere Phase und Dämpfung weichen sichtbar ab, wobei Toleranzen, Kontaktwiderstände und parasitäre Admittanzen systematisch für reduzierte Steilheiten, verschobene Grenzfrequenzen und Phasen verantwortlich sind. In mehreren Fällen (z. B. beim zweiten Tiefpass, 1 F) wird das Idealbild auf dem ASLK-Board deutlich verfehlt, die Phase erreicht ihr Zielniveau nicht mehr. Insgesamt bestätigen die Vergleiche, dass die mathematischen Modelle auch praktisch Aussagekraft besitzen, sofern die physikalischen Umsetzungen präzise genug sind.

5.3 Bandpass

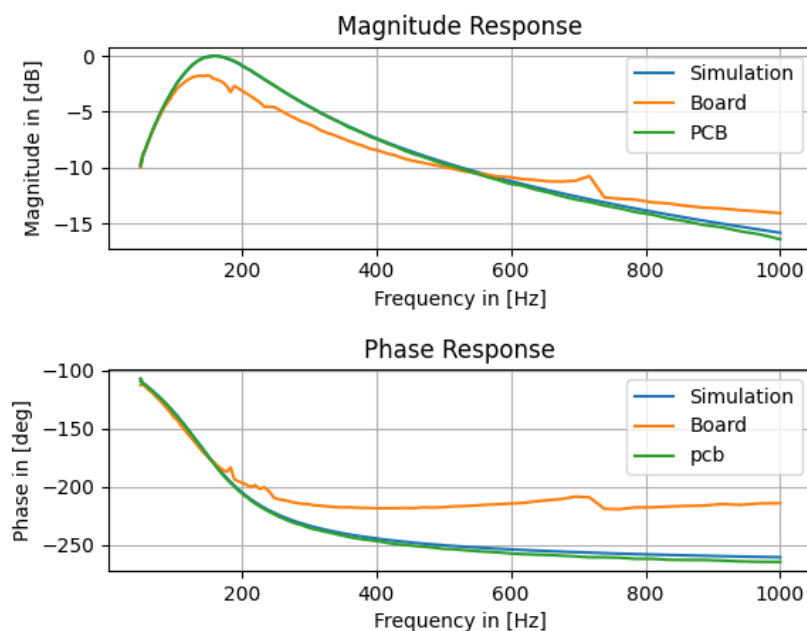


Abbildung 41: Frequenzantwort eines Bandpasses mit $Q=1$, $H=1$, $C=1\mu\text{F}$

Die Abbildung 41 zeigt die Amplituden (oberer Plot) und Phasenantwort (unterer Plot) für den Bandpassfilter mit einer Güte $Q = 1$ und einer Verstärkung $H = 1$. Für den Kondensator wird ein Wert von $1\mu\text{F}$ gewählt. Die blaue Kurve stellt die Werte für den simulierten Bandpass, die orange Kurven stellt den Bandpass auf dem PCB und die grüne Kurve den Bandpass auf dem ASLK PRO Board dar.

Wird die Gleichung 17 nach f_0 umgestellt, kann so die Mittenfrequenz des Bandpasses

bestimmt werden.

$$f_0 = \frac{1}{2\pi \cdot R \cdot C} = \frac{1}{2\pi \cdot 1000\Omega \cdot 1 \cdot 10^{-6}F} = 159.15Hz$$

Sowohl der simulierte Bandpass, als auch der Bandpass auf dem PCB weisen diese Mittenfrequenz auf. Der Bandpass auf dem ASLK PRO Board weist nicht die gewünschte Mittenfrequenz auf. Auf dem Board weist der Bandpass eine niedrigere Mittenfrequenz von ca. $f_{0Board} = 141Hz$ auf. Das entspricht einer Abweichung von $\Delta f_0 = 159.15Hz - 141Hz = 18.15Hz$.

Mit Gleichung 21 kann die Bandbreite für den Bandpass bestimmt werden.

$$BW = \frac{f_0}{Q} = \frac{159.15Hz}{1} = 159.15Hz$$

Demnach hat der simulierte Bandpass eine Bandbreite von $BW = 159.15Hz$. Der Bandpass auf dem PCB weist ebenfalls die berechnete Bandbreite von $BW = 159.15Hz$ auf. Die Bandbreite des Bandpasses auf dem ASLK Pro Board ist mit $BW = f_{g2} - f_{g1} = 229.3Hz - 84.7Hz = 144.6Hz$ etwas kleiner.

Ebenfalls ist die Amplitude des Bandpasses auf dem Board stärker gedämpft. Der Bandpass auf dem ASLK PRO Board erreicht bei seiner Mittenfrequenz von $f_0 = 141Hz$ eine Passbanddämpfung von nur 1.65dB. Der simulierte Bandpass und der Bandpass auf dem PCB erreichen beide bei $f_0 = 159.15Hz$ eine Passdämpfung von fast 0dB

Im weiteren Verlauf weist die Amplitudenantwort des simulierten Bandpasses und die des Bandpasses auf dem PCB minimale Abweichungen auf. Am Ende erreicht der Bandpass auf dem PCB eine minimal bessere Sperrbanddämpfung von 16,48dB, während der Bandpass in der Simulation nur eine minimale Sperrbanddämpfung von 15.7dB erreicht. Der Bandpass auf dem Board schneidet nicht so gut ab, wie die anderen beiden. Dieser erreicht nur eine minimale Sperrbanddämpfung von 13.93dB. Im Vergleich zum Bandpass auf dem PCB sind das $\Delta dB = 16.48dB - 13.93dB = 2.55dB$ Unterschied. Das bedeutet, dass der Bandpass auf dem Board im Sperrband eine doppelt so hohe Leistung passieren lässt, wie der Bandpass auf dem PCB.

Beim Verlauf der Phase gibt es am Anfang bei $f = 50Hz$ keinen Unterschied zwischen dem simulierten Bandpass, dem Bandpass auf dem PCB und dem Bandpass auf dem Board. Alle drei Filter starten bei $\varphi = -108$.

Im weiteren Verlauf machen sich dann aber doch Unterschiede bemerkbar. Ab $f = 157Hz$ driftet die Phase des Bandpasses auf dem Board immer weiter vom simulierten Bandpass weg und weist im weiteren Verlauf immer wieder kleinere Sprünge auf. Der Bandpass auf dem PCB bleibt nah an den simulierten Werten.

Am Ende landen die Phasen des Bandpasses aus der Simulation und des Bandpasses auf dem PCB bei $\varphi = -260$, bzw. $\varphi = -265^\circ$. Die Phase vom Bandpass auf dem ASLK PRO Board landet bei $\varphi = -213$.

Eine mögliche Erklärung, warum sich die Leistung des Bandpasses auf dem PCB und auf dem ASLK PRO Board unterscheiden, könnte die vielen Kabel auf dem Board sein. Diese könnten die Leistung des Bandpasses negativ beeinflussen.

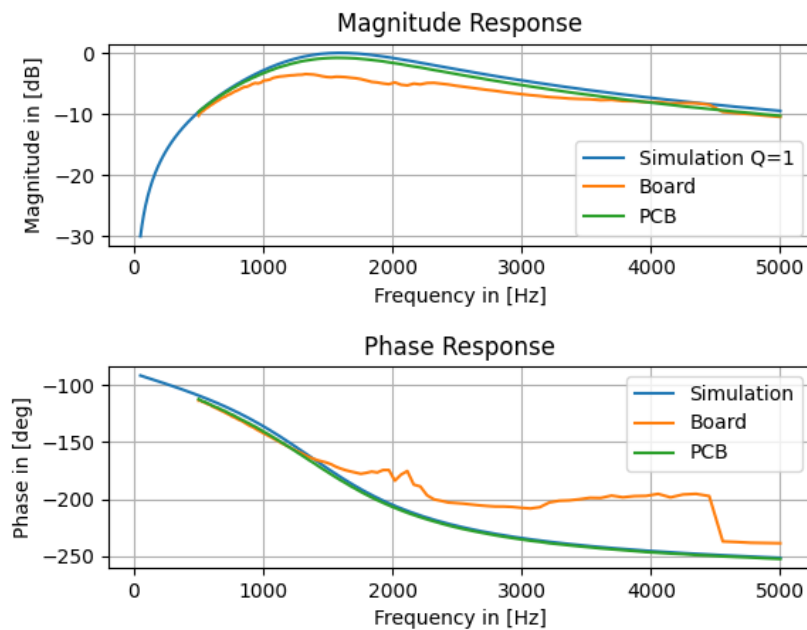


Abbildung 42: Frequenzantwort eines Bandpasses mit $Q=1$, $H=1$, $C=100\text{nF}$

In Abbildung 42 ist die Frequenzantwort eines weiteren Bandpasses zu sehen. Auch hier zeigt der obere Plot die Amplitudenantwort und der untere Plot die Phasenantwort. Der simulierte Bandpass ist in Blau, der Bandpass auf dem PCB in Grün und der Bandpass auf dem ASLK PRO Board in Orange dargestellt.

Weiterhin wird für diesen Bandpass eine Verstärkung von $H = 1$ und eine Güte von $Q = 1$ verwendet. Anders als zum vorherigen Bandpass haben die Kondensatoren eine Kapazität von $C = 100\text{nF}$. Dadurch verschiebt sich die Mittenfrequenz des Bandpasses.

$$f_0 = \frac{1}{2\pi \cdot R \cdot C} = \frac{1}{2\pi \cdot 1000\Omega \cdot 100 \cdot 10^{-9}\text{F}} = 1591.5\text{Hz}$$

Die neue Mittenfrequenz f_0 des simulierten Bandpasses liegt jetzt bei $f_0 = 1591.5\text{Hz}$. Auch der Bandpass auf dem PCB weist die gewünschte Mittenfrequenz von $f_0 = 1591.5\text{Hz}$ auf. Der Bandpass auf dem Board weicht ein bisschen von der gewünschten Mittenfrequenz ab. Bei diesem Bandpass liegt die Mittenfrequenz f_0 bei $f_0 = 1383\text{Hz}$.

Im Vergleich zum vorherigen Bandpass ist auch die Bandbreite BW angestiegen.

$$BW = \frac{f_0}{Q} = \frac{1591.5\text{Hz}}{1} = 1591.5\text{Hz}$$

Der simulierte Bandpass weist eine Bandbreite von $BW = 1591.5\text{Hz}$ auf. Der Bandpass auf dem PCB hat mit $BW = f_{g2} - f_{g1} = 2316\text{Hz} - 1028\text{Hz} = 1288\text{Hz}$ eine etwas kleinere Bandbreite. Im Vergleich zur Simulation ist diese Bandbreite um $\Delta BW = 1644\text{Hz} - 1288\text{Hz} = 356\text{Hz}$ schmaler. Der Bandpass auf dem Board weicht mit einer Bandbreite $BW = f_{g2} - f_{g1} = 2792\text{Hz} - 772\text{Hz} = 2020\text{Hz}$ deutlich von der Simulation ab. Grund dafür ist der eher flacherer Verlauf des Bandpasses auf dem Board nach der Mittenfrequenz f_0 .

Ebenfalls sind deutliche Unterschiede in der Dämpfung im Passband zu sehen. Die Simulation erreicht bei f_0 eine Dämpfung von 0dB . Der Bandpass auf dem PCB weist eine leicht höhere Dämpfung von ca. 1dB an ihrer Mittenfrequenz f_0 auf. Der Bandpass auf

dem Board hingegen weist mit ca. 4dB eine deutlich höhere Dämpfung im Passband auf. Im Vergleich zum Bandpass auf dem PCB sind $\Delta = 3dB$ Unterschied. Das bedeutet, dass der Bandpass auf dem Board im Vergleich zum Bandpass auf dem PCB nur die halbe Leistung passieren lässt.

Bei der Phase gibt es Anfangs zwischen der Simulation, dem Bandpass auf dem PCB und dem Bandpass auf dem Board nur minimale Unterschiede. Der Bandpass auf dem Board und der Bandpass auf dem PCB starten bei $f = 500Hz$ mit ca. $\varphi = -112$. Die Simulation weist eine leichte Differenz zu den anderen beiden Bandpässen auf. Ab ca. $f = 1383Hz$ weicht die Phase des Bandpasses auf dem Board von der Simulation ab. Während sich die Phase des Bandpasses auf dem PCB immer weiter der Simulation annähert, springt die Phase des Bandpasses auf dem Board springt die Phase vom Board immer wieder und driftet immer weiter ab.

Zum Ende hin bei $f = 4453Hz$ macht die Phase des Bandpasses auf dem ASLK PRO Board den größten Sprung mit $\Delta\varphi = -39$. Die Phase des simulierten Bandpasses und die Phase des Bandpasses auf dem PCB weisen keine Sprünge in ihren Verläufen auf. Am Ende ist die Phase der Simulation bei $\varphi = -250$, die Phase des Bandpasses auf dem Board bei $\varphi = -238$ und die Phase des Bandpasses bei $\varphi = -250$

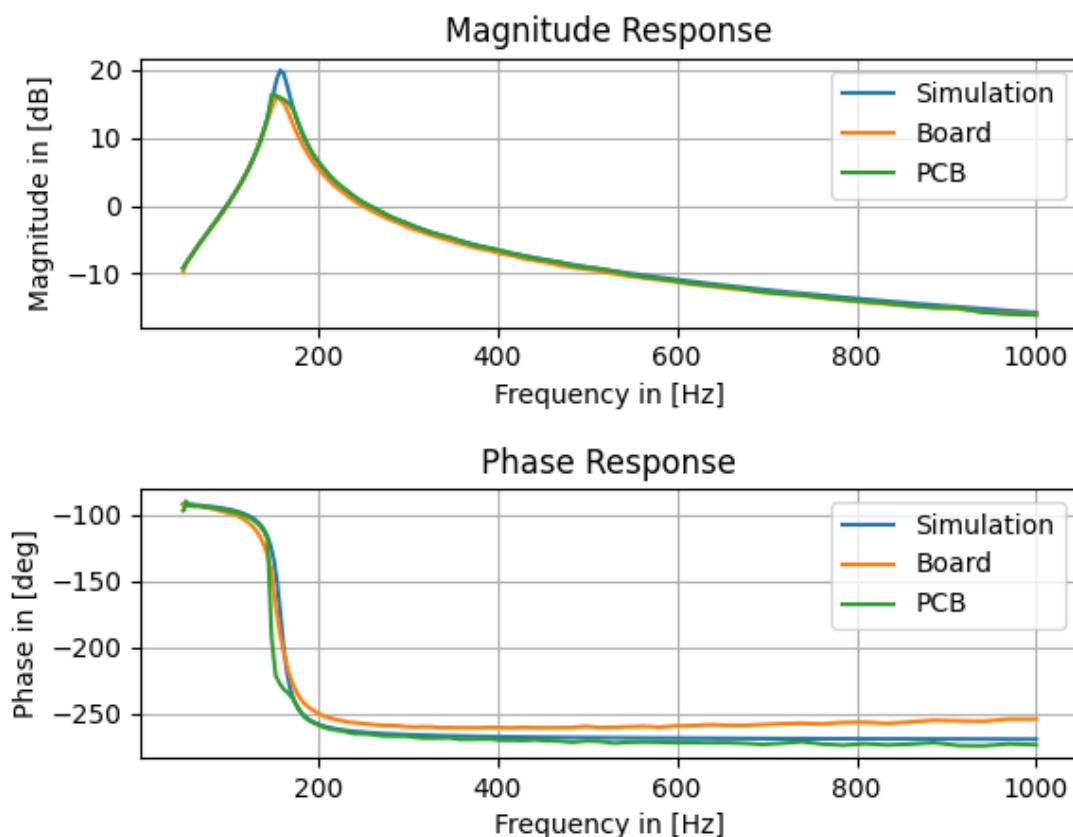


Abbildung 43: Frequenzantwort eines Bandpasses mit $Q=10$, $H=1$, $C=1\mu F$

Die Abbildung 41 zeigt die Frequenzantwort eines weiteren Bandpasses. Wie auch in den vorherigen Plots, zeigt der obere Plot die Amplitudenantwort und der untere Plot die Phasenantwort.

Verwendet wird eine Verstärkung von $H = 1$ und eine Güte von $Q = 10$. Die Kondensa-

toren haben eine Kapazität von $C = 1\mu F$.

$$f_0 = \frac{1}{2\pi \cdot R \cdot C} = \frac{1}{2\pi \cdot 1000\Omega \cdot 1 \cdot 10^{-6}F} = 159.15Hz$$

Dadurch verschiebt sich die Mittenfrequenz f_0 wieder zurück zu $f_0 = 159.15Hz$, welche auch beim simulierten Bandpass zu sehen ist. Der Bandpass auf dem PCB auf dem Board weisen mit $f_0 = 152.8Hz$ eine minimal nach unten verschobene Mittenfrequenz f_0 auf. Es kann aber auch sein, dass die Messauflösung zu gering um den Resonanzpunkt ist und die eigentliche Resonanz des Bandpasses auf dem PCB und auf dem ASLK PRO Board nicht erfasst wird.

Aus Gleichung 21 kann entnommen werden, dass mit höherer Güte bewirkt die Bandbreite sinkt.

$$BW = \frac{f_0}{Q} = \frac{159.15Hz}{10} = 15.92Hz$$

Damit hat der simulierte Bandpass eine Bandbreite von $BW = 15.92Hz$. Auf dem PCB weist der Bandpass mit $BW = f_{g2} - f_{g1} = 176.7Hz - 142.3Hz = 34.4Hz$ eine fast doppelt so große Bandbreite auf. Der Bandpass auf dem ASLK PRO Board hat mit $BW = f_{g2} - f_{g1} = 176.5Hz - 142.3Hz = 34.2Hz$ eine ähnliche Bandbreite, wie der Bandpass auf dem PCB.

Die höhere Güte $Q = 10$ bewirkt zu dem, dass die Resonanzschwingung deutlich stärker ausgeprägt ist. Die Überschwinger an der Mittenfrequenz f_0 können mit Gleichung 20 erklärt werden.

$$s_{1,2} = -\frac{\omega_0}{2Q} \pm \sqrt{\left(\frac{\omega_0}{2Q}\right)^2 - \omega_0^2}$$

Gut zu erkennen ist, dass die Lage der Pole in der s-Ebene abhängig von der Güte Q ist. Solange die Güte $Q > 0$ bleibt, liegen alle Pole in der linken Hälfte der s-Ebene, was ein stabiles System zur Folge. Sobald die Güte $Q > \frac{1}{2}$ ist, kommt es zum Schwingfall. Je größer die Güte wird, desto näher wandern die Pole zur an die imaginäre Achse der s-Ebene heran. Je näher die Pole an die imaginären Achsen liegen, desto stärker wird der Überschwinger an der Mittenfrequenz f_0 . Wenn die Güte zu groß wird, liegen die Pole nahezu direkt auf der imaginären Achse, was eine fast ungedämpfte Schwingung an der Mittenfrequenz f_0 zur Folge hätte.

Der simulierte Bandpass weist an f_0 eine Verstärkung von 20dB auf. Der Bandpass auf dem PCB und auf dem ASLK Pro Board weisen mit 16.5dB, bzw. 16.1dB eine niedrigere Verstärkung auf. Das nicht die 20dB aus der Simulation erreicht werden kann an der Messauflösung liegen. Die Anzahl der Messpunkte um den Bereich der Mittenfrequenz f_0 hätte größer gewählt werden sollen.

Im weiteren Verlauf weisen alle drei Bandpässe eine sehr ähnliche Amplitude auf. Es gibt nur minimale unterschieden zwischen diesen.

Beim Verlauf der Phase gibt es ebenfalls nur minimale Unterschiede. Alle drei Bandpässe starten bei $f = 50Hz$ mit $\varphi = -90.6$. Um die Mittenfrequenz f_0 , von $f = 125Hz$ bis $f = 200Hz$, fällt die Phase bei allen drei Bandpässen von ca. $\varphi = -100$ auf ca. $\varphi = -250$ bei dem Bandpass auf dem Board, bzw. $\varphi = -260$ beim simulierten Bandpass und dem Bandpass auf dem PCB. Die steile Phasendrehung ist ebenfalls der hohen Güte geschuldet. Im weiteren Verlauf weist keine Phasen Sprünge auf. Am Ende bei $f = 1000Hz$ liegt die Phase des simulierten Bandpasses bei ca. $\varphi = -267.8$. Der Bandpass auf dem PCB hat eine am Ende eine Phase von ca. $\varphi = -272^\circ$ und der Bandpass auf dem Board eine von ca. $\varphi = -253.3^\circ$.

5.4 Bandsperre

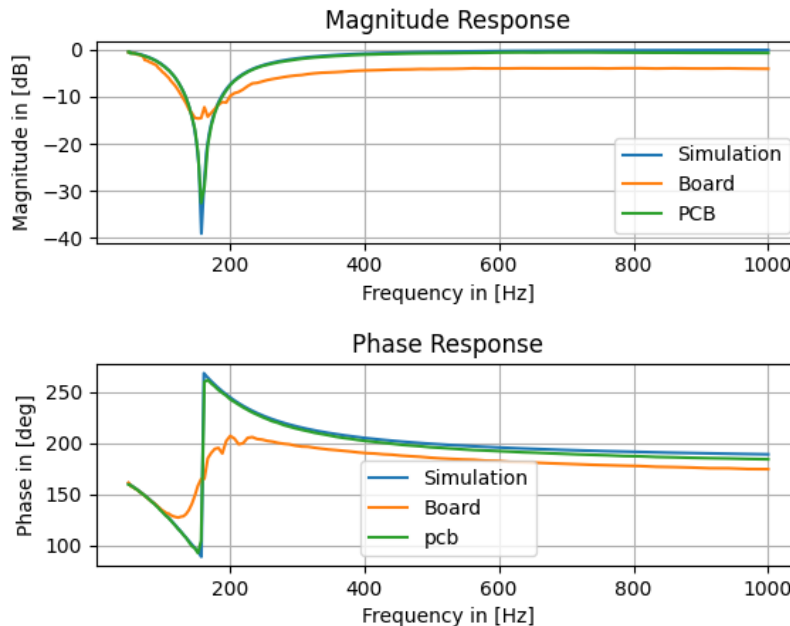


Abbildung 44: Frequenzantwort einer Bandsperre mit $Q=1$, $H=1$, $C=1\mu F$

Die Abbildung 44 zeigt die Frequenzantwort einer Bandsperre. Die Amplitudenantwort ist im oberen Plot und die Phasenantwort im unteren Plot dargestellt. Die blaue Kurve zeigt die simulierte Bandsperre. Die Bandsperre auf dem PCB wird von der grünen Kurve dargestellt und die Bandsperre auf dem ASLK PRO Board durch die orange Kurve. Verwendet wird eine Verstärkung von $H = 1$ und eine Güte von $Q = 1$. Die Kondensatoren haben eine Kapazität von $C = 1\mu F$.

$$f_0 = \frac{1}{2\pi \cdot R \cdot C} = \frac{1}{2\pi \cdot 1000\Omega \cdot 1 \cdot 10^{-6}F} = 159.15Hz$$

Damit liegt die Mittenfrequenz der simulierten Bandsperre bei $f_0 = 159.15Hz$. Sowohl die Bandsperre auf dem PCB, als die Bandsperre auf dem ASLK PRO Board weisen die gewünschte Mittenfrequenz $f_0 = 159.15Hz$ auf.

Mit der Gleichung 21 kann die Bandbreite der Bandsperre bestimmt werden.

$$BW = \frac{f_0}{Q} = \frac{159.15Hz}{1} = 159.15Hz$$

Die simulierte Bandsperre weist eine Bandbreite von $BW = 159.15Hz$ auf. Die Bandbreite der Bandsperre ist identisch zur Simulation. Die 3-dB Bandbreite des Boards kann nicht bestimmt werden. Das Board hat zwar bei $f_{g1} = 85.4Hz$ die erste 3dB Grenze, kommt aber weiteren Verlauf nicht wieder über die 3-dB.

An der Mittenfrequenz f_0 weist die simulierte Bandsperre eine minimale Sperrbanddämpfung von 38.4dB auf. Das PCB hingegen weist an der Mittenfrequenz nur eine Sperrbanddämpfung von 32dB auf. Die tatsächliche Dämpfung kann noch höher sein, jedoch müsste dafür im Bereich um die Mittenfrequenz f_0 die Anzahl der Messpunkte erhöht werden. Das Board weist eine deutlich niedrigere minimale Sperrdämpfung mit 14.1dB auf. Grund

dafür können die vielen Kabel sein, welche auf dem Board verwendet werden, sein.

Die Phasen aller drei Bandsperren starten bei $f = 50\text{Hz}$ mit $\varphi = 161.2$. Im weiteren Verlauf bis zur Mittenfrequenz $f_0 = 159.15\text{Hz}$ weicht die Phase der Bandsperre auf dem PCB nur minimal von der Phase aus der Simulation ab. Die Phase der Bandsperre auf dem ASLK Pro Board weicht immer weiter von der Simulation ab.

Ab der Mittenfrequenz $f_0 = 159.15\text{Hz}$ springt die Phase der Bandsperre auf dem PCB und der Simulation von $\varphi = 90.7$ auf $\varphi = 270$. Da der Phasensprung kleiner als $\Delta\varphi = 180 = \pi$ ist, wird dieser Sprung nicht von der unwrap-Funktion korrigiert. Die Phase der Bandsperre auf dem Board steigt nicht so steil an. Die Phase steigt langsam von $\varphi = 131$ auf $\varphi = 205$ über einen Bereich von ca. 75Hz .

Nach dem Sprung an der Mittenfrequenz f_0 fällt der Phasenverlauf der Simulation langsam ab. Der Phasenverlauf der Bandsperre auf dem PCB weicht dabei nur minimal ab. Die Phase der Bandsperre auf dem Board verläuft mit etwas größerer Differenzen ähnlich zur Simulation. Keine Phase weist weitere Sprünge auf.

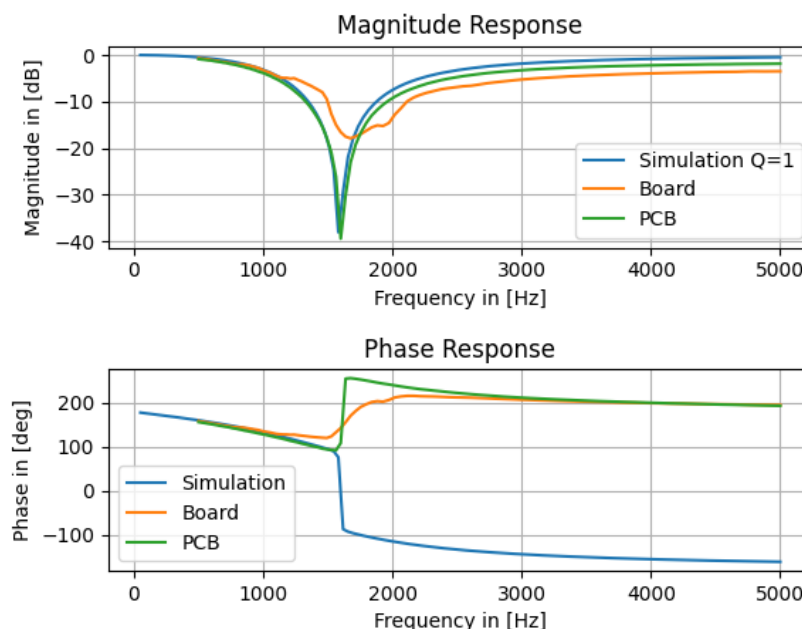


Abbildung 45: Frequenzantwort Bandsperre mit $Q=1, H=1, C=100\text{nF}$

Die Abbildung 44 zeigt die Frequenzantwort einer weiteren Bandsperre. Im oberen Plot sind die Amplitudenantworten und im unteren Plot die Phasenantworten aufgezeigt. Die simulierte Bandsperre ist in Blau, die Bandsperre auf dem PCB in Grün und die Bandsperre auf dem ASLK PRO Board in Orange dargestellt.

Eingestellt ist an der Bandsperre eine Verstärkung von $H = 1$ und eine Güte $Q = 1$. Verändert wird der Wert der Kondensatoren. Die neuen Kondensatoren haben eine Kapazität $C = 100\text{nF}$. Damit verschiebt sich die Mittenfrequenz f_0 der Bandsperre.

$$f_0 = \frac{1}{2\pi \cdot R \cdot C} = \frac{1}{2\pi \cdot 1000\Omega \cdot 1 \cdot 10^{-9}\text{F}} = 1591.5\text{Hz}$$

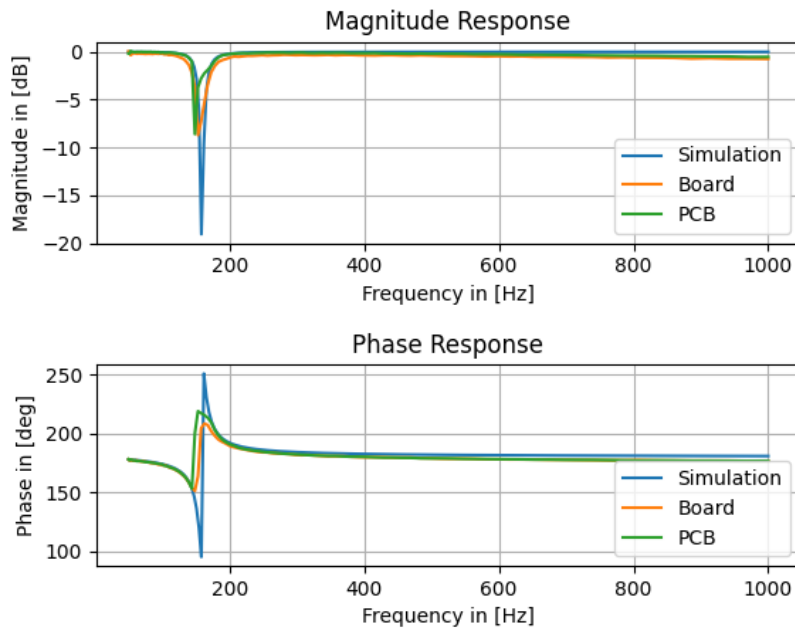
Die neue Mittenfrequenz liegt bei $f_0 = 1591.5\text{Hz}$. Im Vergleich zur Simulation, ist die Mittenfrequenz der Bandsperre auf dem PCB $f_0 = 1600\text{Hz}$ um $\Delta f_0 = 8.5\text{Hz}$ verschoben. Die Mittenfrequenz der Bandsperre auf dem Board $f_0 = 1672\text{Hz}$ weist eine größere

Verschiebung um $\Delta f_0 = 80.5\text{Hz}$ auf. In der Simulation erreicht die Bandsperre eine minimale Sperrdämpfung von 37.3dB. In der Praxis erzielt die Bandsperre auf dem PCB eine minimale Sperrdämpfung von 39.2dB. Die Bandsperre auf dem Board hingegen erreicht nur eine Dämpfung von 17.7dB. Die Dämpfung des Boards ist fast um den Faktor 100 kleiner, als die Dämpfung in der Simulation und auf dem PCB. Vermutet wird auch, dass die Verkabelung auf dem Board, die Leistung der Bandsperre negativ beeinflusst. Über die Gleichung 21 kann die Bandbreite der Bandsperre ermittelt werden

$$BW = \frac{f_0}{Q} = \frac{1591.5\text{Hz}}{1} = 1591.5\text{Hz}$$

Die Bandsperre weist in der Simulation eine Bandbreite von $BW = 1591.5\text{Hz}$ auf. Die Bandsperre auf dem PCB erreicht mit $BW = f_{g2} - f_{g1} = 2446\text{Hz} - 1025\text{Hz} = 1421\text{Hz}$ eine um $\Delta BW = 170.5\text{Hz}$ kleinere Bandbreite. Die Bandsperre auf dem Board erreicht mit $BW = f_{g2} - f_{g1} = 4036\text{Hz} - 1032 = 3004\text{Hz}$ eine viel größere Bandbreite. Das liegt an dem eher flachen Verlauf nach den Mittenfrequenz $f_0 = 1672\text{Hz}$. Der flache Verlauf ist der höheren Dämpfung Passband geschuldet.

Die Phase fängt bei der simulierten Bandsperre, als auch für die Bandsperre auf dem PCB und auf dem Board bei $f = 500\text{Hz}$ mit $\varphi = 159$ an. Bis zur Mittenfrequenz verlaufen die Phase ähnlich. Zwischen der Bandsperre auf dem PCB und der simulierten Bandsperre ist fast kein Unterschied erkennbar. Die Phase der Bandsperre auf dem Board weist etwas größere Abweichungen zur Simulation auf. Interessant wird es um die Mittenfrequenz f_0 . Während die Phase aus der Simulation von $\varphi = 92$ auf $\varphi = -85$ springt, springen die Phasen vom PCB und Board in die entgegengesetzte Richtung. Die Phase der Bandsperre auf dem PCB springt von $\varphi = 95$ auf $\varphi = 255$. Die Phase der Bandsperre auf dem Board steigt nicht so steil, wie die Phase der Bandsperre auf dem PCB. Die Phase steigt hier langsam von $\varphi = 122$ auf $\varphi = 216$. Eine genaue Erklärung für dieses Verhalten konnte, stand jetzt, noch nicht gefunden werden. Nach der Mittenfrequenz wird die Phasendifferenzen zwischen der Bandsperre auf dem PCB und der Bandsperre auf dem Board immer kleiner.

Abbildung 46: Bandsperre mit $Q=10$, $H=1$, $C=1\mu F$

In der Abbildung 46 ist die Frequenzantwort einer weiteren Bandsperre dargestellt. Der obere Plot zeigt die Amplitudenantwort und der untere Plot die Phasenantwort.

Die in 46 dargestellte Bandsperre besitzt eine Verstärkung von $H = 1$ und eine Güte $Q = 10$. Die Kondensatoren haben eine Kapazität von $C = 1\mu F$. Damit verschiebt sich die Mittenfrequenz f_0 wieder zurück zum Ausgangspunkt.

$$f_0 = \frac{1}{2\pi \cdot R \cdot C} = \frac{1}{2\pi \cdot 1000\Omega \cdot 1 \cdot 10^{-6}F} = 159.15Hz$$

Die Bandsperre auf dem PCB weist eine Mittenfrequenz von $f_0 = 145.8Hz$ auf, was einer Differenz von $\Delta f_0 = 13.35Hz$ zur simulierten Bandsperre entspricht. Die Bandsperre auf dem ASLK PRO Board hat eine Mittenfrequenz von $f_0 = 152.1Hz$ und weicht somit um $\Delta f_0 = 7.05Hz$ von der Simulation ab.

In der Simulation erreicht diese Bandsperre eine minimale Sperrbanddämpfung von 19dB. In der Praxis erreicht die Bandsperre auf dem PCB eine minimale Sperrbanddämpfung von nur 8.53dB und die Bandsperre auf dem Board nur eine von 8.74dB. Aufgrund der sehr schmalen Bandbreite, wo gleich noch drauf eingegangen wird, kann es sein, dass nicht die maximalen Werte beider Bandsperren erfasst werden. Die Auflösung der Messung hätte um die Mittenfrequenz f_0 rum deutlich erhöht werden müssen.

Wie in Gleichung 21 zu sehen ist, wird die Bandbreite der Bandsperre mit höherer Güte verkleinert.

$$BW = \frac{f_0}{Q}$$

Ebenfalls führt die höhere Güte zur eine stärker ausgeprägten Resonanz in der Mittenfrequenz f_0 . Diese können mit der Gleichung 20 erklärt werden.

$$s_{1,2} = -\frac{\omega_0}{2Q} \pm \sqrt{\left(\frac{\omega_0}{2Q}\right)^2 - \omega_0^2}$$

Die Lage der Pole ist mit von der Güte Q abhängig. Wenn die Güte $Q > 0$ ist, sind alle Pole in der linken Hälfte der s -Ebene. Mit höherer Güte wandern die Pole immer näher an die imaginäre Achse der s -Ebene. Wenn die Güte $Q > \frac{1}{2}$ wird, kommt es zum Schwingfall. Ab diesem Punkt wird die Schwingung an der Mittenfrequenz f_0 immer ausgeprägter. Wenn die Güte zu groß wird und die Pole nahezu auf der imaginären Achsen liegen, kommt es zu einer nahezu ungedämpften Schwingung. Das kann die Stabilität des Filters negativ beeinflussen.

$$BW = \frac{f_0}{Q} = \frac{159.15Hz}{10} = 15.92Hz$$

Der obigen Gleichung nach sollte die Bandsperre mit einer Güte von $Q = 10$ eine Bandbreite von $BW = 15.92Hz$ aufweisen, was auch in der Simulation zu sehen ist. In der Praxis weist die Bandsperre auf dem PCB eine Bandbreite von $BW = 153.5Hz - 144.4Hz = 9.1Hz$ einen minimal kleinere und die Bandsperre auf dem Board eine Bandbreite von $BW = 167.4Hz - 146.5Hz = 20.9Hz$ eine minimal höhere, als die Bandbreite der simulierten Bandsperre auf.

Die Phase startet für alle Bandsperren bei alle bei $f = 50Hz$ mit $\varphi = 179$. Je näher die Frequenz sich der Mittenfrequenz f_0 annähert, desto steiler fällt die Phase. An der Mittenfrequenz springt die Phase der simulierten Bandsperre von $\varphi = 96.3$ auf $\varphi = 252.2$ hoch. Dieser Sprung ist kleiner als $\Delta\varphi = 180 = \pi$, weshalb die unwrap-Funktion die Phase nicht korrigiert.

In der Praxis verlaufen die Phase der Bandsperre auf dem PCB und auf dem Board ähnlich zueinander. Beide Phasen springen bei der Mittenfrequenz f_0 nicht so hoch, wie die der simulierten Bandsperre. Die Phase der Bandsperre auf dem PCB springt von $\varphi = 155$ auf $\varphi = 220$ und die Phase von der Bandsperre auf dem von $\varphi = 151$ auf $\varphi = 208$. Die kleineren Sprünge kann an der Messauflösung von nur 100 Punkten liegen. Die Auflösung hätte um die Mittenfrequenz f_0 herum erhöht werden müssen.

Nach der Mittenfrequenz näher sich alle Phasen wieder an und weisen keine weiteren Sprünge auf.

6 Zusammenfassung

Die vorliegende Arbeit behandelt die umfassende Analyse und praktische Implementierung von biquadratischen IIR-Filtern zweiter Ordnung, wobei sowohl theoretische Grundlagen als auch experimentelle Validierung durch verschiedene Realisierungsformen systematisch untersucht wurden.

Das Hauptziel bestand in der Evaluierung der Konsistenz zwischen mathematischen Modellen, digitaler Umsetzung und physischen Schaltungsrealisierungen unter Verwendung des Multiple-Feedback-Biquad (MFB) als zentrale Filterarchitektur.

Der theoretische Teil der Untersuchung konzentrierte sich auf die systematische Herleitung der Übertragungsfunktionen für alle vier grundlegenden Filtertypen des MFB-Biquads: Tiefpass, Hochpass, Bandpass und Bandsperre.

Ausgehend vom idealisierten Operationsverstärker-Modell und der Anwendung des Knotenstromsatzes wurden die charakteristischen Übertragungsfunktionen entwickelt. Dabei konnte ein Vorzeichenfehler in der Standardreferenz für die Übertragungsfunktion $\frac{V_4}{V_i}$ identifiziert und durch konsistente Rückwärts-Herleitung korrigiert werden.

Die Pol-Nullstellen-Analyse in der komplexen s-Ebene verdeutlichte systemtheoretische Eigenschaften der Biquad-Struktur. Besonders hervorzuheben ist die Erkenntnis, dass sämtliche Filtertypen identische Polstellen aufweisen, während die spezifische Filtercharakteristik ausschließlich durch die Anordnung der Nullstellen im Zählerpolynom bestimmt wird. Die Untersuchung des Gütefaktors Q zeigte dessen zentrale Rolle für die Systemstabilität und das Frequenzverhalten auf, wobei drei charakteristische Betriebsbereiche identifiziert wurden: Schwingfall ($Q > 0,5$), aperiodischer Grenzfall ($Q = 0,5$) und Kriechfall ($Q < 0,5$).

Die Beziehung zwischen Gütefaktor und Bandbreite wurde für Bandpass- und Bandsperrenfilter gemäß $\Delta\omega = \frac{\omega_0}{Q}$ beschrieben, was die theoretische Vorhersagbarkeit des Filterverhaltens bestätigte. Die Grenzfrequenz $\omega_0 = \frac{1}{RC}$ erwies sich als zentrale Designgleichung, die eine Bestimmung der Filtercharakteristika durch Bauteilwahl ermöglicht.

Die Übertragung der analogen Filterkonzepte in den digitalen Bereich wurde durch die bilinearer Transformation beschrieben, wobei eine Abtastfrequenz von 44,1 kHz gewählt wurde. Diese Wahl eliminierte die Notwendigkeit von Prewarping-Techniken, da die Abtastfrequenz deutlich oberhalb der relevanten Signalfrequenzen lag. Die Implementierung wurde in der Second-Order-Section (SOS) Struktur realisiert, die aufgrund der natürlichen Biquad-Topologie optimale numerische Stabilität gewährleistet.

Der Vergleich zwischen analoger und digitaler Implementierung zeigte größtenteils Übereinstimmung in den Frequenzcharakteristika. Allerdings wurde beispielsweise bei der digitalen Bandsperre eine verbesserte Sperrbanddämpfung von -74,9 dB gegenüber -65,7 dB der analogen Version beobachtet. Die Pol-Nullstellen-Analyse im z-Bereich bestätigte die Systemstabilität durch Lokalisierung aller Pole innerhalb des Einheitskreises.

Die Python-basierte Referenzimplementierung mit Live-Audio-Verarbeitung validierte die praktische Funktionsfähigkeit der digitalen Filter und ermöglichte eine direkte Verifikation der theoretischen Vorhersagen durch akustische Tests.

Die experimentelle Validierung erfolgte mittels eines Red Pitaya STEMLab 125-14 als multifunktionales Messinstrument, das als Bode-Analysator konfiguriert wurde. Ein entscheidender Innovationsaspekt lag in der Entwicklung einer vollautomatisierten Messkette mit SCPI-basierter Fernsteuerung und Python-gestützter Datenverarbeitung. Diese Automatisierung umfasste Frequenz-Sweeps, CSV-Export, Kreuzkorrelations-basierte Phasenbestimmung und grafische Overlay-Darstellung mit Simulationsdaten.

Die Messgenauigkeit wurde durch implementierte Phase-Unwrapping-Algorithmen erheblich verbessert, die systematische 2π -Sprünge in der Phasendarstellung eliminierten. Die entwickelte Messtool-Chain ermöglichte reproduzierbare Messungen mit automatischer Datenarchivierung und direkter Vergleichsmöglichkeit zwischen theoretischen und experimentellen Ergebnissen.

Die Untersuchung umfasste drei Realisierungsebenen. Zunächst führten wir eine LTspice-Simulation durch, bauten anschließend den Filter auf dem ASLK-PRO-Board auf und entwickelten schließlich eine Leiterplatte (PCB). Die Analyseergebnisse zeigten markante Unterschiede in der Übereinstimmung mit theoretischen Vorhersagen.

Die Übereinstimmung zwischen Simulation und PCB-Messungen erwies sich als größtenteils deckend, mit geringen Abweichungen in der Amplitude sowie in der Phase. Diese Korrelation bestätigte sowohl die Genauigkeit der SPICE-Modelle als auch die Qualität der PCB-Implementierung. Teilweise übertraf das PCB sogar die Simulationsleistung, beispielsweise beim Bandpass mit einer Sperrbanddämpfung von 16,48 dB gegenüber 15,7 dB in der Simulation.

Die Steckboard-basierte Implementierung zeigte klare Leistungseinbußen, die auf parasitäre Effekte zurückzuführen sind. Typische Abweichungen umfassten Frequenzverschiebungen (bis zu 18 Hz bei 159 Hz Sollfrequenz), reduzierte Sperrbanddämpfung (teilweise über 10 dB schlechter) und Phasenverwaschungen. Diese Effekte wurden hauptsächlich auf parasitäre Kapazitäten zurückgeführt.

Der Tiefpassfilter zeigte die erwarteten -40 dB/Dekade oberhalb der Grenzfrequenz. Bei hohen Verstärkungsfaktoren ($H=100$) traten anomale Verläufe in beiden physischen Implementierungen auf, was auf Operationsverstärker-Begrenzungen hindeutet. Die Kapazitätsvariation bestätigte die erwartete Frequenzverschiebung gemäß $f_0 \propto \frac{1}{RC}$.

Die Hochpassfilter wiesen eine gute Übereinstimmung zwischen Simulation und PCB auf, jedoch fehlte der charakteristische +40 dB/Dekade Anstieg bei niedrigen Frequenzen in den praktischen Implementierungen. Das ASLK-Board zeigte konsistent höhere Dämpfung um etwa 3 dB.

Der Bandpassfilter zeigte eine teilweise gute Simulation-PCB-Korrelation. Bei $Q=10$ wurde die erwartete Resonanzüberhöhung beobachtet, wobei praktische Werte (16,5 dB) unter den Simulationswerten (20 dB) blieben, möglicherweise aufgrund unzureichender Messauflösung um die Resonanzfrequenz.

Die Bandsperrenfilter erreichten in der Simulation eine ausgezeichnete Sperrleistung (37-38 dB), die vom PCB nahezu reproduziert wurde (32-39 dB). Das ASLK-Board zeigte drastisch reduzierte Sperrwirkung (14-18 dB), was die Grenzen der Steckboard-Technologie für präzise Filteranwendungen aufzeigt.

Die Simulationen wurden direkt in KiCad durchgeführt und die Ergebnisse automatisch als CSV-Dateien exportiert. Diese Daten wurden anschließend in Python weiterverarbeitet, sodass sie unkompliziert mit echten Messergebnissen verglichen werden können. Im Messskript wurde eine Kreuzkorrelations-basierte Phasenmessung implementiert.

7 Fazit

Die vorliegende Untersuchung zu biquadratischen IIR-Filtern zweiter Ordnung liefert Erkenntnisse über die Übertragbarkeit theoretischer Filterkonzepte in praktische Implementierungen und zeigt dabei sowohl die Möglichkeiten als auch die Grenzen verschiedener Realisierungsansätze auf. Die durchgeführten Analysen bestätigen eindeutig, dass mathematisch fundierte Filtermodelle bei sachgerechter technischer Umsetzung nahezu ideale praktische Ergebnisse erzielen können.

Die theoretische Fundierung erwies sich als besonders wertvoll durch die Herleitung der Übertragungsfunktionen aller vier Filtertypen des Multiple-Feedback-Biquads. Dabei gelang nicht nur eine konsistente Darstellung der mathematischen Zusammenhänge, sondern auch ein Beitrag zur bestehenden Fachliteratur durch die Identifikation und Korrektur eines Vorzeichenfehlers in der Standardreferenz für die Übertragungsfunktion V_4/V_i . Die Pol-Nullstellen-Analyse verdeutlichte fundamentale systemtheoretische Eigenschaften der Biquad-Struktur und bestätigte, dass sämtliche Filtertypen identische Polstellen aufweisen, während die spezifische Filtercharakteristik ausschließlich durch die Anordnung der Nullstellen im Zählerpolynom bestimmt wird.

Ursprünglich war auch eine Implementierung einer dritten Filterordnung vorgesehen, die allerdings durch einen Konflikt des Messskripts zunächst verschoben und anschließend nicht umgesetzt wurde.

Ein zentrales Ergebnis der experimentellen Validierung ist der eindeutige Nachweis der Überlegenheit professionell gefertigter Leiterplatten gegenüber laborüblichen Steckboard-Aufbauten. Vor allem bei Frequenzen, bei denen parasitäre Effekte entstehen können. Die exzellente Korrelation zwischen LTspice-Simulationen und PCB-Messungen mit geringen Abweichungen belegt die hohe Vorhersagekraft der entwickelten mathematischen Modelle über alle vier Filtertypen hinweg.

Aus den quantifizierten Leistungsunterschieden lassen sich klare Designempfehlungen ableiten. Für alle Anwendungen mit Präzisionsanforderungen ist eine direkte PCB-Implementierung unverzichtbar, selbst in frühen Entwicklungsphasen.

Gerne wären noch detaillierte Fehleranalysen und Monte Carlo Simulationen durchgeführt worden, um die statistischen Auswirkungen von Bauteiltoleranzen systematisch zu quantifizieren und robuste Designrichtlinien zu entwickeln. Aufgrund der begrenzten verfügbaren Zeit konnten diese weiterführenden Untersuchungen jedoch nicht mehr realisiert werden.

Abschließend lässt sich festhalten, dass der Filter gut mit der Theorie übereinstimmt. Zugleich wird deutlich, wo Steckbrett-Aufbauten oder ungenaue Layouts an ihre Grenzen stoßen. Die automatisierte Messroutine und die kombinierte Analog-/ Digitalarchitektur sind nicht nur für dieses Projekt nützlich, sondern lassen sich leicht auf andere Schaltungsentwicklungen übertragen.

8 Anhang

Literatur

- [1] K. Rao und C. Ravikumar, *Analog System Lab Kit PRO MANUAL*. Texas Instruments, 2012.
- [2] S. Franco, *Design with Operational Amplifiers and Analog Integrated Circuits*, 4th. McGraw-Hill, 2015.
- [3] Göble, *Einführung in die Halbleiter-Schaltungstechnik, Auflage 6*. Springer, 2019.
- [4] R. Schaumann und M. E. V. Valkenburg, *Design of Analog Filters*, 2. Aufl. New York: Oxford University Press, 2009.
- [5] Juni 2009. Adresse: <https://quadrivium.nl/fa/manual/node40.html>.
- [6] Adresse: <https://de.wikipedia.org/wiki/Quantisierungsabweichung#/media/Datei:QuantErr.svg>.
- [7] P.-I. M. Goldenbaum, “Grundlagen der Informationstechnik,” 2024.
- [8] J. Wilczek, *How to digitize an analog filter with the bilinear transform: Wolfsound*, Jan. 2022. Adresse: <https://thewolfsound.com/bilinear-transform/>.
- [9] E. Funkner und M. Schwarz, *Efunkner/thesis-funkner-schwarz*, 2025. Adresse: <https://github.com/efunkner/Thesis-Funkner-Schwarz>.
- [10] J. E. “Implementierung digitaler Filter.” Beitrag im Mikrocontroller.net Forum. (2025), Adresse: <https://www.mikrocontroller.net/topic/320048>.
- [11] Adresse: <https://redpitaya.com/stemlab-125-14/>.
- [12] Adresse: <https://numpy.org/doc/stable/reference/generated/numpy.unwrap.html>.

Abbildungsverzeichnis

1	Operationsverstärker als Invertierender Integrator	2
2	Multiple-Feedback-Biquad [1]	3
3	Invertierender Integrator	4
4	invertierender Verstärker	4
5	invertierender Addierer	5
6	Blockschaltbild des Systems	9
7	Pol- und Nullstellenansicht des Bandpass mit $Q = 1$	12
8	Pol- und Nullstellen in der S-Ebene	12
9	SOS der Bandpass Übertragungsfunktion	13
10	Beispiel für die Quantisierung. Schwarz: Ursprungssignal, Grün: Quanti- siertes Signal, Rot: Abweichung der Quantisierung[6]	14
11	Nicht lineare Beziehung zwischen der s-Ebene und der z-Ebene[8]	16
12	Umgesetzte Differenzengleichung in Direkt Form 1	18
13	Umgesetzte Differenzengleichung in Direkt Form 2	18
14	Umgesetzte Differenzengleichung in Direkt Form 2	19
15	Umgesetzte Differenzengleichung in Direkt Form 2	19
16	Vergleich zwischen der analogen und digitalen Amplitudenantwort	21
17	Pol- und Nullstellen aller Filtertypen des digitalen Biquad-Filter	22
18	Ablaufplan des Testskripts für den digitalen Biquad-Filter	23
19	Ausgabe Fenster des Testprogramms	24
20	Darstellung des Frequenzverhaltens aller Ausgänge des Biquads	26
21	Red Pitaya STEMLab 125-14 v.1.1[11]	26
22	Gewählter Messaufbau	27
23	ASLK-Board	29
24	PCB-Layout	29
25	PCB-Schematic	30
26	PCB-Layer 1	31
27	PCB-Layer 2	31
28	Bode Analyser in der Weboberfläche	32
29	Flowchart des erstellten Messprogramms	34
30	Flowchart des erstellten Messprogramms	35
31	Teilaufbauplan zur Auswahl des Red Pitaya	36
32	Teilaufbauplan zur Bestimmung der Phasenantwort	38
33	Vergleich zwischen der Phasenantwort mit und ohne Unwrap	39
34	Tiefpass mit $Q=1$, $H=1$, $C=100\text{nF}$	41
35	Tiefpass mit $Q=1$, $H=1$, $C=1\mu\text{F}$	42
36	Tiefpass mit $Q=10$, $H=1$, $C=1\mu\text{F}$	43
37	Tiefpass mit $Q=1$, $H=100$, $C=1\mu\text{F}$	43
38	Hochpass mit $Q=1$, $H=1$, $C=100\text{nF}$	44
39	Hochpass mit $Q=1$, $H=1$, $C=1\mu\text{F}$	45
40	Hochpass mit $Q=10$, $H=1$, $C=1\mu\text{F}$	45
41	Frequenzantwort eines Bandpasses mit $Q=1$, $H=1$, $C=1\mu\text{F}$	46
42	Frequenzantwort eines Bandpasses mit $Q=1$, $H=1$, $C=100\text{nF}$	48
43	Frequenzantwort eines Bandpasses mit $Q=10$, $H=1$, $C=1\mu\text{F}$	49
44	Frequenzantwort einer Bandsperre mit $Q=1$, $H=1$, $C=1\mu\text{F}$	51
45	Frequenzantwort Bandsperre mit $Q=1$, $H=1$, $C=100\text{nF}$	52

46	Bandsperr mit $Q=10$, $H=1$, $C=1\mu F$	54
----	---	----