

Hochschule der Medien Stuttgart
Fakultät Electronic Media
Audiovisuelle Medien (AM7)

Referat zum Thema:

Digitale Filter

Funktionsweise digitaler Filter in der Audioproduktion

Vorgelegt von: Benjamin Schardt
Matrikelnummer: 5017119
E-mail: bs127@hdm-stuttgart.de

Vorgelegt am: 30. Juli 2026
Prüfer: Prof. Oliver Curdt

Inhaltsverzeichnis

1	Einleitung.....	1
2	Grundlagen.....	1
2.1	<i>Digitale Signale.....</i>	<i>1</i>
2.2	<i>Signalanalyse</i>	<i>4</i>
3	Digitale Filter.....	5
3.1	<i>Faltung</i>	<i>5</i>
3.2	<i>Filter Eigenschaften.....</i>	<i>7</i>
3.3	<i>FIR-, und IIR-Filter.....</i>	<i>8</i>
3.4	<i>Biquad Filter</i>	<i>11</i>
4	Fazit und Aussichten.....	12
5	Quellen	12

1 Einleitung

Digitale Filter sind aus der modernen Musikproduktion nicht mehr wegzudenken. Verwendung finden sie in zahlreichen Digital Audio Workstation's (DAW's), Effekt-Plugins wie Equalizer, Phaser oder Hall, aber auch etwas versteckter beim Oversampling, und in sehr vielen weiteren fundamentalen Bereichen der Musikproduktion. Im Folgenden wird zunächst auf die Grundlagen digitaler Signale eingegangen, was die Basis für das Verständnis digitaler Filter bildet. Anschließend werden Grundlegende Filtertypen und Methoden vorgestellt, welche in der Programmierumgebung Max/MSP weiter untersucht werden. Besonders werden Faltungsfiler und IIR Filter wie der Biquad Filter beleuchtet und deren Funktionsweise erklärt. Da das Thema sehr umfangreich ist, versteht sich diese Arbeit mehr als Einführung, und hat keinen Anspruch auf Vollständigkeit.

2 Grundlagen

2.1 Digitale Signale

Um sich mit digitalen Filtern zu beschäftigen, braucht man Kenntnisse über digitale Signale. Denn diese werden mithilfe eines Filters letztendlich manipuliert. Im Folgenden nun die wichtigsten Eigenschaften digitaler Signale.

Digitale Signale sind zeitdiskret, bedeutet sie besitzen diskrete Abtastzeitpunkte, die je nach Abtastfrequenz f_A mit der Periodendauer T_A auseinanderliegen. Da es diese diskreten Abtastpunkte gibt, wird bei der Analog-Digital Wandlung aus einem analogen Signal $x(t)$, das digitale Signal $x[n \cdot T_A]$, wobei $T_A = \frac{1}{f_A}$ und $n \in \mathbb{N}$. Das Signal ist also nur zu den Zeitpunkten definiert, an denen es auch abgetastet wurde.

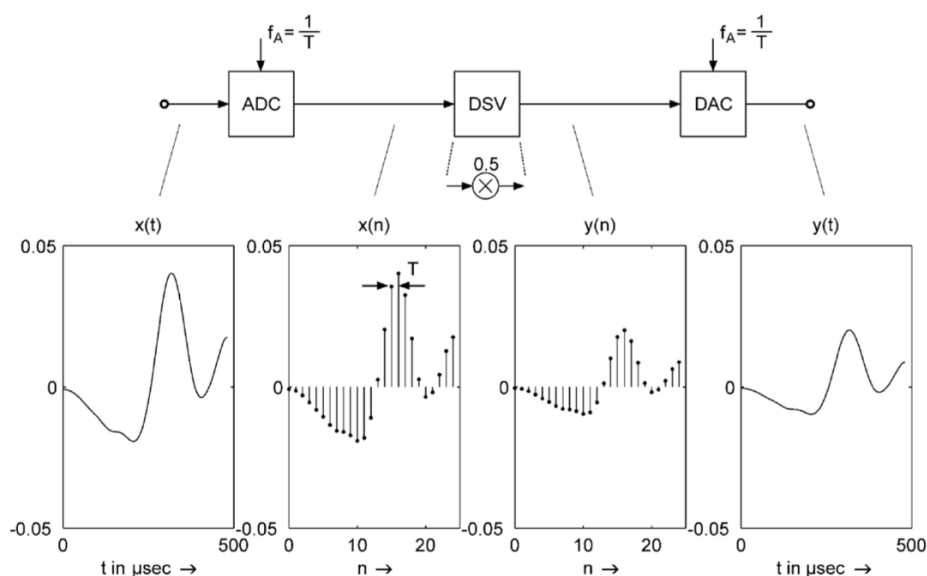


Abbildung 1: Diskretes Signal
Quelle: Weinzierl (2008)

Wie auf Abbildung 1 zu erkennen, wird das Signal zuerst gewandelt, dann bearbeitet (in dem Beispiel oben mit einem Gewichtungsfaktor von 0,5) und anschließend wieder analog gewandelt, womit es

wieder in eine Spannung übersetzt wird, die dann z.B. an einen Treiber eines Lautsprechers weitergeht. Um den genauen Wandlungsprozess soll es hier nicht gehen, aber der Kontext ist hilfreich.

Für den Rechner ist das Signal damit eine Zahlenfolge, welches i.d.R. in Buffer, also Abschnitte mit einer festgelegten Anzahl an Samples (z.B. 512) verarbeitet wird, und das in nahezu Echtzeit. Welche Werte die einzelnen Samples annehmen können, hängt von der verwendeten Bittiefe ab, was für die theoretische Signalverarbeitung allerdings keine große Rolle spielt, da man von idealisierten Werten ausgeht und Einschränkungen und Rundungsfehler zunächst ignorieren kann. Diese Rundungsfehler können allerdings eine sehr große Rolle spielen, wenn das System eine niedrige Bittiefe verwendet, beispielsweise bei Microcontrollern. In der DAW spielt dies meist keine große Rolle, weshalb es in dieser Arbeit weitestgehend ignoriert wird.

Es gibt drei grundlegende Rechenoperationen die der Rechner auf diese Buffer anwenden kann.

1. Gewichtung eines Signals.

Dabei wird das Eingangssignal $x[n]$ mit einem Faktor a multipliziert. Für das Ausgangssignal $y[n]$ gilt

$$y[n] = a \cdot x[n] \quad (1)$$

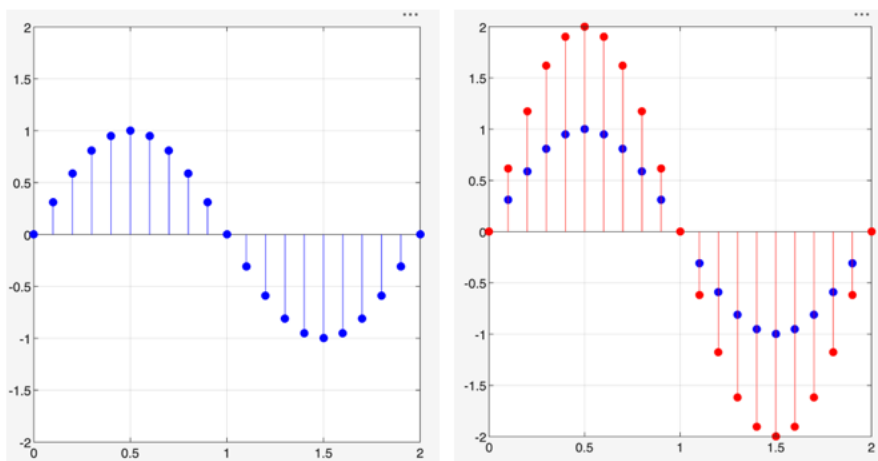


Abbildung 2: Gewichtetes Signal
Quelle: Eigene Darstellung

Hierbei wird jeder einzelne Samplewert des Eingangssignals $x[n]$ (blau) mit einem Faktor a multipliziert und bildet den neuen Ausgangswert $y[n]$ (rot).

2. Summation von gewichteten Signalen.

Hierbei werden zwei oder mehr gewichtete Eingangssignale Sample für Sample addiert. Dieser Vorgang kann mit beliebig vielen Signalen durchgeführt werden. Für das Ausgangssignal $y[n]$ gilt

$$y[n] = a_1 \cdot x_1[n] + a_2 \cdot x_2[n] \quad (2)$$

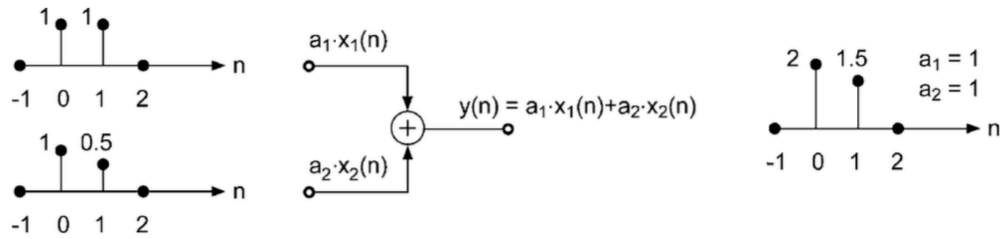


Abbildung 3: Addition von gewichteten Signalen
Quelle: Weinzierl 2008

3. Verzögerung eines Signals um m Samples.

Das Signal $x[n]$ (blau) wird um m Samples zeitlich verzögert und ergibt das Ausgangssignal $y[n]$ (rot). Dabei ist wichtig, dass das Signal nur um ganze Abtastperioden verzögert werden kann.

$$y[n] = x[n - m] \quad (3)$$

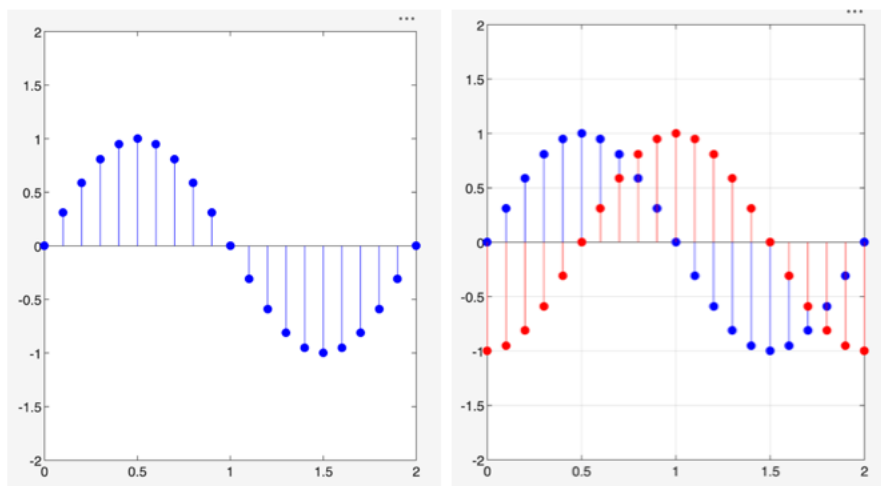


Abbildung 4: Verzögerung eines Signals um m Sample
Quelle: eigene Darstellung

Kombiniert man diese drei Rechenoperationen kann man daraus Filter realisieren. Gleichung 1, 2 und 3 sind Differenzgleichungen. Diese verwendet man i.d.R. um einen Filter in einer Programmiersprache zu implementieren.

Eine weiterer wichtiger Grundsatz der digitalen Audiotechnik ist das Nyquist-Shannon Theorem, welches besagt, dass die Abtastfrequenz mehr als doppelt so hoch sein muss wie die höchste Frequenz des Signals welches man abbilden will. In einigen Darstellungen wird daher gerne eine normalisierte Frequenz benutzt, da man dadurch die tatsächliche Abtastrate ausklammern kann.

Beispiel:

Um ein Signal mit 20 kHz darzustellen benötigt man mindestens eine Abtastrate von 40 kHz, da es sonst zu Aliasingartefakten kommt. Da die Nyquistfrequenz $f_N = \frac{f_s}{2}$, ergibt sich daraus $f_{norm} = \frac{f}{f_N} = 2 \cdot \frac{f}{f_s}$. Somit nimmt die höchste darstellbare Frequenz den Wert 1 an, und alle Frequenzen darunter einen Wert zwischen 0 und 1.

2.2 Signalanalyse

Die Systemanalyse greift auf ähnliche Methoden zurück wie die Signalanalyse. Die grundlegendste Methodik bildet hierbei die Fouriertransformation, genauer die diskrete Fouriertransformation (DFT) im Fall von digitalen Signalen.

Mithilfe der Fouriertransformation lässt sich jedes Signal in seine Frequenzanteile zerlegen. Wie man an der Definition dieser erkennt, handelt es sich um ein komplexes Spektrum, welches deshalb in Betrags- und Phasenspektrum unterteilt werden kann. Das Betragsspektrum entspricht dem Amplitudenspektrum eines Signals und das Phasenspektrum der Phasenverschiebung einzelner Frequenzen.

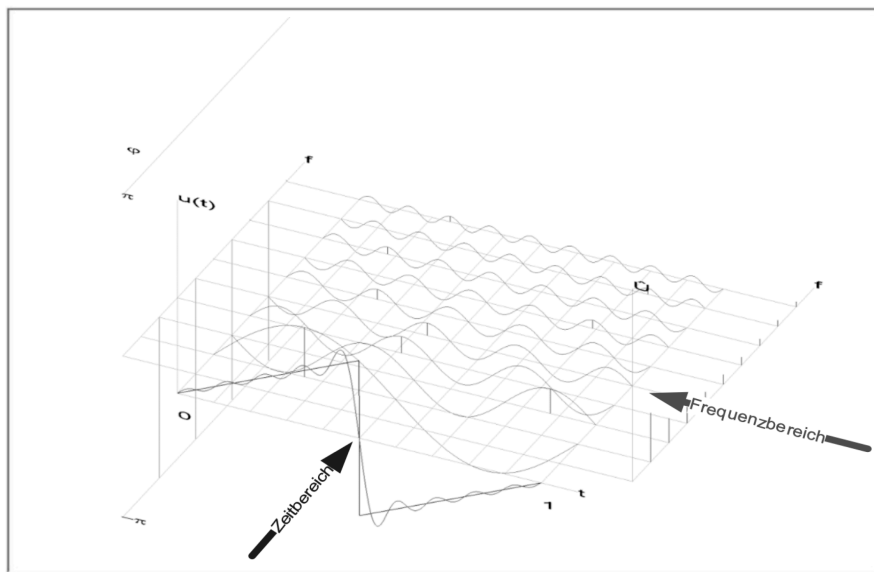


Abbildung 5: Sägezahnsignal und einzelne Bestandteile
Quelle: Karrenberg 2012

Mit Abbildung 5 lässt sich das Prinzip der DFT gut veranschaulichen. Im Zeitbereich sieht man eine Sägezahn-Schwingung $u(t)$, welche im Bereich 0 bis 1 genau eine Periodendauer umfasst. Diese lässt sich mit den Sinusschwingungen, die im Frequenzbereich dargestellt sind bilden, indem diese aufaddiert werden. Um eine perfekte Sägezahnschwingung so zu reproduzieren, bräuchte man unendlich viele Sinusschwingungen (auch Obertöne genannt), was in der Realität natürlich nicht vorkommen kann. Deshalb sieht man oben auch die etwas wellig aussehende Funktion, welche näherungsweise der Sägezahnschwingung entspricht. Zu erkennen ist auch, dass die Obertöne in der Phase variieren, also nicht alle Sinusschwingungen den „gleichen Startpunkt“ haben. Ein wichtiger Grundsatz: jedes periodische Signal lässt sich in seine einzelnen Sinusschwingungen aufteilen.

Mathematisch funktioniert das ganze so; Das Eingangssignal $x[n]$ wird mit einer komplexen e-Funktion einer bestimmten, normierten Frequenz k multipliziert, und anschließend die Werte summiert. Dadurch wird die Korrelation dieser beiden Signale getestet. Dies wird für $k = 1, 2, 3, \dots, N - 1$, wobei N die Anzahl der Samples des zu testenden Signals ist. Hier wird auch wieder das Unschärfeprinzip der Fouriertransformation deutlich, denn je weniger Abtastwerte das Signal hat, desto ungenauer bzw. breiter sind die Frequenz Bins. Die DFT ist definiert als:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{j2\pi kn}{N}} \quad (4)$$

wobei X_k = komplexes Spektrum, abhängig von k : Frequenz, N : Anzahl der Samples, x_n : Eingangssignal.

Dadurch enthält man das komplexe Spektrum X_k , welches in Betrag und Argument aufgeteilt werden kann. Beide Informationen sind sehr wichtig für die Analyse eines Signals. Warum diese Analyse in Betrag und Phase für die Systemanalyse nützlich ist, wird im nächsten Kapitel deutlich.

3 Digitale Filter

Ein digitaler Filter ist eine Art von digitalem System. „Die Abbildung eines Eingangssignals $x(n)$ in ein Ausgangssignal $y(n)$ mit Hilfe eines Rechenalgorithmus, der das Ausgangssignal $y(n)$ aus dem Eingangssignal $x(n)$ mit Hilfe der oben genannten Grundoperationen berechnet, wird als digitales System bezeichnet.“ (Weinzierl 2008, S. 817). Dabei spielen vorrangig LTI-Systeme die größte Rolle. Ein LTI System (linear, time-invariant System) hat bestimmte Eigenschaften, welche es vereinfachen ein System zu analysieren:

Linearität: Das System erfüllt das Superpositionsprinzip. Das bedeutet, dass die Antwort auf die Summe mehrerer Eingangssignale gleich der Summe der einzelnen Antworten ist und eine Skalierung des Eingangssignals zu einer entsprechenden Skalierung des Ausgangssignals führt. (Smith III, 2026).

Zeitinvarianz: Die Eigenschaften des Systems ändern sich nicht mit der Zeit. Wird ein Eingangssignal zeitlich verschoben, verschiebt sich auch das Ausgangssignal um denselben Betrag, ohne seine Form zu verändern. (Smith III, 2026).

Diese beiden Eigenschaften erlauben eine genaue Analyse des Systems mit Hilfe der Fourier- bzw. Z-Transformation.

3.1 Faltung

Man betrachte das Signal $x[n]$. Dieses zeitdiskrete Signal lässt sich ebenfalls als Folge von Diracstößen darstellen. Das funktioniert, da ein Diracstoß per Definition eine Funktion ist, die bei $n = 0$ den Wert 1 hat und ansonsten den Wert 0 annimmt. Diese Definition gilt allerdings nur für zeitdiskrete Signale (Die Definition für die zeitkontinuierliche Deltafunktion ist über ein Integral definiert). Damit „baut“ man sich das Signal x quasi aus ganz vielen einzelnen Diracstößen zusammen (deshalb die Summe). Die Diracstöße werden verschoben (durch sampleweise Verzögerung) und mit den Werten des Signals x gewichtet.

$$x[n] = \sum_{m=-\infty}^{+\infty} x[m] \cdot \delta[n - m] \quad (5)$$

Man nennt diesen Diracstoß auch Impuls. Ein Impuls ist ein Signal, welches alle Frequenzen enthält, weshalb es ein ideales Testsignal für Systeme darstellt. Schickt man so ein Impuls nun durch ein LTI System, kann man anhand der Impulsantwort das System vollständig beschreiben. Da unser Eingangssignal allerdings eine Reihe von gewichteten Impulsen ist, lässt sich das Ausgangssignal einfach als gewichtete Summe von Impulsantworten darstellen. Aus der obigen Gleichung wird dann

$$x[n] = \sum_{m=-\infty}^{+\infty} x[m] \cdot h[n - m] \quad (6)$$

Die Summe verändert sich also nur bezüglich des Diracstoßes, dieser wird zur Impulsantwort $h[n]$.

Da die Impulsantwort (IR) das System vollständig beschreibt, kann man durch die Analyse dieser IR herausfinden, welchen Einfluss das System auf Betrag und Phase für jedes beliebige Eingangssignal haben muss. Wendet man also auf $h[n]$ die Fouriertransformation an, erhält man eine Funktion die das System beschreibt.

$$H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h[n]e^{-j\omega n}$$

Zerlegt man diese Funktion nun in Betrags- und Phasenspektrum, erhält man den Frequenzgang und Phasengang des Filters. Diese Funktion nennt man auch Übertragungsfunktion. Sie beschreibt gleichzeitig das Verhältnis von Ausgang zu Eingang.

$$y[n] = x[n] * h[n]$$

Wendet man auf die Faltungsgleichung die Fouriertransformation an, erhält man eine Multiplikation im Frequenzbereich. Diese Eigenschaft ist sehr fundamental. Es ergibt sich:

$$Y(e^{j\omega}) = H(e^{j\omega})X(e^{j\omega})$$

Durch umstellen erhält man:

$$H(e^{j\omega}) = \frac{Y(e^{j\omega})}{X(e^{j\omega})} \quad (7)$$

Diese Funktion wird dann typischerweise z.B. in einem Bodediagramm dargestellt.

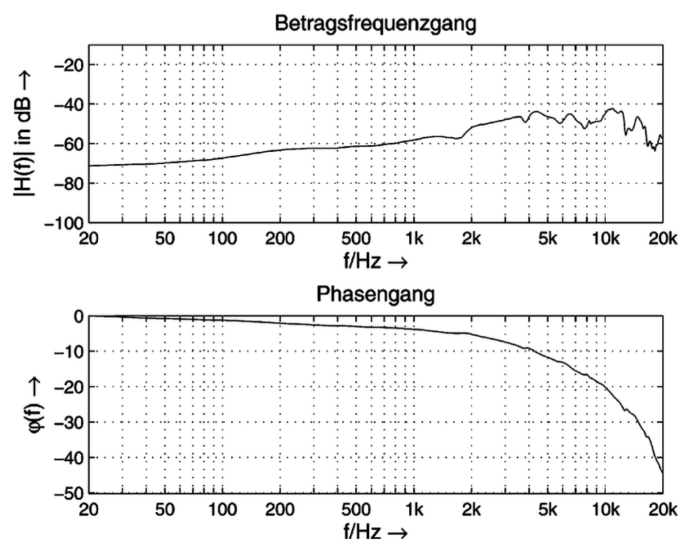


Abbildung 6: Beispiel Betragsfrequenzgang und Phasengang eines Lautsprechers
Quelle: Weinzierl 2008

Aus dem Phasengang leitet sich dazu noch die Gruppenlaufzeit ab, welche mathematisch die Ableitung des Phasengangs ist. Sie beschreibt für alle Frequenzen die Änderungsrate der Phasenverschiebung. Sie wird meist in Millisekunden angegeben und gibt den absoluten Verzögerungswert pro Frequenz an. In manchen Equalizern und anderen Anwendungen wird die Gruppenlaufzeit angezeigt, um dem Nutzer zu zeigen, ob und wie das System die einzelnen

Frequenzen verschieden verzögert. Je nach Anwendung kann das aussagekräftiger als das Phasenspektrum sein.

3.2 Filter Eigenschaften

Alle nicht-trivialen Filter beeinflussen entweder das Betrags- oder Phasenspektrum eines Signals. Für gewöhnlich erwartet man in der Musikproduktion eine Veränderung des Betrags- bzw. Amplitudenspektrums. Z.B. will man bestimmte Frequenzen eines Signals anheben oder absenken. Ein typischer Filter, der z.B. in parametrischen Equalizern verwendet wird, soll nun anhand eines Tiefpass Filters erläutert werden.

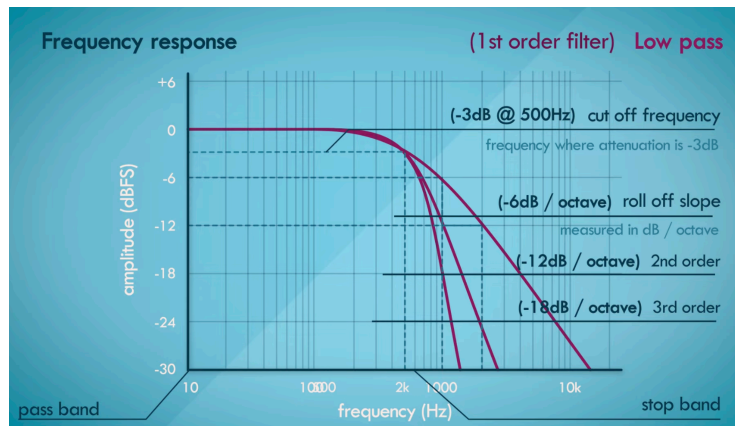


Abbildung 7: Frequenzgang eines Tiefpass Filter
Quelle: Murthy 2022

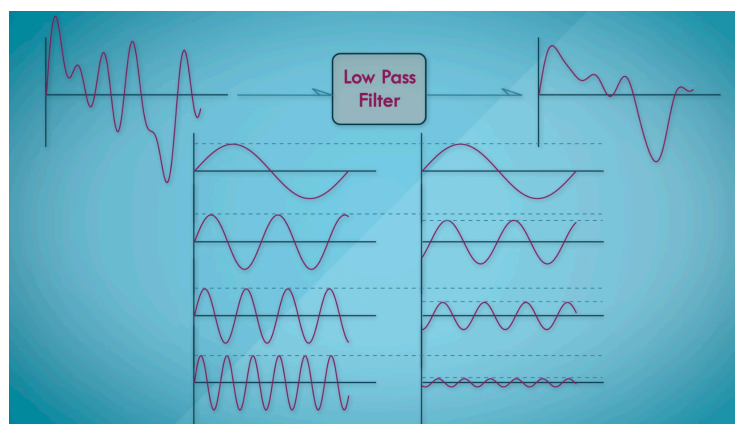


Abbildung 8: Amplituden- und Phasenveränderung nach Filterung
Quelle: Murthy 2022

Dieser Filter besitzt zunächst einen Durchlass- sowie Sperrbereich. Ein idealer Filter (den es in der Realität so nicht geben kann) würde alle Frequenzen im Sperrbereich auf eine Amplitude von null setzen, und die Amplitude im Durchlassbereich nicht verändern. Dies ist nicht möglich aufgrund des Unschärfepinzips der Fouriertransformation (Karrenberg 2012). Daraus folgt, dass es eine sogenannte Flankensteilheit gibt, welche in dB pro Oktave angegeben wird. Diese gibt an, wie steil der Verlauf der Übertragungsfunktion $|H(\omega)|$ ist. Die Grenzfrequenz befindet sich dabei bei -3dB (oder $\frac{1}{\sqrt{2}}$ mal den Eingangspegel). Die Darstellung auf der rechten Seite von Abbildung 8 zeigt die einzelnen Frequenzen, welche nach der Filterung in der Amplitude und Phase beeinflusst sind. Es ist zu erkennen, dass je mehr die Frequenzen abgesenkt werden, desto mehr sind sie auch in der Phase verschoben. Diese Eigenschaft ist typisch und kann nur bei sogenannten linearphasigen Filtern umgangen werden.

Bei Glockenfiltern kommt noch der Gütefaktor hinzu, welcher beschreibt, wie breitbandig der Filter wirkt. Diese wird mit der Differenz beider Grenzfrequenzen ausgerechnet.

$$Q = \frac{f_0}{\Delta f} \quad (8)$$

Darüber hinaus gibt es noch Shelvingfilter, welche es erlauben ab einer Grenzfrequenz einen ganzen Bereich (entweder nach oben oder unten im Frequenzbereich) Anzuheben oder abzusenken. Genauer soll hier nicht auf die verschiedenen Filtertypen eingegangen werden, da sie im Grunde alle gleich aufgebaut sind. Sie alle verwenden in der digitalen Domäne dieselben Bausteine; die oben vorgestellten Rechenoperationen.

3.3 FIR-, und IIR-Filter

FIR steht für „finite impuls response“, IIR für „infinite impulse response“. Die Namen beziehen sich auf die oben erläuterte Impulsantwort eines Filters, also das Signal was das System als Ausgang rausgibt, wenn der Eingang ein Impuls ist. Bei allen bisherigen Filtern in diesem Referat handelte es sich um FIR Filter, da die Impulsantwort vorher festgelegt wurde. Zum Beispiel beim gleitenden Mittelwert Filter wird immer der Mittelwert von n benachbarten Samplewerten gebildet. Da n nicht unendlich groß sein kann, handelt es sich zwangsläufig um einen FIR Filter. Dieser sieht als Schaltbild in etwa so aus:

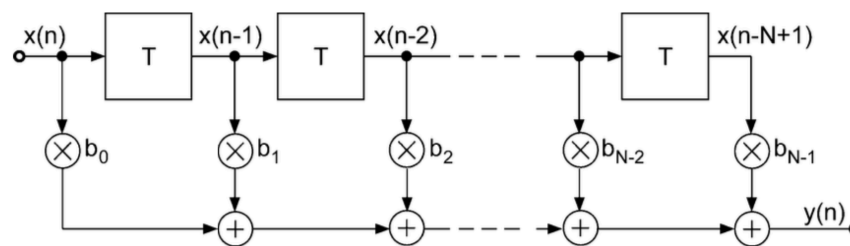


Abbildung 9: FIR Filter
Quelle: Weinzierl 2008

Im Schaltbild entsprechen $b_0, b_1, \dots, b_{N-1}$ der Impulsantwort $h[n]$. Zu erkennen ist auch, dass für die Berechnung des Ausgangssignals lediglich Eingangswerte verwendet werden. Das muss auch so sein, denn die Gleichung (6) beschreibt genau diese Summation von mit dem Eingangssignal gewichteten Impulsantworten.

Anders sieht das bei IIR Filtern aus. Hier werden ebenfalls die (verzögerten) Ausgangswerte wieder mit verrechnet, was dazu führt, dass eine Feedbackschleife entsteht. Dadurch kann ein System auch instabil werden, wenn nämlich das Signal ungebremst größer wird. Wie man herausfindet ob ein System stabil ist oder nicht, wird noch genauer beschrieben.

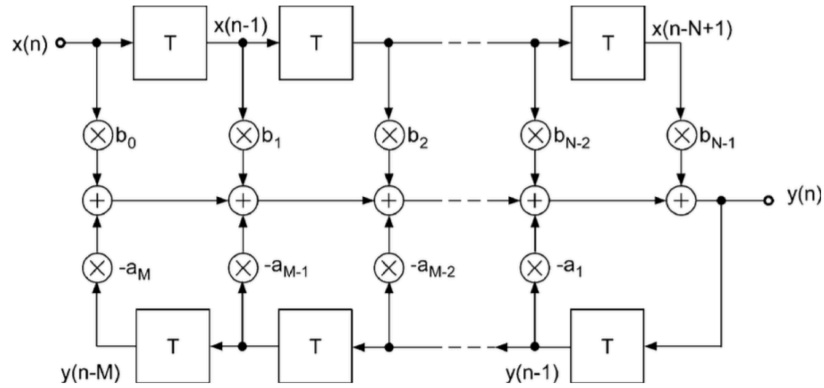


Abbildung 10: IIR Filter
Quelle: Weinzierl 2008

Auf Abbildung 10 sieht man ein Schaltsdiagramm eines IIR Filters. Es ist zu erkennen, dass die verzögerten Ausgangswerte, gewichtet mit Koeffizienten $a_0, a_1, \dots, a_M$ mitaddiert werden. Eine typische Differenzengleichung kann z.B. so aussehen:

$$y[n] = (1 - a)x[n] + ay[n - 1] \quad (9)$$

Hier sieht man, dass ein verzögerter Ausgangswert auf der rechten Seite steht. Da die Fouriertransformation aber nur für Funktionen existiert, die nicht divergieren, was bei (9) nicht gesichert ist, braucht man hier ein anderes Werkzeug: Die Z-Transformation.

Sie erlaubt es neben der Frequenzanalyse auch Auskunft über exponentielle Verläufe in Signalen zu machen. Sie ist definiert als:

$$H(z) = \sum_{n=-\infty}^{\infty} h[n] \cdot z^{-n} \quad (10)$$

mit

$$z = r \cdot e^{j\omega}$$

Mit r erhalten wir also noch eine zusätzliche Variable zur Fouriertransformation. Diese ist wichtig für Systeme, welche Feedback erzeugen (also IIR Filter), da mit r^{-n} ein exponentieller Verlauf an die komplexe e -Funktion drangmultipliziert wird. Diese reelle Exponentialfunktion wirkt sozusagen der womöglich divergenten Eingangsfunktion entgegen.

Die Z-Transformation ist essentiell für das entwerfen von Filtern, da man einfach zwischen einer Differenzengleichung und der Übertragungsfunktion hin- und her wechseln kann, aufgrund der Eigenschaften der Z-Transformation. Zum einen ist die Z Transformation linear, d.h. es gilt:

$$x[n] + y[n] \xleftrightarrow{Z} X(z) + Y(z)$$

Außerdem gilt die Verschiebungseigenschaft:

$$x[n - 1] \xleftrightarrow{Z} X(z)z^{-1}$$

$$x[n - 2] \xleftrightarrow{Z} X(z)z^{-2}$$

...

$$x[n - k] \xleftrightarrow{Z} X(z)z^{-k}$$

Mit diesen zwei Eigenschaften der Z-Transformation lassen sich Differenzengleichungen, mit denen Filter typischerweise beschrieben und implementiert werden, im Frequenzbereich beschreiben. Damit ist es das wichtigste Werkzeug, um Systeme analysieren zu können. (Kim, 2021).

Nehmen wir als Beispiel Gleichung (9), welche einen Lowpass Filter erster Ordnung beschreibt.

$$y[n] = (1 - a)x[n] + ay[n - 1]$$

Es gibt nur einen Filterkoeffizient a , das Eingangssignal $x[n]$ und das um ein Sample verzögerte Ausgangssignal $y[n - 1]$. Rein durch das Verändern des Koeffizienten, lässt sich die Grenzfrequenz dieses Filters bestimmen. Die Bestimmung der Koeffizienten kann recht kompliziert werden, weshalb es in dieser Arbeit nicht thematisiert wird.

Die Z-Transformation der Differenzengleichung lautet:

$$Y(z) = (1 - a)X(z) + az^{-1}Y(z)$$

Dabei wurde benutzt:

$$Z\{y[n - 1]\} = z^{-1}Y(z)$$

Durch Umstellen erhält man:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 - a}{1 - az^{-1}}$$

Wir erhalten also eine komplexe gebrochen-rationale Funktion, mit wie üblich Polen und Nullstellen. Sie lässt sich am besten auf der komplexen Ebene betrachten. Die Pole und Nullstellen beeinflussen den Frequenzgang des Filters, je nachdem wie nahe sie am Einheitskreis sind. Sind sie allerdings auf oder außerhalb des Einheitskreises, wird das System instabil. Im Audibereich entspricht das einer Feedbackschleife, die immer lauter wird, was man generell vermeiden möchte, vor allem im digitalen Bereich.

Eine Besonderheit der Z-Transformation ist, dass sich der Frequenzbereich nicht auf der X-Achse (bzw. auf der reellen Achse) befindet, sondern auf dem Einheitskreis. Und zwar erstreckt sich der Bereich von 0 Hz bis zur Nyquistfrequenz von 1 bis π auf der oberen Hälfte des Einheitskreises.

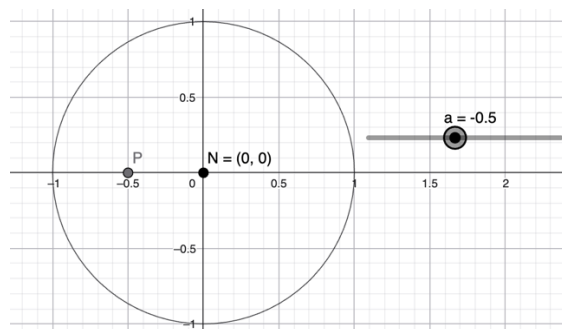


Abbildung 11: Pol- und Nullstellendiagramm
Quelle: Eigene Darstellung

Da in Abbildung 11 der Filterkoeffizient immer reell ist, befindet sich der Pol immer auf der reellen Achse. Je nachdem wo er sich befindet, werden eher hohe, mittlere oder tiefe Frequenzen beeinflusst. Außerdem muss in diesem Fall der Filterkoeffizient $|a| < 1$ gelten, da ansonsten der Pol außerhalb des Einheitskreises wandern würde.

IIR Filter sind in der Regel günstiger zu implementieren, dass sie pro Ausgangssample oft nur wenige Rechenoperationen durchführen müssen. Das macht sie für jegliche Anwendungen sehr attraktiv.

3.4 Biquad Filter

Biquad Filter sind mit die am häufigsten verwendeten Filter in der digitalen Audioproduktion. Mit ihnen lassen sich sehr günstig komplette parametrische Equalizer aufbauen, da sie alle typischen Filtertypen wie Lowpass, Highpass, Notch, Peak, etc. bilden können. Kombiniert, bzw. kaskadiert man sie, kann man einen parametrischen EQ mit beliebig vielen Filtern aufbauen. Der Name stammt aus der Übertragungsfunktion, welche wie folgt lautet:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (11)$$

Sowohl im Zähler, als auch im Nenner Polynom befindet sich eine quadratische Gleichung, deshalb bi-quad. Setzt man den Zähler gleich 0 erhält man die Nullstellen, setzt man den Nenner gleich null erhält man die Pole. Da es sich um zwei quadratische Gleichungen handelt, erhält man als Lösung ein konjugiert komplexes Paar, weshalb die Pole und Nullstellen auch auf der imaginären Achse wandern können, aber dadurch an der reellen Achse gespiegelt sind.

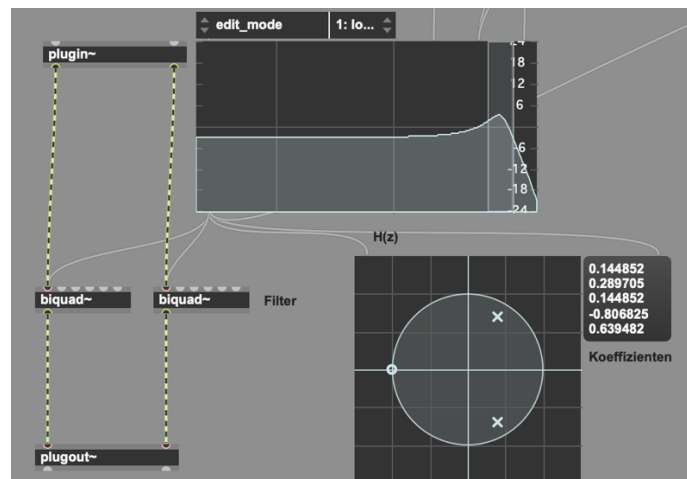


Abbildung 12: Max/MSP Biquad Implementierung
Quelle: Eigene Darstellung

Diese Abbildung zeigt zwei Biquad Filter in der Programmiersprache Max/MSP. Um die Grenzfrequenz, Filtertyp und Güte einzustellen, braucht man die Koeffizienten des Filters, welche scheinbar nicht viel mit den Zahlen zu tun haben, die man sonst verwendet (z.B. 500 Hz, und eine Güte von 1,4. Damit kann der Filter nicht viel anfangen). Diese Koeffizienten berechnet in Max ein Objekt namens ‚Filtergraph‘, und gibt diese als Liste aus. Wie man rechts auf der Abbildung erkennen kann, entziehen sich diese Koeffizienten jeder Intuition und müssen genau berechnet werden. Außerdem kann man das Pol- (durch „X“ gekennzeichnet) und Nullstellen- (durch „0“ gekennzeichnet) Diagramm sehen. Da der Filter schon etwas Resonanz zeigt, sind die Pole recht nah am Einheitskreis, was bedeutet, dass die dortige Frequenz angehoben werden.

Durch die Kaskadierung von mehreren Biquad Filtern lassen sich auch komplexere Filtertypen und ganze Equalizer aufbauen, was in der Praxis sehr üblich ist. Beispielsweise gibt es in Max/MSP ein Objekt namens ‚filterdesign‘ mit dem man auch Filter mit höherer Flankensteilheit und verschiedenen Topologien (Butterworth, Chebychev) realisieren kann. Filterdesign kommuniziert wiederum mit ein Objekt namens ‚cascade~‘ welches intern viele Biquadfilter kaskadiert.

4 Fazit und Aussichten

Durch das intensive Auseinandersetzen mit diesem Thema hat sich mein Verständnis und meine Intuition für Filter deutlich erhöht. Besonders faszinierend finde ich, dass auch komplexe Filter und Systeme im Grunde aus sehr einfachen Rechenoperationen bestehen, und nur geschickt miteinander kombiniert werden müssen, um den gewünschten Effekt zu erzielen. Ich möchte mich in Zukunft weiter mit der Implementierung in c++ auseinandersetzen und glaube, dass mich dieses Thema auch in meiner praktischen Arbeit als Ton-Ingenieur beeinflussen wird.

5 Quellen

KARRENBERG, Ulrich. *Signale—Prozesse—Systeme: Eine multimediale und interaktive Einführung in die Signalverarbeitung*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

KIM, Youngmoo. *Applied DSP No. 9: The z-Domain and Parametric Filter Design* [Online-Video]. YouTube, 23.08.2021. Verfügbar unter: https://www.youtube.com/watch?v=xIN5Mnj_MAk&t=682s (Zugriff: 27.07.2026).

MURTHY, Akash. 2. *Filter Characteristics - Digital Filter Basics* [Online-Video]. YouTube, 25.11.2022. Verfügbar unter: https://www.youtube.com/watch?v=autPNZ9XMa4&list=PLbqhA-NKGP6Afr_KbPUuy_yIBpPR4jzWo&index=2 (Zugriff: 28.07.2026).

SMITH III, Julius Orion. *Julius O. Smith III*. [online]. Stanford: Center for Computer Research in Music and Acoustics (CCRMA), Stanford University. Verfügbar unter: <https://ccrma.stanford.edu/~jos/>. [Zugriff am: 28.07.2026].

WEINZIERL, Stefan (Hg.). *Handbuch der Audiotechnik*. Springer Science & Business Media, 2008.