

Digitale Implementierung analoger Klangsyntheseverfahren

Diplomarbeit

durchgeführt von

Günter Trick

am
Institut für Elektronische Musik
an der
Universität für Musik und darstellende Kunst Graz
Vorstand: O.HProf. Dr. techn. Dipl.-Ing. Mag.art. Robert Höldrich

in Zusammenarbeit mit dem
Institut für Nachrichtentechnik und Wellenausbreitung
an der
Technischen Universität Graz
Vorstand: O.Univ.-Prof Dipl.-Ing. Dr.techn. Dr.phil. Willibald RIEDLER

Begutachter: O.Univ.-Prof Dipl.-Ing. Dr.techn. Dr.phil. Willibald Riedler

Betreuer: O.HProf. Dr. techn. Dipl.-Ing. Mag.art. Robert Höldrich

Graz, im Februar 1999

Für meine Eltern

Danksagung

Den größten Dank schulde ich meinem Vater und meiner Mutter. Dafür, daß sie die vielen Entbehrungen auf sich nahmen, um mir das Studium zu ermöglichen, und für die Ermutigung und Unterstützung.

Für die fachliche Unterstützung durch das Studium - insbesondere während der Diplomarbeit - bedanke ich mich im speziellen bei meinem Betreuer Herrn Dr. Robert Höldrich, sowie bei Herrn Dr. Gerhard Graber und Herrn Thomas Musil.

Vielen Dank auch an Peppo, Heidi und Christian für das Korrekturlesen und die schöne Studienzeit und meiner lieben Andrea für alles.

Zusammenfassung

Diese Diplomarbeit behandelt die *Digitale Implementierung analoger Klangsyntheseverfahren*. Dabei werden mittels Algorithmen der digitalen Signalverarbeitung Audiosignale generiert und modifiziert, so daß sie im wesentlichen den Klangeigenschaften ihrer Module aus der analogen Signalverarbeitung entsprechen. Die Arbeit befaßt sich mit der Problematik des Aliasings und des Quantisierungsrauschens, welche durch die Implementierung in der digitalen Ebene entsteht. Neben der Betrachtung dieser akustischen Qualitätsmerkmale erfolgt auch eine Untersuchung der Algorithmen hinsichtlich des Rechenaufwandes.

Zuerst wird ein Überblick über analoge Klangsyntheseverfahren und Module von Analogsynthesizern gegeben. Dabei wurden jene Klangsyntheseverfahren ausgewählt, die aufgrund ihrer Eigenschaften in der digitalen Realisierung besonders kritisch sind.

Der Schwerpunkt der Arbeit ist der Realisierung von Signalgeneratoren und Filtern gewidmet. Darüber hinaus wurden Rauschgenerator, Linear-Exponential-Konverter, Ringmodulator, Spannungsgesteuerter Verstärker, sowie Flanger und Phaser behandelt.

Aus den im Rahmen dieser Diplomarbeit vorgestellten Algorithmen wurden zwei Verfahren zur Realisierung eines Signalgenerators und eines Filters ausgewählt und als C-External auf der IRCAM Signal Processing Workstation implementiert.

Abstract

This thesis *Digital implementation of analog sound synthesis methods* is concerned with the development of signals and systems with analog characteristics by means of digital signal processing.

The study is about aliasing and quantization errors due to discrete time representation of signals and signal truncation. The algorithms are also analyzed with respect to their computation time.

A general view of analog sound synthesis is followed by a selection of special modules of analog synthesizers, which are particularly difficult to implement for digital applications.

This thesis deals primarily with the realization of signal oscillators and filters. Further topics are noise generator, linear-exponential-converter, ringmodulator, voltage controlled amplifier as well as flanger and phaser.

Two selected algorithms for oscillators and filters are implemented for the IRCAM signal processing workstation.

Inhaltsverzeichnis

1	Einleitung	4
2	Analoge Klangsyntheseverfahren	5
2.1	Überblick über analoge Klangsyntheseverfahren	5
2.1.1	Signalgeneratoren	5
2.1.1.1	Sinusgenerator	6
2.1.1.2	Rechteck-, Dreieck- und Sägezahngenerator	6
2.1.1.3	Rauschgenerator	6
2.1.2	Additive Synthese	7
2.1.3	Subtraktive Synthese	7
2.1.4	Nichtlineare Synthese	8
2.2	Module von Analogsynthesizer	9
2.2.1	Spannungsgesteuerter Oszillator (VCO)	9
2.2.2	Rauschgenerator	10
2.2.3	Linear-Exponential-Konverter	10
2.2.4	Ringmodulator	11
2.2.5	Spannungsgesteuerter Verstärker (VCA)	11
2.2.6	Spannungsgesteuertes Filter (VCF)	12
2.2.7	Flanger	12
2.2.8	Phaser	13
3	Algorithmen zur Implementierung analoger Klangsyntheseverfahren	15
3.0	Grundsätzliche Probleme der Analog-Digital-Umsetzung	15
3.0.1	Abtasttheorem und Aliasing	15
3.0.2	Auswirkungen der endlichen Rechengenauigkeit	18
3.0.2.1	Zahlendarstellung und Quantisierungsfehler	18
3.0.2.1.1	Zweierkomplement	18
3.0.2.1.2	Gleitkomma-Darstellung	19
3.0.2.2	Koeffizientenquantisierung	20
3.0.2.3	Quantisierungsrauschen	20
3.0.2.3.1	Quantisierungsrauschen bei Amplituden- gleichverteilung	20
3.0.2.3.2	Spektrum des Quantisierungsrauschens	21
3.0.2.4	Signal-/Rauschabstand realer Systeme	21
3.0.2.4.1	Signal-/Rauschabstand über Abtastfrequenz	22
3.0.2.4.2	Granularrauschen	22
3.0.2.5	Grenzyklen	22
3.1	Spannungsgesteuerter Oszillator (VCO)	23
3.1.1	Sinusgenerator	23
3.1.1.1	Direkte Berechnung von $\sin(\Theta n)$	23
3.1.1.1.0	Allgemeines	23
3.1.1.1.1	Berechnung einer Periode	25
3.1.1.1.2	Berücksichtigung der Halb- und Viertel- wellensymmetrie	26

3.1.1.1.3	Berücksichtigung der Komplementär- beziehung zwischen $\sin(\Theta n)$ und $\cos(\Theta n)$. . .	29
3.1.1.1.4	Frequenz- und Phasenmodulation	31
3.1.1.2	Wavetable-Algorithmen	32
3.1.1.2.0	Allgemeines	32
3.1.1.2.1	Abschneiden des Phasenwertes	34
3.1.1.2.2	Runden des Phasenwertes	35
3.1.1.2.3	Interpolationsmethoden	36
3.1.1.2.4	Berücksichtigung der Viertelwellen- symmetrie	43
3.1.1.2.5	Frequenz- und Phasenmodulation	43
3.1.1.3	Rekursive Algorithmen	44
3.1.1.3.1	Gekoppelte Form	44
3.1.1.3.2	Modifizierte gekoppelte Form	45
3.1.1.3.3	Digitaler Resonator	47
3.1.1.3.4	Digitaler Oszillator mittels Gitterstruktur . . .	47
3.1.2	Impuls-, Rechteck-, Dreieck- und Sägezahngeneratoren	49
3.1.2.0	Allgemeines	49
3.1.2.1	Additive Synthese im Zeitbereich	49
3.1.2.1.1	Additive Synthese mittels Sinusgeneratoren .	49
3.1.2.1.2	Wavetable-Synthese	50
3.1.2.1.3	Periodische Impulsfunktion	50
3.1.2.1.4	Tiefpaßgefilterte periodische Impulsfunktion	52
3.1.2.1.5	Erweiterungen zur Generierung von Recht- eck-, Dreieck- und Sägezahnfolgen	58
3.1.2.1.6	Approximation im Zeitbereich	61
3.1.2.2	Additive Synthese im Frequenzbereich	63
3.1.2.3	Subtraktive Synthese im Frequenzbereich	66
3.2	Rauschgenerator	69
3.2.1	Weißes Rauschen	69
3.2.1.1	Abtasten eines Zufallssignals.	69
3.2.1.2	Pseudo-Zufallszahlen-Generator	69
3.2.2	Farbiges Rauschen	70
3.2.2.1	Brownsches Rauschen	70
3.2.2.2	Rosa Rauschen	71
3.2.2.2.1	Approximation von rosa Rauschen	71
3.2.2.2.2	Rosa Rauschen mittels Modifikation im Frequenzbereich	72
3.3	Linear-Exponential-Konverter	73
3.4	Ringmodulator	75
3.4.1	Grundprinzip	75
3.4.2	Ringmodulator mit Halbbandfilter	76
3.4.2.1	Halbbandfilter	76
3.4.2.2	Algorithmus des Ringmodulators	78
3.5	Spannungsgesteuerter Verstärker (VCA)	79
3.6	Spannungsgesteuertes Filter (VCF)	80
3.6.0	Allgemeines	80
3.6.0.1	FIR-Filter	80
3.6.0.1.1	Rundungsfehler	81
3.6.0.1.2	Filtertopologien	81

3.6.0.2	IIR-Filter	82
3.6.0.2.1	Rundungsfehler	82
3.6.0.2.2	Skalierung	82
3.6.0.2.3	Filtertopologien	82
3.6.1	Elimination von Transienten	89
3.6.1.1	Elimination von Transienten bei FIR-Filtern	89
3.6.1.2	Elimination von Transienten bei IIR-Filtern	89
3.6.2	Tiefpaßfilter	91
3.6.2.1	Analyse eines analogen Tiefpaßfilters	91
3.6.2.2	Implementierungen des VCF	92
3.6.2.2.1	Entwurf mittels Analog-Digital-Transformationen	92
3.6.2.2.2	Entwurf von digitalen Filtern	99
3.6.3	Hochpaßfilter	102
3.6.4	Bandpaßfilter	103
3.6.5	Bandsperre	105
3.7	Flanger und Phaser	107
3.7.1	Flanger	107
3.7.1.1	Allgemeines	107
3.7.1.2	Zeitvariables Verzögerungselement	108
3.7.1.2.1	FIR-Delay-Filter	108
3.7.1.2.2	Allpaßfilter	109
3.7.1.2.3	Steueralgorithmus	109
3.7.2	Phaser	110
3.7.2.1	Allgemeines	110
3.7.2.2	Allpaßfilter	111
4	Realisierungen	112
4.0	IRCAM Musical Workstation und MAX	112
4.1	VCO-Realisierungen	113
4.1.1	Impulsgenerator	113
4.1.2	Bipolarer Impulsgenerator	116
4.1.3	Bipolarer Impulsgenerator mit Steuerung des Tastverhältnisses	116
4.1.4	Rechteckgenerator	116
4.1.5	Dreieckgenerator	116
4.1.6	Auswertung des VCO	117
4.2	Realisierungen des VCF	123
4.2.1	Moog-VCF	123
4.2.2	Auswertung des Moog-VCF	123
4.2.3	Zustandsvariablen-Filter	124
4.2.4	Auswertung des Zustandsvariablen-Filters	124
	Anhang	129
	Beschreibung der Hörbeispiele	129
	Literatur	131

1 Einleitung

Diese Diplomarbeit befaßt sich mit der digitalen Implementierung analoger Klangsyntheseverfahren. Dabei werden mittels Algorithmen der digitalen Signalverarbeitung Audiosignale generiert und modifiziert, so daß sie im wesentlichen den Klangmerkmalen ihrer Module aus der analogen Signalverarbeitung entsprechen. Die Aufgabenstellung umfaßte folgende Punkte:

- Auswahl und Einteilung analoger Klangsyntheseverfahren
- Literaturrecherche über digitale Signalverarbeitungsalgorithmen zur Implementierung analoger Klangsyntheseverfahren
- Gegenüberstellung der verschiedenen Algorithmen für das jeweilige analoge Klangsyntheseverfahren
- Implementierung einiger ausgewählter Algorithmen als C-Programm auf der IRCAM Signal Processing Workstation

Kapitel 2 gibt einen Überblick über ausgewählte analoge Klangsyntheseverfahren und Module von Analogsynthesizern, welche sich bei der Implementierung in der digitalen Ebene hinsichtlich Aliasing und Quantisierungsrauschen als besonders problematisch erweisen.

Kapitel 3 befaßt sich einleitend mit den grundsätzlichen Problemen beim Übergang von der analogen in die digitale Ebene. Schließlich werden die Algorithmen des jeweiligen Klangsyntheseverfahrens auf ihre Qualität und ihren Rechenaufwand hin analysiert und verglichen.

Kapitel 4 beschreibt die realisierten Algorithmen im Detail und analysiert die Ergebnisse.

Im Anhang befindet sich die Auflistung der Hörbeispiele der beiliegenden CD.

2 Analoge Klangsinteseverfahren

Klangsinteseverfahren dienen zur Generierung eines musikalischen Produktionsprozesses. Dabei wird ein Signal generiert, welches die gewünschten akustischen Eigenschaften besitzt.

Einen entscheidenden Schritt in der Entwicklung und Realisierung von analogen Klangsinteseverfahren stellten die von Robert Moog ab dem Jahre 1964 entwickelten spannungsgesteuerten Module dar. Neben der manuellen Einstellung von klangformenden Parametern war dies nun auch durch Steuerspannungen, welche wiederum von weiteren Modulen generierte Signale sein konnten, möglich geworden. Strukturen zur Klangerzeugung, die in wesentlichen Merkmalen übereinstimmen und sich meist nur in der Größe der verwendeten Parameter unterscheiden, werden in Sinteseverfahren eingeteilt.

Der erste Abschnitt dieses Kapitels gibt eine Übersicht über die wichtigsten Klangsinteseverfahren. Kapitel 2.2 stellt die Funktion der wichtigsten spannungsgesteuerten Module, wie sie als Komponenten in Analogsynthesizer Verwendung finden, vor. Die Literaturhinweise beziehen sich bereits auf die digitale Realisation der Verfahren, worauf in Kapitel 3 noch ausführlich eingegangen wird.

2.1 Überblick über analoge Klangsinteseverfahren

Die analoge Klangsintese läßt sich grundlegend in zwei Funktionsblöcke unterteilen:

- Signalerzeugung:

Oszillatoren erzeugen Signale, deren Parameter über Steuerspannungen bzw. manuell variiert werden können. Sie bilden die Quellen eines Klangsintesesystems.

- Signalverarbeitung:

Dabei wirken verschiedene Manipulationsmechanismen auf ein Signal und verändern dessen Klangeigenschaften.

Die Darstellung von Klangsintesesystemen erfolgt in Signalflußgraphen, wie sie in Computermusiksprachen (Music 4, Music 5, etc.) zur Illustration der Instrumenten- und Orchesterdefinitionen verwendet wurden. Die einzelnen Module werden als Symbole mit ihren spezifischen Ein- und Ausgängen dargestellt. Eine Verbindung zwischen Modulen wird durch eine Linie angezeigt. Einfache Operationen werden durch Kreise mit der jeweiligen mathematischen Operation im Inneren symbolisiert.

Kapitel 2.1.1 bis 2.1.5 erläutern verschiedene Verfahren von analogen Klangverarbeitungssystemen.

2.1.1 Signalgeneratoren

Oszillatoren liefern Signale zur weiteren Verarbeitung in der Klangsintese. Die in der analogen Klangsintese gebräuchlichsten Oszillatoren werden kurz vorgestellt.

2.1.1.1 Sinusgenerator

Sinus- oder Cosinusgeneratoren liefern als Ausgangssignal die entsprechenden trigonometrischen Funktionen. Als Eingangsparameter werden die Amplitude und die Frequenz ausgeführt. In digitalen Implementierungen wird auch die Phase als Eingangsparameter verwendet, um auch Phasenmodulation zu ermöglichen.

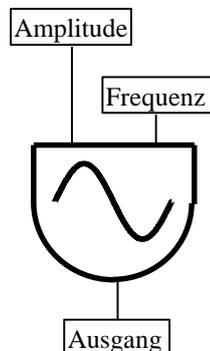


Bild 1: Signalflußsymbol eines Sinusgenerators.

2.1.1.2 Rechteck-, Dreieck- und Sägezahngenerator

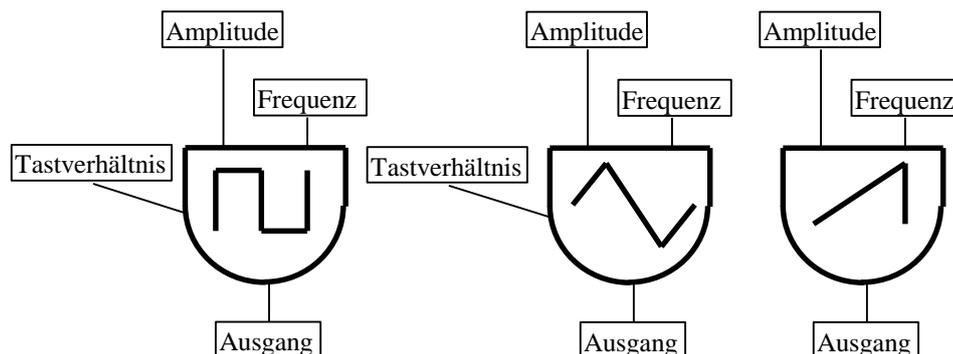


Bild 2: Signalflußsymbol von Rechteck-, Dreieck- und Sägezahngenerator.

Diese Generatoren können ebenfalls in Amplitude und Frequenz gesteuert werden. Zusätzlich kann bei einigen analogen Komponenten eine Einstellung des Tastverhältnisses manuell oder mittels Steuerspannung vorgenommen werden, wodurch sich die spektrale Zusammensetzung der Harmonischen bzw. die Klangfarbe variieren läßt.

2.1.1.3 Rauschgenerator

Neben den oben erwähnten deterministischen Signalen können durch Rauschgeneratoren auch stochastische Signale erzeugt werden. Ihre Charakteristik wird durch das Leistungsdichtespektrum beschrieben. Dabei wird bevorzugt weißes (frequenzunabhängige Charakteristik), rosa ($1/f$ -Charakteristik) oder Brownsches ($1/f^2$ -Charakteristik) Rauschen verwendet.

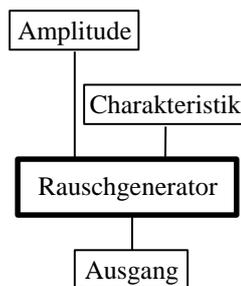


Bild 3: Signalflußsymbol eines Rauschgenerators.

2.1.2 Additive Synthese

Bei der additiven Synthese wird das gewünschte Signal durch Superposition der einzelnen Teiltöne erzeugt. Für jeden Ton wird ein eigener Oszillator verwendet, welcher in Amplitude, Frequenz und Phase beliebig angesteuert wird. Diese Methode ist vielseitig und ermöglicht auch die Synthese nichtstationärer Klänge. Als Nachteil ist der hohe technische Aufwand zu erwähnen, da für jede spektrale Komponente ein eigener Oszillator und dessen Steuersignale notwendig sind. Rauschhafte Anteile im Signal lassen sich aufgrund des dichten Spektrums nicht realisieren. Bild 4 illustriert das Grundprinzip der additiven Synthese.

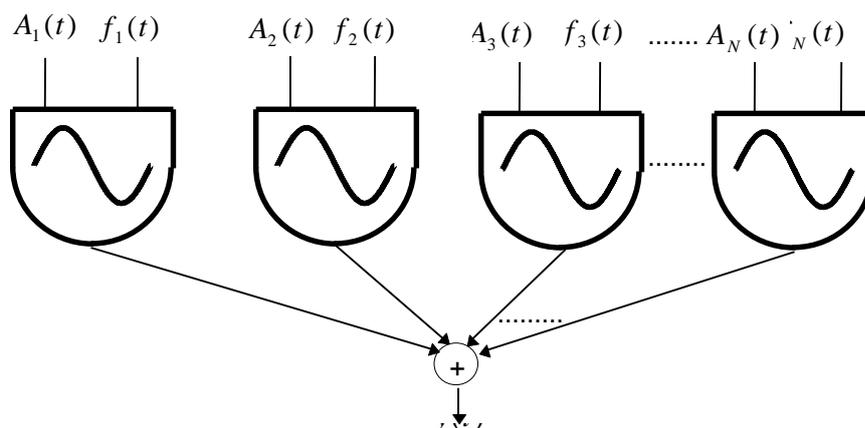


Bild 4: Prinzip der additiven Synthese.

2.1.3 Subtraktive Synthese

Bei der subtraktiven Synthese werden ausgewählte spektrale Komponenten eines Signals mittels Filtern bearbeitet und so das gewünschte Spektrum realisiert.

Es werden Generatoren mit obertonreichen Signalen wie Impulsfunktion, Sägezahn-schwingung, aber auch Rauschen verwendet.

Verschiedene Filter ermöglichen eine gezielte Bearbeitung des Spektrums. So lassen sich durch Filter mit konstanten Parametern Formanten nachbilden. Will man jedoch die Amplitude der Obertöne in konstantem Verhältnis zur Amplitude der Grundfrequenz erzeugen, so müssen auch die Frequenzparameter der Filter mit der Grundfrequenz gekoppelt mitgesteuert werden, was durch sogenannte spannungsgesteuerte Filter (VCF) realisiert werden kann.

Bild 5 zeigt ein Beispiel mit einem konstanten Filter und einem VCF.

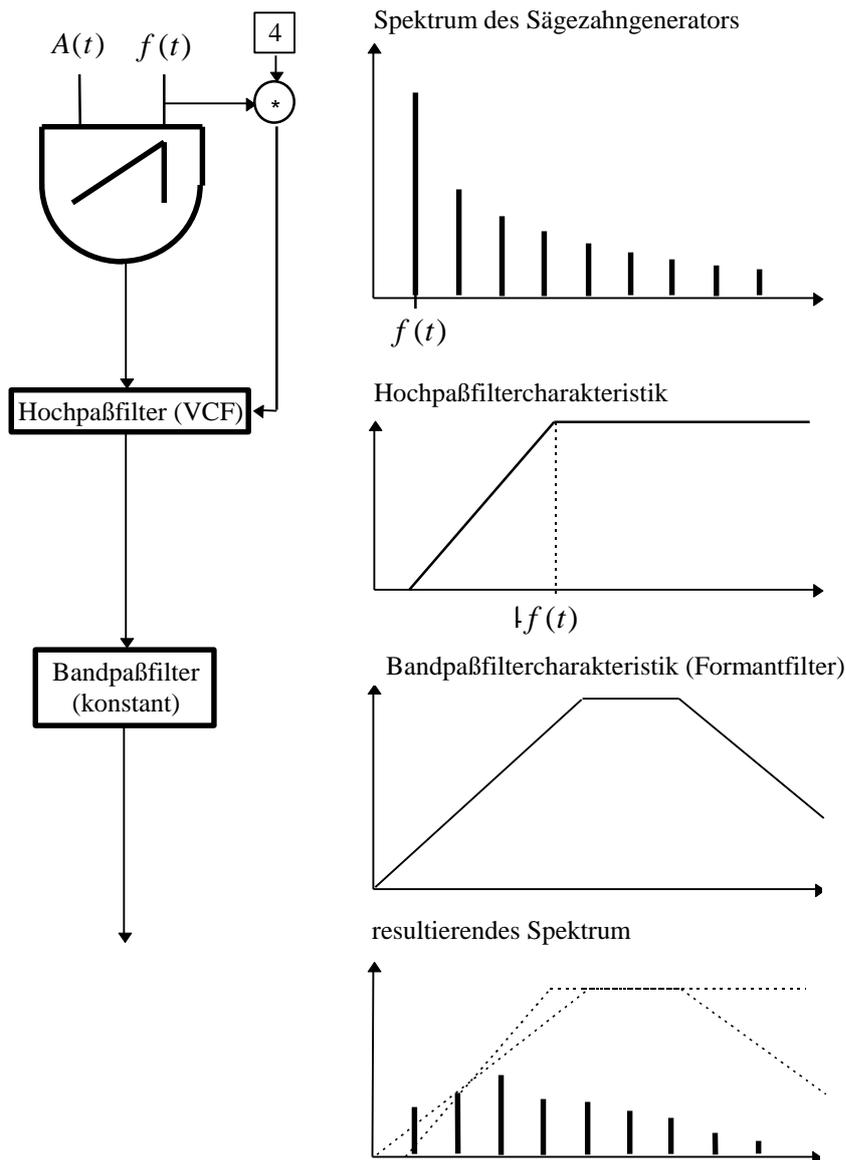


Bild 5: Beispiel für eine subtraktive Synthese mit einem variablen Hochpaßfilter (VCF) und einem zur Formantbildung verwendeten konstantem Bandpaßfilter sowie deren spektralen Charakteristiken.

2.1.4 Nichtlineare Synthese

Bei der nichtlinearen Synthese wird ein obertonreiches Spektrum durch Verwendung von klangformenden Modulen mit nichtlinearen Übertragungseigenschaften erzeugt. Der Vorteil liegt im geringen technischen Aufwand. Je nach verwendeter Struktur lassen sich durch Variation der Eingangsparameter verschiedene Klangmerkmale steuern.

Zu dieser Kategorie zählen unter anderem der Ringmodulator und die Frequenzmodulation, welche sich durch zwei Oszillatoren realisieren läßt, wobei ein Oszillator die Frequenz des zweiten steuert. Weiters sei die in Bild 6 dargestellte Waveshaping-Synthese erwähnt, welche mittels einer nichtlinearen Transferkennlinie neue Frequenzkomponenten erzeugt.

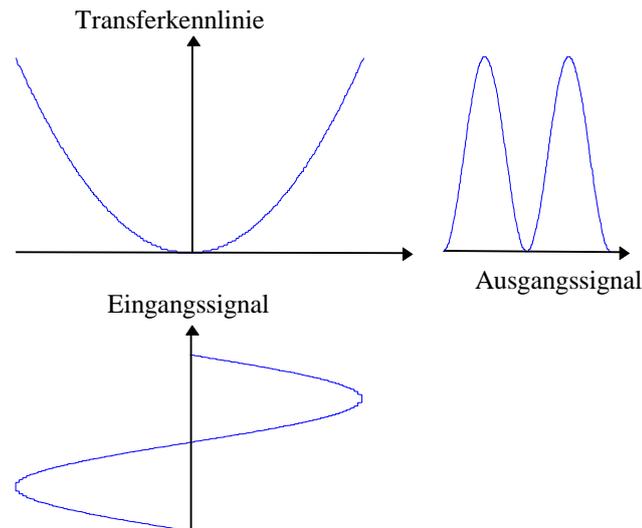


Bild 6: Prinzip der Waveshaping-Synthese.

2.2 Module von Analogsynthesizer

2.2.1 Spannungsgesteuerter Oszillator (VCO)

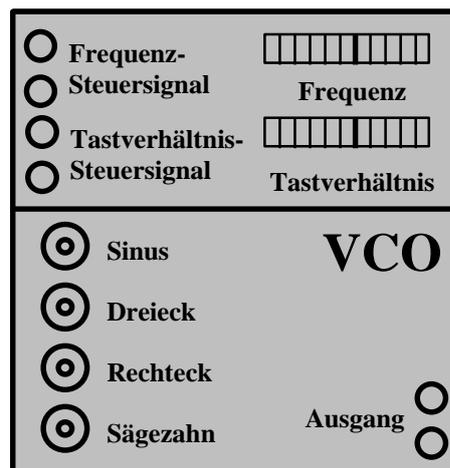


Bild 7: Ein- und Ausgangselement eines VCO-Moduls.

Bild 7 zeigt das typische Panel eines VCO-Moduls. Das Frequenz-Steuersignal wird einem Gleichanteil, welcher mit dem Regler "Frequenz" manuell eingestellt wird, hinzuaddiert und damit die Momentanfrequenz des Ausgangssignals generiert. Aufgrund der einfachen Realisierung kann nach dem gleichen Prinzip das Tastverhältnis beeinflusst werden, wodurch sich die Klangfarbe variieren läßt. Die gewünschte Wellenform des Oszillators wird über Taster eingestellt.

[1] und [2] stellen Implementierungen von Sinusgeneratoren mittels rekursiven Algorithmen vor. [3] untersucht den Signal-/Rauschabstand von Sinusgeneratoren, die mit der Table-Lookup-Methode arbeiten. [4] liefert die mathematischen Grundlagen der Spline-Interpolation, welche ebenfalls bei der Table-Lookup-Methode verwendet werden kann. [5] gibt einen Überblick über die Möglichkeiten der Sinusgenerierung und deren Anwendungen in der additiven Synthese.

[6] gibt einen Überblick über die Erzeugung von aliasingfreien Rechteck-, Dreieck- und Sägezahnsignalen. Im speziellen befassen sich [7] bis [10] mit der Generierung von aliasingfreien periodischen Impulssignalen. In [11] bis [13] werden die Signale im Frequenzbereich generiert und mittels inverser Fourier-Transformation in den Zeitbereich übergeführt. [14] stellt eine Approximationsmethode zur Erzeugung von Rechteck-, Dreieck- und Sägezahnsignalen vor.

2.2.2 Rauschgenerator

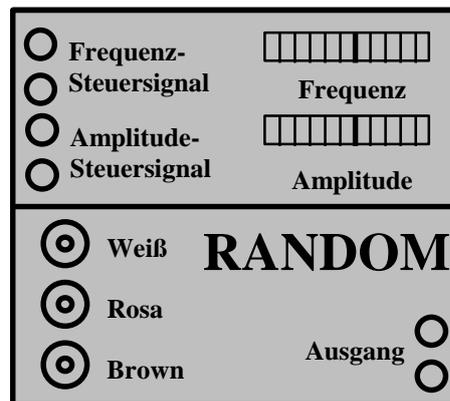


Bild 8: Ein- und Ausgangselement eines Rauschgenerator-Moduls.

Bild 8 zeigt ein Beispiel eines analogen Rauschgenerators. Das Modul bietet neben der Einstellung auf die gewünschte Charakteristik auch die Möglichkeit, das Ausgangssignal über zwei Steuersignale in Amplitude und Grenzfrequenz eines nachgeschalteten Tiefpaßfilters zu beeinflussen.

[15] beschreibt Möglichkeiten zur Generierung digitaler Zufallsfolgen, welche auch als Rauschsignale verwendet werden können.

2.2.3 Linear-Exponential-Konverter



Bild 9: Linear-Exponential-Konverter.

Linear-Exponential-Konverter dienen zur Umsetzung von Steuerspannungen in ein logarithmisches Maß, um bei der Ansteuerung eines Amplitudeneinganges nicht die Signalamplitude sondern direkt die Lautstärke zu beeinflussen. Ebenso wird bei der Ansteuerung eines Frequenzeinganges mit einem konvertiertem Signal die relative Tonhöhe beeinflusst und nicht die Frequenz. In Analogsynthesizer sind diese Module häufig schon in den entsprechenden Eingangsschaltungen integriert.

2.2.4 Ringmodulator



Bild 10: Ringmodulator.

Ringmodulatoren multiplizieren die beiden Eingangssignale miteinander. Im Gegensatz zu den VCA-Modulen (siehe unten) werden sie als 4-Quadranten-Multiplizierer ausgeführt. Da sie nicht zur Steuerung der Amplitude verwendet werden, ist die Möglichkeit der Einstellung einer Grundverstärkung (siehe Bild 10) nicht vorhanden.

Sie dienen zur Generierung neuer spektraler Komponenten, die sich durch die Faltung der beiden Eingangsspektren ergeben.

2.2.5 Spannungsgesteuerter Verstärker (VCA)

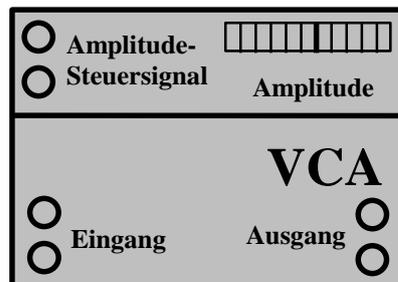


Bild 11: VCA-Modul mit einstellbarer Vorverstärkung.

VCA-Module ermöglichen neben der einfachen Verstärkung die zeitliche Steuerung der Amplitude eines Klages. Wird als Eingangssignal ein LFO-Signal (LFO = low-frequency-oscillator) verwendet, so ergibt sich ein Tremolo. Steuert jedoch ein Hüllkurvengenerator den VCA, so läßt sich die zeitliche Entwicklung der Amplitude mittels der Parameter des Hüllkurvengenerators (Attack, Decay, Sustain & Release) beeinflussen.

2.2.6 Spannungsgesteuertes Filter (VCF)

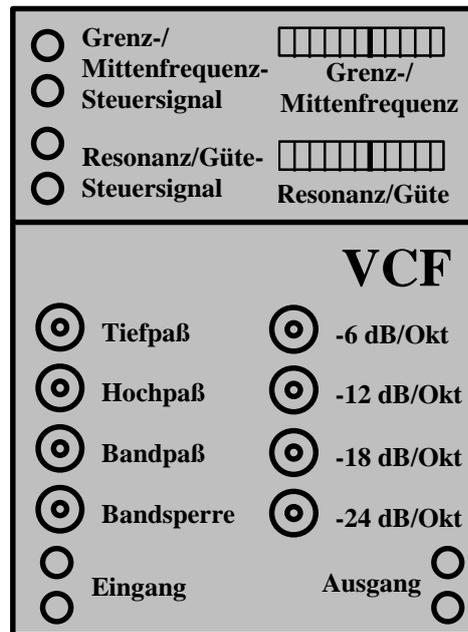


Bild 12: Panel eines spannungsgesteuerten Filters mit steuerbarer Grenz- bzw. Mittelfrequenz und Resonanz bzw. Güte sowie einstellbarer Charakteristik und Flankensteilheit.

VCF-Module dienen vorwiegend zur Realisierung der subtraktiven Synthese. Durch die Steuerung der Grenzfrequenz läßt sich diese mit der Grundfrequenz des Eingangssignals mitbewegen, wodurch die spektralen Komponenten im Verhältnis zueinander unverändert bleiben. Umgekehrt läßt sich bei einem stationären Eingangsspektrum durch Verändern der Filtercharakteristika die Klangfarbe variieren.

[16] bis [18] geben einen Überblick über digitale Filtertopologien und deren Eigenschaften. [19] erwähnt Möglichkeiten zur Elimination von Transienten, welche durch die abrupte Änderung der Filterkoeffizienten entstehen. [20] analysiert einen analogen VCF und erörtert digitale Realisierungsmöglichkeiten. [21] bis [23] befassen sich mit der Realisierung von digitalen Hochpaß-, Bandpaß- und Bandsperrenfilter.

2.2.7 Flanger

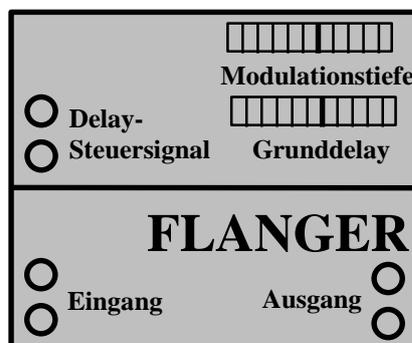


Bild 13: Flanger mit einstellbarer Modulationstiefe.

Flanger verzögern das Eingangssignal im Bereich von einigen Millisekunden und addieren dieses Signal mit dem unverzögerten Eingangssignal, wodurch sich ein Kammfiltereffekt ergibt. Die Verzögerungszeit kann über ein Steuersignal variiert werden, wodurch sich die Abstände der Frequenzminima des Kammfilters verändern. Die Stärke der Frequenzauslöschungen läßt sich über die Modulationstiefe, mit der das verzögerte Signal verstärkt wird, einstellen. Bild 14 zeigt die Übertragungsfunktion eines Flangers.

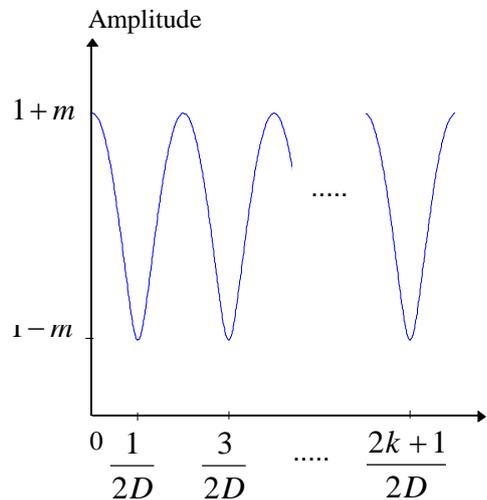


Bild 14: Übertragungsfunktion eines Flangers bei einer Verzögerung des Signals von D Sekunden und einer Modulationstiefe von m .

[24] erklärt Realisierungsmöglichkeiten des Flangers. [25] befaßt sich mit der Implementierung von variablen Verzögerungselementen.

2.2.8 Phaser

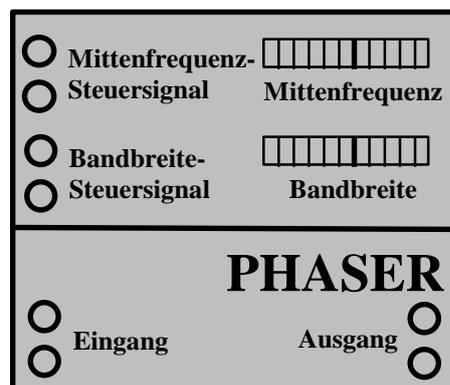


Bild 15: Modul eines Phasers.

Ähnlich wie beim Flanger kommt es zu Auslöschungen von bestimmten Frequenzen durch Addition des Eingangssignals mit dem zeitverzögerten Eingangssignal. Die Verzögerungszeit ist hier aber frequenzabhängig, was durch eine Kaskadenschaltung von mehreren Phasenschiebern realisiert wird. Je nach Anzahl der Phasenschieber ergeben sich mehrere Frequenzauslöschungen, die um eine Mittenfrequenz symmetrisch angeordnet sind. Über die Bandbreite läßt sich der Abstand der Auslöschungen zur Mittenfrequenz beeinflussen.

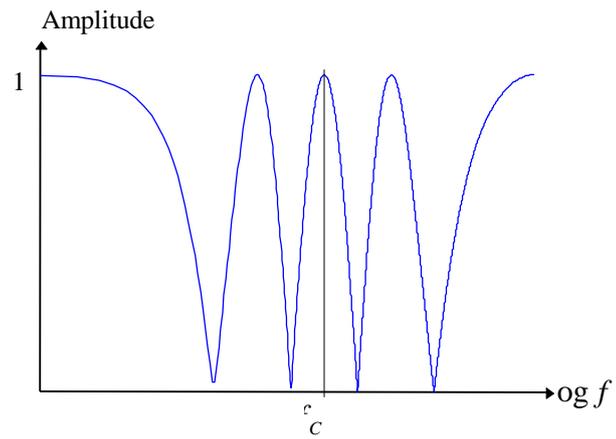


Bild 16: Übertragungsfunktion eines 8-stufigen Phasers.

[23] und [25] befassen sich mit der Realisierung von frequenzabhängigen Verzögerungselementen.

3 Algorithmen zur Implementierung analoger Klangsyntheseverfahren

3.0 Grundsätzliche Probleme der Analog-Digital-Umsetzung

3.0.1 Abtasttheorem und Aliasing

Die gebräuchlichste Variante, um aus einem zeitkontinuierlichen Signal eine zeitdiskrete Folge zu erhalten, ist die periodische Abtastung. Für die so gewonnene Folge $x[n]$ aus dem zeitkontinuierlichen Signal $x_c(t)$ gilt:

$$x[n] = x_c(nT_s) \quad \text{mit } -\infty < n < \infty$$

wobei: T_s ... Abtastperiode; Weiters wird $f_s = 1/T_s$ als die Abtastfrequenz oder Samplingfrequenz definiert.

Dieser Vorgang der Signalfolgengewinnung kann in zwei Schritte zerlegt werden:

- Multiplikation des Analogsignals mit einer periodischen Dirac-Impulsfunktion.
- Umwandlung der gewichteten Dirac-Impulse in eine zeitdiskrete Folge.

Bild 17 zeigt dieses Prinzip der periodischen Abtastung.

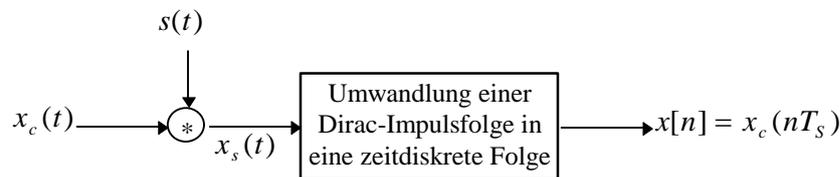


Bild 17: Prinzip der periodischen Abtastung zur Gewinnung einer zeitdiskreten Folge $x[n]$ aus einem zeitkontinuierlichen Signal $x_c(t)$.

zu a.) Multiplikation mit einer periodischen Dirac-Impulsfolge:

Das Analogsignal $x_c(t)$ wird mit einer periodischen Funktion von Dirac-Impulsen $s(t)$ multipliziert, wobei:

$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s)$$

Das resultierende Signal $x_s(t)$ lässt sich wie folgt anschreiben:

$$x_s(t) = x_c(t)s(t) = x_c(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s) = \sum_{n=-\infty}^{\infty} x_c(nT_s) \delta(t - nT_s)$$

Die Fourier-Transformierte der Funktion $x_s(t)$ setzt sich aus der Faltung der Fourier-Transformierten der Signale $s(t)$ und $x_c(t)$ zusammen.

$$\mathcal{F}\{s(t)\} = S(j\Omega) = \frac{2\pi}{T_s} \sum_{k=-\infty}^{\infty} \delta(\Omega - k\Omega_s) \quad \text{wobei} \quad \Omega_s = \frac{2\pi}{T_s}$$

$$\mathcal{F}\{x_c(t)\} = X_c(j\Omega)$$

Die Fourier-Transformierte von $x_s(t)$ ergibt sich zu

$$\mathcal{F}\{x_s(t)\} = X_s(j\Omega) = \mathcal{F}\{x_c(t)s(t)\} = \frac{1}{2\pi} X_c(j\Omega) * S(j\Omega) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X_c(j\Omega - k\Omega_s)$$

wobei "*" hier die Faltungsoperation kennzeichnet.

Die Multiplikation des Analogsignals mit der periodischen Dirac-Impulsfunktion führt im Frequenzbereich zu sich wiederholenden Kopien der Fourier-Transformierten von $x_c(t)$ im Abstand von ganzzahligen Vielfachen der Abtastfrequenz. Die Zusammenhänge im Zeit- und Frequenzbereich erläutert Bild 18.

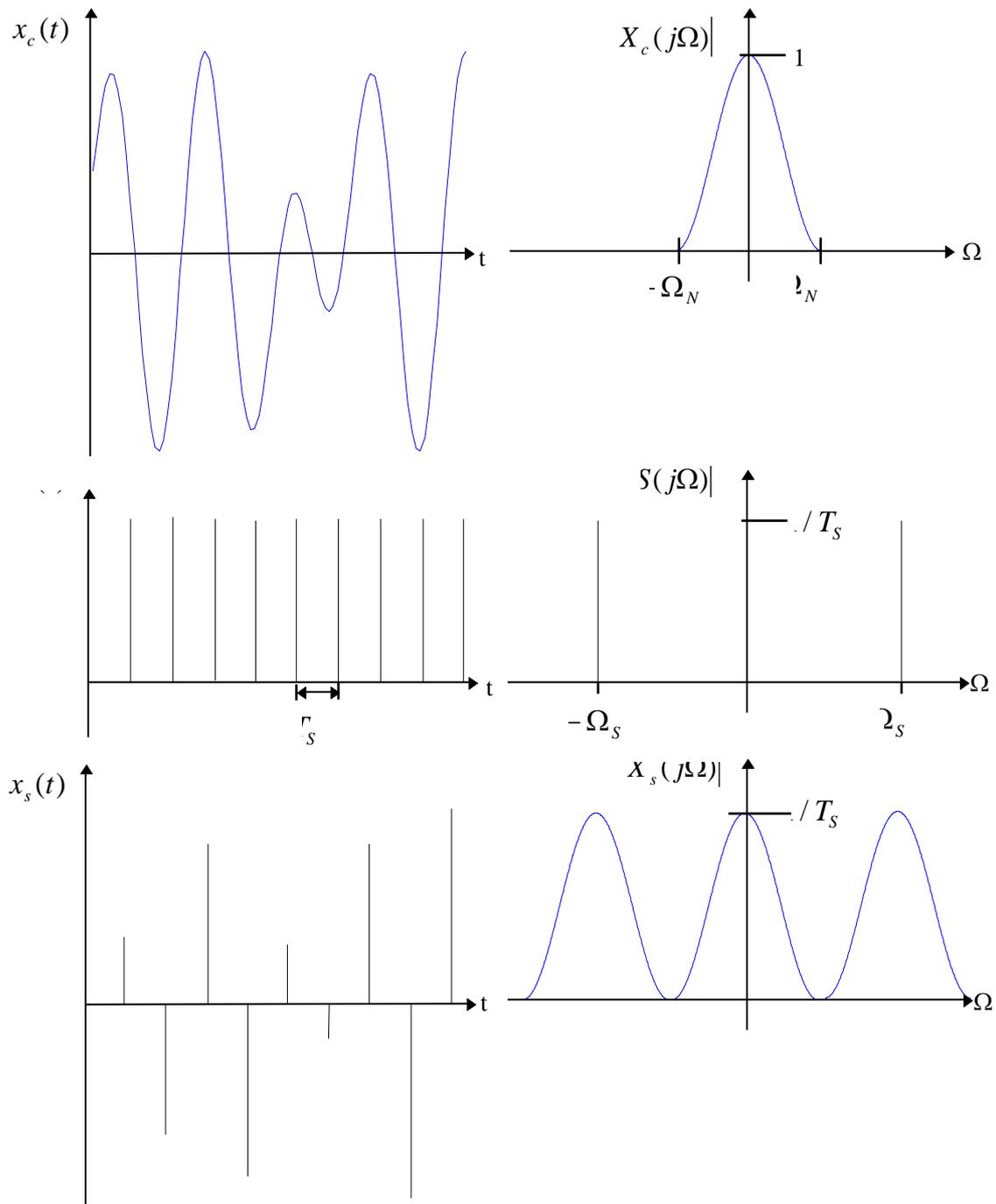


Bild 18: Multiplikation eines Analogsignals $x_c(t)$ mit einer periodischen Dirac-Impulsfunktion $s(t)$ und resultierendes Signal $x_s(t)$ im Zeit- und Frequenzbereich.

Wenn Ω_N den höchstfrequenten Anteil von $X_c(j\Omega)$ repräsentiert, so gilt unter der Bedingung $\Omega_s > 2\Omega_N$, daß sich die Kopien von $X_c(j\Omega)$ nicht überlappen. Sie treten als mit dem Faktor $1/T_s$ skalierte identische Kopien an jedem ganzzahligen Vielfachen von Ω_s auf, wodurch sich das ursprüngliche Analogsignal mit einem idealen Tiefpaßfilter aus $x_s(t)$ rekonstruieren läßt.

Diese Feststellung wird im Nyquist-Abtasttheorem formuliert:

$x_c(t)$ sei ein bandbegrenztetes Signal mit $X_c(j\Omega) = 0$ für $|\Omega| > \Omega_N$. Dann ist $x_c(t)$ eindeutig durch seine Abtastwerte $x[n] = x_c(nT)$, $n = 0, \pm 1, \pm 2, \dots$ bestimmt, wenn

$$\Omega_s > 2\Omega_N.$$

Wird die Abtastfrequenz zu klein gewählt, so kommt es im Frequenzbereich zur Überlappung der einzelnen Komponenten von $X_c(j\Omega)$, wodurch Frequenzanteile, die höher sind als $f_s/2$, um eben diese Frequenz gespiegelt werden. Dadurch läßt sich das Originalsignal nicht mehr durch einen Tiefpaß rekonstruieren. Statt dessen erfährt dieses Signal Verzerrungen, welche man als Aliasing bezeichnet.

zu b.) Umwandlung der gewichteten Dirac-Impulse in eine zeitdiskrete Folge:

Mit der Skalierung

$$\Theta = \Omega T_s = \frac{2\pi f}{f_s} \dots \text{digitale Frequenz}$$

wird der Frequenzbereich $0 < \Omega \leq \Omega_s$ auf den Bereich $0 < \Theta \leq 2\pi$ normiert. Dadurch wird auch die Zeitskala normiert, und zwar derart, daß der Abstand T_s zwischen zwei Abtastwerten von $x_s(t)$ die Dauer Eins zwischen zwei Abtastwerten von $x[n]$ ergibt.

3.0.2 Auswirkungen der endlichen Rechengenauigkeit

3.0.2.1 Zahlendarstellung und Quantisierungsfehler

In digitalen Signalverarbeitungssystemen ist es notwendig, die Signale und Koeffizienten in einem digitalen Zahlensystem darzustellen. Die meisten Rechenmaschinen verwenden ein binäres System. Verschiedene Formate wie Vorzeichen und Betrag, Einerkomplement, Zweierkomplement und Gleitkomma-Darstellung finden Verwendung, wobei das Zweierkomplement und die Gleitkomma-Darstellung am häufigsten angewendet werden.

3.0.2.1.1 Zweierkomplement

Die exakte Darstellung einer reellen Zahl durch das Zweierkomplement lautet:

$$x = X_m \left(-b_0 + \sum_{i=1}^{\infty} b_i 2^{-i} \right)$$

wobei: X_m beliebiger Skalierungsfaktor

b_i 0 oder 1

b_0 Vorzeichen-Bit:

$$b_0 = 0 \Rightarrow 0 \leq x \leq X_m$$

$$b_0 = 1 \Rightarrow -X_m \leq x < 0$$

Durch die Einschränkung auf $B + 1$ Binärstellen ergibt sich:

$$\hat{x} = Q_B[x] = X_m \left(-b_0 + \sum_{i=1}^B b_i 2^{-i} \right) = X_m \hat{x}_B$$

Dieser Wert von x ist quantisiert und liegt im Bereich $-X_m \leq \hat{x} < X_m$. Die geringste Differenz zwischen zwei Zahlen beträgt $\Delta = X_m 2^{-B}$. Die Quantisierung ist eine nichtlineare gedächtnislose Operation und führt zu einem Fehler, der als Quantisierungsfehler e_q bezeichnet wird und folgend definiert ist:

$$e_q = Q_B[x] - x$$

Durch das Runden einer Zweierkomplementzahl ist $-\Delta/2 < e_q \leq \Delta/2$; im Falle des Abschneidens ergibt sich: $-\Delta < e_q \leq 0$.

3.0.2.1.2 Gleitkomma-Darstellung

Eine Zahl x in Gleitkomma-Darstellung ist definiert als:

$$x = 2^c \hat{x}_M$$

wobei: c Exponent des Skalierungsfaktors (Exponent)

\hat{x}_M ..rein gebrochener Teil (Mantisse)

Sowohl der Exponent als auch die Mantisse werden explizit als binäre Festpunktzahlen dargestellt. Dadurch lassen sich hohe Darstellungsgenauigkeiten und große Aussteuerungsbereiche erzielen. Der Exponent wird angepaßt und die Mantisse normiert, so daß $0.5 \leq \hat{x}_M < 1$ gilt.

Beim Multiplizieren zweier Gleitkommazahlen werden die Exponenten addiert und die Mantissen multipliziert, wobei letztere quantisiert werden können. Auch beim Addieren müssen vor der eigentlichen Addition der Mantissen die Exponenten aneinander angepaßt werden, wodurch unter Umständen wiederum eine Quantisierung vorgenommen wird. Unter der Annahme, daß der Bereich des Exponenten ausreicht, um alle möglichen Werte von x zu skalieren, wirkt sich der Quantisierungsfehler nur auf die Mantissen aus, wobei dieser jedoch mit 2^c skaliert wird. Der Quantisierungsfehler ε einer Binärzahl in Gleitkomma-Darstellung wird als Bruchteil der Zahl x beschrieben, um eine Skalierung mit dem Signalpegel zu berücksichtigen:

$$\hat{x} = x(1 + \varepsilon) = x + \varepsilon x$$

3.0.2.2 Koeffizientenquantisierung

Beim Entwurf eines digitalen Systems wird im allgemeinen von einer speziellen Systemfunktion ausgegangen. Für diese Funktion werden die Systemkoeffizienten mit einer begrenzten Genauigkeit berechnet.

In IIR-Systemen (siehe auch Kapitel 3.6.0.2 und [26]) wirkt sich die Koeffizientenquantisierung dahingehend aus, daß sich die Pol- und Nullstellen in der z -Ebene auf neue Positionen verschieben, wodurch die so realisierte Systemfunktion von der originalen Systemfunktion abweicht. Im Extremfall kann das System auch instabil werden.

In FIR-Systemen (siehe auch Kapitel 3.6.0.1 und [26]) besteht zwischen der Impulsantwort und dem Quantisierungsfehler der Koeffizienten ein linearer Zusammenhang, wodurch sich die realisierte Systemfunktion als Summe einer unquantisierten Systemfunktion und einer Fehlersystemfunktion darstellen läßt.

3.0.2.3 Quantisierungsrauschen

3.0.2.3.1 Quantisierungsrauschen bei Amplitudengleichverteilung

Der Quantisierungsfehler führt zu einer Signalveränderung, die als Rauschen modelliert wird. Unter der Annahme, daß $n = 2^k$ Quantisierungsstufen gleich oft durchlaufen werden, läßt sich eine Aussage über das Signal-/Rauschleistungsverhältnis machen.

$$P_{vor} = \frac{1}{2X_m} \int_{-X_m}^{X_m} x^2 dx = \frac{X_m^2}{3} = n^2 \frac{\Delta^2}{12}$$

$$P_{nach} = \frac{1}{n} (1^2 + 3^2 + 5^2 + \dots + (n-1)^2) \frac{\Delta^2}{2} = (n^2 - 1) \frac{\Delta^2}{12}$$

Die Quantisierungsrauschleistung P_q ergibt sich als

$$P_q = P_{vor} - P_{nach} = n^2 \frac{\Delta^2}{12} - (n^2 - 1) \frac{\Delta^2}{12} = \frac{\Delta^2}{12}$$

und das Signal-/Rauschleistungsverhältnis (SNR) zu

$$SNR = \frac{S}{N} = \frac{P_{nach}}{P_q} = \frac{(n^2 - 1) \frac{\Delta^2}{12}}{\frac{\Delta^2}{12}} = (n^2 - 1)$$

bzw. in dB (auch Signal-/Rauschabstand genannt):

$$SNR(dB) = 10 \log(n^2 - 1) \approx 20 \log(n)$$

3.0.2.3.2 Spektrum des Quantisierungsrauschens

Unter der zusätzlichen Annahme, daß die Quantisierungsfehler verschiedener Abtastwerte nicht korrelieren und auch zwischen zu quantisierendem Signal und Quantisierungsfehler kein Zusammenhang besteht, was für genügend großes n erfüllt ist, läßt sich das Quantisierungsrauschen im Frequenzbereich untersuchen.

Die Autokorrelationsfunktion ist

$$R_x(\tau) = \lim_{\tau_0 \rightarrow \infty} \frac{1}{\tau_0} \int_{\tau=-\tau_0/2}^{\tau_0/2} x(t)x(t+\tau)dt = \begin{cases} \frac{\Delta^2}{12} \left(1 - \frac{|\tau|}{T}\right) & |\tau| < T_s \\ 0 & |\tau| > T_s \end{cases}$$

und mit

$$G_q(f) = \mathcal{F}\{R_x(\tau)\} = \frac{\Delta^2}{12} T_s \left(\frac{\sin(\pi f T_s)}{\pi f T_s}\right)^2 = \frac{\Delta^2}{12} T_s \text{sinc}^2(f T_s)$$

sowie der Näherung, daß $\text{sinc}^2(f T_s) \approx 1$ bis zur Frequenz $f_s / 2$ ist, erhält man:

$$G_q(f) = \frac{\Delta^2}{12 f_s}$$

Die Rauschleistung innerhalb des Nutzfrequenzbandes Δf beträgt dann:

$$P_q(f_s, \Delta f) = \int_{\Delta f} G_q(f) df = \frac{\Delta^2}{12} \frac{2\Delta f}{f_s} \tag{1}$$

3.0.2.4 Signal-/Rauschabstand realer Systeme

Im allgemeinen ist der Signal-/Rauschabstand auch von der Signalform und der Aussteuerung abhängig. Mit dem Modulationsfaktor m , wobei $m=0$ ein unmoduliertes und $m=1$ ein voll moduliertes Signal bedeutet, sowie dem Crest-Faktor

$$c = \frac{X_m}{X_{eff}}$$

läßt sich der Signal-/Rauschabstand wie folgt anschreiben:

$$\begin{aligned} SNR(m, c, k) &= 10 \log \frac{P_S}{P_q} = 10 \log \frac{X_{eff}^2 m^2}{\Delta^2 / 12} = 10 \log \left(m^2 \frac{3}{c^2} n^2\right) = \\ &= 20 \log m + 20 \log \frac{\sqrt{3}}{c} + 6.02k \end{aligned} \tag{2}$$

3.0.2.4.1 Signal-/Rauschabstand über Abtastfrequenz

Mit dem Oversamplingfaktor

$$L = \frac{f_s}{2\Delta f}$$

und den Ergebnissen aus den Gleichungen (1) und (2) läßt sich der Signal-/Rauschabstand zusammenfassen zu:

$$SNR(L, k, m, c) = 10 \log L + 20 \log m + 20 \log \frac{\sqrt{3}}{c} + 6.02k$$

3.0.2.4.2 Granularrauschen

Granularrauschen ist eine Intermodulation mit der Abtastfrequenz. Bei niedrigen Frequenzen und kleinen Signalwerten ist die Dekorrelation aufeinanderfolgender Abtastwerte nicht mehr gegeben, wodurch eine spektrale Veränderung des Quantisierungsrauschens auftritt.

3.0.2.5 Grenzyklen

Durch die endliche Genauigkeit der Arithmetik eines rekursiven Systems (IIR-System) ist es möglich, daß das Ausgangssignal trotz Nullansteuerung unbegrenzt weiterschwingt. Dieses Grenzyklusverhalten bei Nullansteuerung entsteht entweder durch nichtlineares Quantisieren in den Rückkopplungsschleifen des Systems oder durch Zahlenbereichsüberschreitungen bei der Addition (siehe [26]).

3.1 Spannungsgesteuerter Oszillator (VCO)

Im folgenden werden Algorithmen untersucht, welche zeitdiskrete, aliasingfreie Signale von analogen Oszillatoren, wie sie im Kapitel 2.2.1 vorgestellt wurden, zur Verfügung stellen.

Kapitel 3.1.1 befaßt sich mit der Realisierung von sinusförmigen Folgen. Die Algorithmen zur Erzeugung von Rechteck-, Dreieck- und Sägezahnswingungen werden gemeinsam in Kapitel 3.1.2 behandelt, da sie durch einfache Beziehungen miteinander verknüpft sind.

Dabei wird insbesondere der Signal-/Rauschabstand und der erforderliche Rechenaufwand untersucht. Zusätzlich werden die Beziehungen für die dynamische Frequenz- und Phasensteuerung angegeben.

3.1.1 Sinusgenerator

[1] unterscheidet drei Kategorien von Algorithmen zur Erzeugung von harmonischen Folgen:

- Direkte Berechnung von $\sin(\Theta n)$
- Wavetable-Algorithmen
- Rekursive Algorithmen

3.1.1.1 Direkte Berechnung von $\sin(\Theta n)$

3.1.1.1.0 Allgemeines

Werte von trigonometrischen Funktionen werden numerisch über folgende Reihenentwicklung berechnet:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!} \quad \text{für } |x| < \infty$$

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!} \quad \text{für } |x| < \infty$$

Die Reihe wird zur Berechnung nach dem N-ten Glied abgebrochen, was zu einem Fehler führt. Der Wert des Restgliedes R_N ist für den minimalen Signal-/Rauschabstand der Ausgangsfolge maßgebend und beträgt:

$$R_N = \sum_{n=N+1}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!} \quad \text{für } \sin(x)$$

$$R_N = \sum_{n=N+1}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!} \quad \text{für } \cos(x)$$

Es läßt sich feststellen, daß der Zähler um so langsamer ansteigt, je kleiner der Betrag des Argumentes ist. Diesen Zusammenhang am Beispiel der Sinusfunktion zeigt Bild 19.

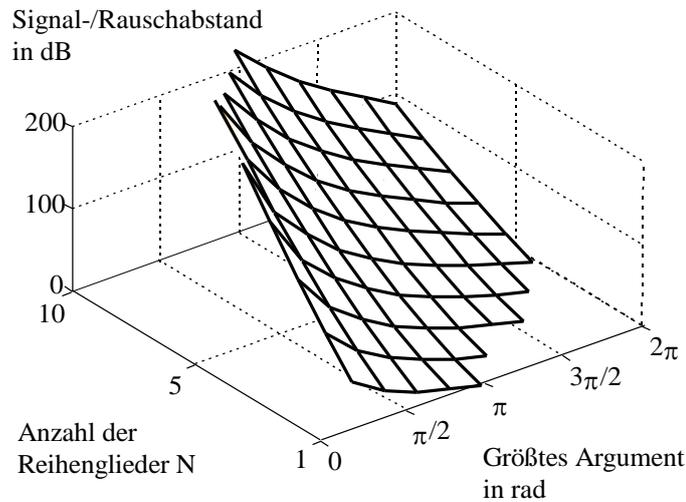


Bild 19: Signal-/Rauschabstand bei direkter Berechnung der Reihenentwicklung von $\sin(\Theta_n)$ und Abbruch nach N Gliedern mit zusätzlicher Einschränkung des Argumentes auf einen Maximalwert.

Die zur Berechnung verwendeten Reihen lassen sich wie folgt anschreiben:

$$\sin(x) \approx \sum_{n=0}^N (-1)^n \frac{x^{2n+1}}{(2n+1)!} = x \left(1 - x^2 \left(\frac{1}{6} - x^2 \left(\frac{1}{20} - x^2 \left(\frac{1}{42} - \dots \left(\frac{1}{(2N-1)!} - \frac{x^2}{(2N+1)!} \right) \right) \right) \right) \right)$$

$$\cos(x) \approx 1 + \sum_{n=1}^N (-1)^n \frac{x^{2n}}{(2n)!} = 1 - x^2 \left(\frac{1}{2} - x^2 \left(\frac{1}{24} - x^2 \left(\frac{1}{720} - \dots \left(\frac{1}{(2N-2)!} - \frac{x^2}{(2N)!} \right) \right) \right) \right)$$

In dieser Form ist der Rechenaufwand und der benötigte Speicherplatz ersichtlich und beträgt:

Für die Berechnung von $\sin(\Theta_n)$:

- $N+2$ Multiplikationen
- N Additionen
- $N+2$ Speicherplätze

Für die Berechnung von $\cos(\Theta_n)$:

- $N+1$ Multiplikationen
- N Additionen
- $N+1$ Speicherplätze

3.1.1.1.1 Berechnung einer Periode

Das Restglied der berechneten Folge läßt sich durch die Einschränkung des Argumentes auf kleine Werte reduzieren. Aufgrund der Periodizität der Sinus- und Cosinusfunktion genügt es, das Argument auf den Bereich von 0 bis 2π einzuschränken.

Die Zuweisung des Argumentes muß dann wie folgt vorgenommen werden:

$$\sin(\Theta n) = \sin(x) \quad \text{mit: } x = (\Theta n) \text{ modulo}(2\pi)$$

Der so erreichbare minimale Signal-/Rauschabstand als Funktion der Anzahl der für die Reihenberechnung berücksichtigten Glieder berechnet sich zu:

$$SNR_{\sin(\Theta n)} = 10 \log \left\{ \frac{\int_{x=0}^{2\pi} \left(\sum_{n=0}^N (-1)^n \frac{x^{2n+1}}{(2n+1)!} \right)^2 dx}{\int_{x=0}^{2\pi} \left(\sum_{n=0}^N (-1)^n \frac{x^{2n+1}}{(2n+1)!} - \sin(x) \right)^2 dx} \right\}$$

$$SNR_{\cos(\Theta n)} = 10 \log \left\{ \frac{\int_{x=0}^{2\pi} \left(\sum_{n=0}^N (-1)^n \frac{x^{2n}}{(2n)!} \right)^2 dx}{\int_{x=0}^{2\pi} \left(\sum_{n=0}^N (-1)^n \frac{x^{2n}}{(2n)!} - \cos(x) \right)^2 dx} \right\}$$

Tabelle 1 faßt die Ergebnisse zusammen.

Berücksichtigte Glieder (N)	$\frac{SNR_{\sin(\Theta n)}}{dB}$	$\frac{SNR_{\cos(\Theta n)}}{dB}$
7	33.0 (-20.6)	23.9 (-12.1)
8	52.0 (-39.2)	42.2 (-29.7)
9	72.8 (-59.6)	62.2 (-49.2)
10	95.2 (-81.6)	83.8 (-70.4)
11	119.1 (-105.2)	107.0 (-93.2)
12	144.4 (-130.1)	131.6 (-117.5)
13	170.9 (-156.3)	157.5 (-143.1)

Tabelle 1: Signal-/Rauschabstand (und größter relativer Fehler) in Dezibel bei direkter Berechnung der Reihenentwicklung von $\sin(\Theta n)$ bzw. $\cos(\Theta n)$ und Abbruch nach N Gliedern unter Berücksichtigung der Periodizität.

Ein Signal-/Rauschabstand von 96 dB würde die Berücksichtigung von 11 Gliedern zur Reihenberechnung erfordern und benötigt somit:

Für die Berechnung von $\sin(\Theta n)$:

- 23 Multiplikationen
- 11 Additionen
- 13 Speicherplätze

Für die Berechnung von $\cos(\Theta n)$:

- 22 Multiplikationen
- 11 Additionen
- 12 Speicherplätze

3.1.1.1.2 Berücksichtigung der Halb- und Viertelwellensymmetrie

Der Fehler läßt sich durch Berücksichtigen der Halb- und Viertelwellensymmetrie der Sinusfunktion weiter reduzieren. Daher genügt es, die Funktion nur für den Bereich von 0 bis π bzw. von 0 bis $\pi/2$ durch eine Reihe zu entwickeln und die Berücksichtigung des Argumentes wie folgt vorzunehmen:

Halbwellensymmetrie

$$\sin(\Theta n) = \begin{cases} \sin(x) \dots \dots \dots \text{für } 0 \leq x < \pi \\ -\sin(x - \pi) \dots \dots \text{für } \pi \leq x < 2\pi \end{cases}$$

bzw.

$$\cos(\Theta n) = \begin{cases} \cos(x) \dots \dots \dots \text{für } 0 \leq x < \pi \\ -\cos(x - \pi) \dots \dots \text{für } \pi \leq x < 2\pi \end{cases}$$

mit: $x = (\Theta n) \text{ modulo } (2\pi)$

Der Signal-/Rauschabstand ist nun:

$$SNR_{\sin(\Theta n)} = 10 \log \left\{ \frac{\int_{x=0}^{\pi} \left(\sum_{n=0}^N (-1)^n \frac{x^{2n+1}}{(2n+1)!} \right)^2 dx}{\int_{x=0}^{\pi} \left(\sum_{n=0}^N (-1)^n \frac{x^{2n+1}}{(2n+1)!} - \sin(x) \right)^2 dx} \right\}$$

$$SNR_{\cos(\Theta n)} = 10 \log \left\{ \frac{\int_{x=0}^{\pi} \left(\sum_{n=0}^N (-1)^n \frac{x^{2n}}{(2n)!} \right)^2 dx}{\int_{x=0}^{\pi} \left(\sum_{n=0}^N (-1)^n \frac{x^{2n}}{(2n)!} - \cos(x) \right)^2 dx} \right\}$$

Tabelle 2 faßt den erreichbaren Signal-/Rauschabstand für die Anzahl der berücksichtigten Glieder zusammen.

Berücksichtigte Glieder (N)	$\frac{SNR_{\sin(\Theta n)}}{dB}$	$\frac{SNR_{\cos(\Theta n)}}{dB}$
3	32.1 (-22.5)	23.0 (-13.5)
4	53.7 (-43.2)	42.5 (-32.4)
5	78.2 (-67.0)	65.6 (-54.8)
6	105.3 (-93.5)	91.5 (-80.0)
7	134.6 (-122.2)	119.7 (-107.6)
8	165.8 (-153.0)	145.0 (-137.4)

Tabelle 2: Signal-/Rauschabstand (und größter relativer Fehler) in Dezibel bei direkter Berechnung der Reihenentwicklung von $\sin(\Theta n)$ bzw. $\cos(\Theta n)$ und Abbruch nach N Gliedern unter Berücksichtigung der Halbwellensymmetrie.

CD-Qualität benötigt somit die Berücksichtigung von 6 Gliedern bei der Berechnung von $\sin(\Theta n)$ und 7 Gliedern bei der Berechnung von $\cos(\Theta n)$. Der Rechenaufwand und Speicherplatzbedarf beträgt:

Für die Berechnung von $\sin(\Theta n)$:

- 13 Multiplikationen
- 6 Additionen
- 8 Speicherplätze

Für die Berechnung von $\cos(\Theta n)$:

- 14 Multiplikationen
- 7 Additionen
- 8 Speicherplätze

Viertelwellensymmetrie

$$\sin(\Theta n) = \begin{cases} \sin(x) \dots\dots\dots \text{für } 0 \leq x < \pi/2 \\ \sin(\pi - x) \dots\dots\dots \text{für } \pi/2 \leq x < \pi \\ -\sin(x - \pi) \dots\dots\dots \text{für } \pi \leq x < 3\pi/2 \\ -\sin(2\pi - x) \dots\dots\dots \text{für } 3\pi/2 \leq x < 2\pi \end{cases}$$

bzw.

$$\cos(\Theta n) = \begin{cases} \cos(x) \dots\dots\dots \text{für } 0 \leq x < \pi/2 \\ -\cos(\pi - x) \dots\dots \text{für } \pi/2 \leq x < \pi \\ -\cos(x - \pi) \dots\dots \text{für } \pi \leq x < 3\pi/2 \\ \cos(2\pi - x) \dots\dots \text{für } 3\pi/2 \leq x < 2\pi \end{cases}$$

wobei: $x = (\Theta n) \text{ modulo}(2\pi)$

Der auf diese Weise erreichbare Signal-/Rauschabstand berechnet sich zu:

$$SNR_{\sin(\Theta n)} = 10 \log \left\{ \frac{\int_{x=0}^{\pi/2} \left(\sum_{n=0}^N (-1)^n \frac{x^{2n+1}}{(2n+1)!} \right)^2 dx}{\int_{x=0}^{\pi/2} \left(\sum_{n=0}^N (-1)^n \frac{x^{2n+1}}{(2n+1)!} - \sin(x) \right)^2 dx} \right\}$$

$$SNR_{\cos(\Theta n)} = 10 \log \left\{ \frac{\int_{x=0}^{\pi/2} \left(\sum_{n=0}^N (-1)^n \frac{x^{2n}}{(2n)!} \right)^2 dx}{\int_{x=0}^{\pi/2} \left(\sum_{n=0}^N (-1)^n \frac{x^{2n}}{(2n)!} - \cos(x) \right)^2 dx} \right\}$$

Die Ergebnisse sind in Tabelle 3 zusammengefaßt.

Berücksichtigte Glieder (N)	$\frac{SNR_{\sin(\Theta n)}}{dB}$	$\frac{SNR_{\cos(\Theta n)}}{dB}$
1	29.6 (-22.5)	18.8 (-12.6)
2	55.6 (-46.9)	42.0 (-34.0)
3	85.8 (-76.1)	70.2 (-61.0)
4	119.6 (-109.0)	102.3 (-92.1)
5	156.2 (-145.0)	137.6 (-126.7)
6	195.4 (-183.6)	175.5 (-164.0)

Tabelle 3: Signal-/Rauschabstand (und größter relativer Fehler) in Dezibel bei direkter Berechnung der Reihenentwicklung von $\sin(\Theta n)$ bzw. $\cos(\Theta n)$ und Abbruch nach N Gliedern unter Berücksichtigung der Viertelwellensymmetrie.

Ein Signal-/Rauschabstand von 96 dB läßt sich also unter Berücksichtigung von 4 Gliedern erreichen.

Der Rechenaufwand und Speicherplatzbedarf beträgt also:

für $\sin(\Theta n)$:

- 9 Multiplikationen
- 4 Additionen
- 6 Speicherplätze

für $\cos(\Theta n)$:

- 8 Multiplikationen
- 4 Additionen
- 5 Speicherplätze

Durch die Argumentzuweisung bei Halb- und Viertelwellensymmetrie entsteht ein zusätzlicher Rechenaufwand, der folgende Operationen notwendig macht:

- **Bedingungsanweisungen:**
Zuerst muß festgestellt werden, in welchem Quadranten das Argument liegt, um den resultierenden Wert des Argumentes berechnen zu können. Dies benötigt eine if-Anweisung bei Halb- und zwei if-Anweisungen bei Viertelwellensymmetrie.
- **Berechnung des aktuellen Argumentes:**
Die Berechnung des resultierenden Argumentes benötigt eine Addition.
- **Vorzeichenzuweisung:**
Dem Ergebnis wird entsprechend dem Argument ein Vorzeichen zugewiesen.

3.1.1.1.3 Berücksichtigung der Komplementbeziehung zwischen $\sin(\Theta n)$ und $\cos(\Theta n)$

Eine weitere Möglichkeit, den Rechenaufwand zu verringern, besteht durch die Einschränkung des Argumentes auf $0 \leq x < \pi/4$ unter der Berücksichtigung der Beziehungen

$$\sin(x) = \begin{cases} \sin(x) \dots\dots\dots 0 \leq x < \pi/4 \\ \cos(\pi/2 - x) \dots\dots\dots \pi/4 \leq x < \pi/2 \end{cases}$$

und

$$\cos(x) = \begin{cases} \cos(x) \dots\dots\dots 0 \leq x < \pi/4 \\ \sin(\pi/2 - x) \dots\dots\dots \pi/4 \leq x < \pi/2 \end{cases}$$

Hier wird wiederum die Viertelwellensymmetrie ausgenutzt, wobei zusätzlich für $\sin(x)$ und Argumentwerte von $\pi/4$ bis $\pi/2$ die Cosinusreihe von 0 bis $\pi/4$ mit dem resultierenden Argument berechnet wird. Entsprechend wird für $\cos(x)$ und Argumentwerte von $\pi/4$ bis $\pi/2$ die Sinusreihe von 0 bis $\pi/4$ verwendet.

Das Argument muß folgend interpretiert und zugewiesen werden:

$$\sin(\Theta n) = \begin{cases} \sin(x) \dots\dots\dots \text{für } 0 \leq x < \pi/4 \\ \cos(\pi/2 - x) \dots\dots \text{für } \pi/4 \leq x < \pi/2 \\ \cos(x - \pi/2) \dots\dots \text{für } \pi/2 \leq x < 3\pi/4 \\ \sin(\pi - x) \dots\dots\dots \text{für } 3\pi/4 \leq x < \pi \\ -\sin(x - \pi) \dots\dots\dots \text{für } \pi \leq x < 5\pi/4 \\ -\cos(3\pi/2 - x) \dots\dots \text{für } 5\pi/4 \leq x < 3\pi/2 \\ -\cos(x - 3\pi/2) \dots\dots \text{für } 3\pi/2 \leq x < 7\pi/4 \\ -\sin(2\pi - x) \dots\dots\dots \text{für } 7\pi/4 \leq x < 2\pi \end{cases}$$

und

$$\cos(\Theta n) = \begin{cases} \cos(x) \dots\dots\dots \text{für } 0 \leq x < \pi/4 \\ \sin(\pi/2 - x) \dots\dots\dots \text{für } \pi/4 \leq x < \pi/2 \\ -\sin(x - \pi/2) \dots\dots\dots \text{für } \pi/2 \leq x < 3\pi/4 \\ -\cos(\pi - x) \dots\dots\dots \text{für } 3\pi/4 \leq x < \pi \\ -\cos(x - \pi) \dots\dots\dots \text{für } \pi \leq x < 5\pi/4 \\ -\sin(3\pi/2 - x) \dots\dots\dots \text{für } 5\pi/4 \leq x < 3\pi/2 \\ \sin(x - 3\pi/2) \dots\dots\dots \text{für } 3\pi/2 \leq x < 7\pi/4 \\ \cos(2\pi - x) \dots\dots\dots \text{für } 7\pi/4 \leq x < 2\pi \end{cases}$$

wobei wiederum $x = (\Theta n) \text{ modulo } (2\pi)$ gilt.

Der Signal-/Rauschabstand

$$SNR_{\sin(\Theta n)} = 10 \log \left\{ \frac{\int_{x=0}^{\pi/4} \left(\sum_{n=0}^N (-1)^n \frac{x^{2n+1}}{(2n+1)!} \right)^2 dx}{\int_{x=0}^{\pi/4} \left(\sum_{n=0}^N (-1)^n \frac{x^{2n+1}}{(2n+1)!} - \sin(x) \right)^2 dx} \right\}$$

$$SNR_{\cos(\Theta n)} = 10 \log \left\{ \frac{\int_{x=0}^{\pi/4} \left(\sum_{n=0}^N (-1)^n \frac{x^{2n}}{(2n)!} \right)^2 dx}{\int_{x=0}^{\pi/4} \left(\sum_{n=0}^N (-1)^n \frac{x^{2n}}{(2n)!} - \cos(x) \right)^2 dx} \right\}$$

ergibt folgende in Tabelle 4 zusammengefaßten Werte:

Berücksichtigte Glieder (N)	$\frac{SNR_{\sin(\Theta n)}}{dB}$	$\frac{SNR_{\cos(\Theta n)}}{dB}$
1	55.2 (-52.2)	44.8 (-36.2)
2	93.1 (-88.8)	80.1 (-69.8)
3	135.5 (-130.1)	120.3 (-109.0)
4	181.3 (-175.1)	164.5 (-152.2)

Tabelle 4: Signal-/Rauschabstand (und größter relativer Fehler) in Dezibel bei direkter Berechnung der Reihenentwicklung von $\sin(\Theta n)$ bzw. $\cos(\Theta n)$ und Abbruch nach N Gliedern bei einem Argument kleiner $\pi/4$.

Der Rechenaufwand und Speicherplatzbedarf beträgt hier bei der Berücksichtigung von 3 Gliedern:

- 3 Bedingungsanweisungen zur Argumentzuweisung
- 7 Multiplikationen bei der Sinusreihenberechnung
6 Multiplikationen bei der Cosinusreihenberechnung
- 1 Addition zur Berechnung des resultierenden Argumentes
3 Additionen bei der Reihenberechnung
- 8 Speicherplätze, da die Konstanten der Sinus- und Cosinusreihe benötigt werden

3.1.1.1.4 Frequenz- und Phasenmodulation

Bei dynamischer Ansteuerung mit der Frequenz $f_1[n]$ und Phase $\Psi[n]$ ergibt sich das Argument zu

$$x = \Theta[n]n + \Psi[n] = \frac{2\pi f_1[n]}{f_s} n + \Psi[n]$$

was zu einem zusätzlichen Rechenaufwand von

- 1 Multiplikation
- 1 Addition

führt. Die Genauigkeit der Darstellung des Argumentes legt auch die Frequenzauflösung fest. Wird der Bereich von $0 \leq x < 2\pi$ in q Elemente unterteilt, so wird damit auch der Frequenzbereich von $0 \leq f_1 < f_s/2$ in q Elemente aufgelöst, was zu einer Frequenzauflösung von $f_\Delta = f_s/2q$ führt.

3.1.1.2 Wavetable-Algorithmen

3.1.1.2.0 Allgemeines

Ein weiteres Verfahren zur Erzeugung von Sinusfolgen ist die sogenannte Table-Lookup-Methode. Dabei werden mit beliebiger Genauigkeit berechnete Sinuswerte in ein Speicherfeld geschrieben, welches ausgelesen wird. Es wird wiederum nur der Hauptwert des Argumentes verwendet, welcher nun - abgesehen vom Rundungsfehler - direkt proportional der zum Speicherfeldbeginn relativen Adresse ist. Dabei wird ein Inkrementwert I , welcher sich aus der gewünschten Frequenz ableiten läßt, dem letzten Adreßwert im sog. Phasenregister hinzuaddiert und so die aktuelle Speicheradresse berechnet. Bild 20 erläutert das Prinzip.

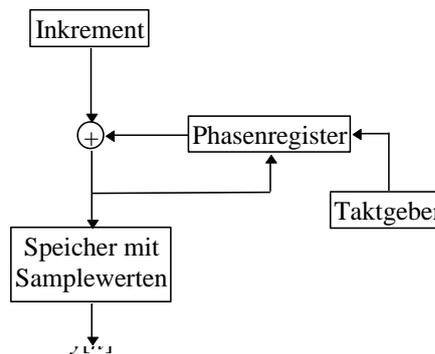


Bild 20: Prinzip der Table-Lookup-Methode.

Der Inkrementwert wird vorzeichenbehaftet interpretiert, um auch negative Frequenzen für die additive Synthese zu ermöglichen. Durch eine Abstimmung der Phasenregistergröße auf die Speicherfeldgröße L läßt sich die Modulo-Arithmetik zur Erzeugung des Hauptwertes automatisch durch Ignorieren des Registerüberlaufes realisieren. Dabei muß die Speicherfeldgröße auf $L = 2^m$ Samplewerte festgelegt werden. Vom errechneten Phasenwert werden m Bit zur Adressierung des Speicherfeldes übernommen. Dabei gibt es drei Möglichkeiten zur Funktionswertgewinnung (siehe [3]).

- Abschneiden des Phasenwertes auf m Bit
- Runden des Phasenwertes auf m Bit
- Genauen Phasenwert zur Interpolation verwenden

Die Ausgangsfrequenz f_1 ist gegeben durch:

$$f_1 = \frac{If_s}{L} = If_s 2^{-m}$$

Der benötigte Inkrementwert beträgt also:

$$I = \frac{f_1 L}{f_s} = \frac{f_1 2^m}{f_s}$$

Im folgenden wird der Einfluß der Speicherfeldgröße und die Methode der Funktionswertgewinnung auf den Signal-/Rauschabstand des Ausgangssignales untersucht. Zusätzlich wird die resultierende Frequenzauflösung betrachtet.

Maximale Speicherfeldgröße

Der Signal-/Rauschabstand wird durch die Anzahl der für die Darstellung des jeweiligen Wertes verwendeten Bit festgelegt.

Der Signal-/Rauschabstand ändert sich nicht, wenn die Funktionswerte der Tabelle dem tatsächlichen Wert - mit der jeweiligen Auflösung quantisiert - entsprechen. In diesem Fall unterscheiden sich benachbarte Tabellenwerte um nicht mehr als 1 LSB. Die größte Signaländerung in der Sinusfunktion tritt beim Nulldurchgang auf. Die beiden Tabellenwerte um diese Stelle dürfen sich nur um 1 LSB unterscheiden, um eine Verminderung des Signal-/Rauschabstandes zu verhindern. Bild 21 erläutert den Zusammenhang zwischen Bitauflösung k und benötigtem Speicherplatz L . Tabelle 5 enthält einige Ergebnisse.

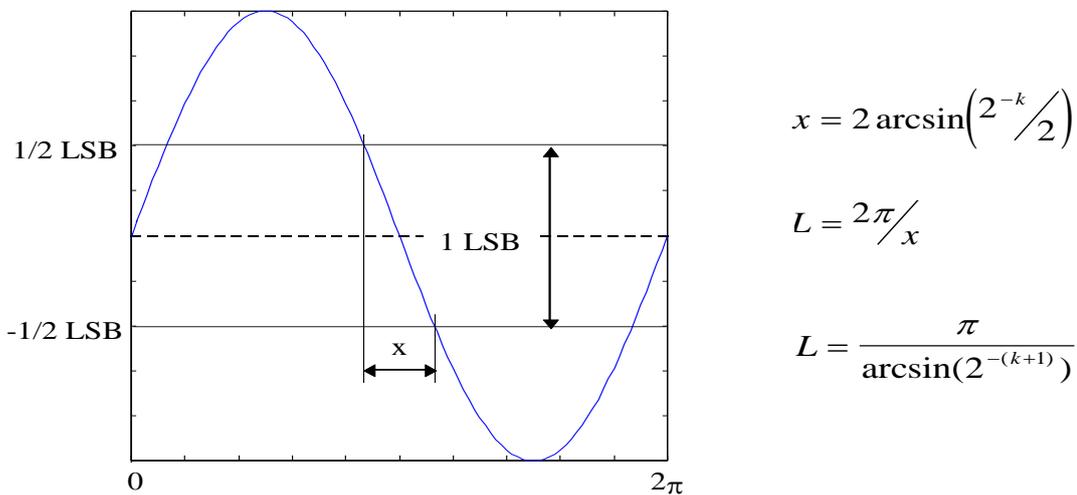


Bild 21: Zusammenhang zwischen Bitauflösung k und benötigtem Speicherplatz L .

Bitauflösung k	Anzahl der benötigten Speicherplätze L	Benötigter Speicher in Byte
2	26	7
4	101	51
8	1609	1609
10	6434	8043
12	25,73 k	38,60 k
14	102,94 k	180,15 k
16	411,77 k	823,55 k
20	6,588 M	16,471 M
24	105,41 M	316,24 M

Tabelle 5: Benötigte Anzahl von Speicherplätzen L bzw. Speicherfeldgröße in Byte bei Zahlendarstellung mit k Bit, um optimalen Signal-/Rauschabstand zu gewährleisten.

Ein größeres Speicherfeld würde keinen verbesserten Signal-/Rauschabstand zufolge haben, jedoch müßte - um wiederum die Modulo-Arithmetik zu gewährleisten - das Feld auf den nächsthöheren Wert von $L = 2^m$ erweitert werden, was beispielsweise für die Auflösung mit 16 Bit den Speicherplatz auf 528288 Tabellenwerte bzw. 1MByte erhöht.

Andererseits stellt dieser Wert auch die obere Grenze dar, mit der die Argumentwerte quantisiert werden sollen, wodurch auch die maximale Größe des Phasenregisters festgelegt wird.

3.1.1.2.1 Abschneiden des Phasenwertes

Dabei wird das Argument x mit $0 \leq x < 2\pi$ durch Abschneiden auf den Speicherplatz S mit $0 \leq S < L-1$ umgerechnet, was wiederum ein Quantisierungsvorgang ist. Der dadurch entstehende Fehler macht sich als Rauschen bemerkbar und wird als "phase jitter" bezeichnet, da die resultierende Sinusfunktion die Argumentwerte sprunghaft durchquert. Der Signal-/Rauschabstand wird nun von der Auflösung der Funktionswerte in k Bit und von der Anzahl der Speicherwerte L bestimmt. Diesen Zusammenhang veranschaulicht Bild 22.

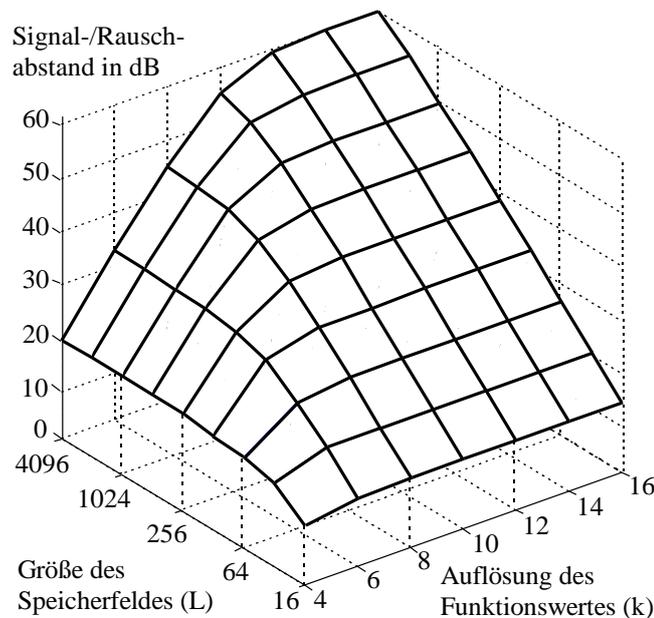


Bild 22: Signal-/Rauschabstand in Abhängigkeit von der Funktionswertauflösung k und der Größe des Speicherfeldes L sowie Quantisierung des Argumentes auf L Werte.

Man erkennt, daß ab einem bestimmten Wert für k und L keine wesentliche Steigerung des Signal-/Rauschabstandes mehr möglich ist. [3] formuliert diese Feststellung wie folgt:

Einen annähernd optimalen Signal-/Rauschabstand von

$$SNR_{\substack{\text{Wavetable} \\ \text{abschneiden}}} = 6(e - 2) \quad [dB]$$

erhält man bei einer Funktionswertquantisierung auf $e+1$ Bit und einer Speicherfeldgröße von $L = 2^e$.

Die Berechnung des Signal-/Rauschabstandes nach der Formel

$$SNR_{\text{Wavetable abscheiden}} = 10 \log \left\{ \frac{\sum_{n=1}^L \left\{ \left(Q_{k-1} \left[\sin \left(\frac{2\pi}{L} (n-0.5) \right) \right] \frac{2\pi}{L} \right)^2 \right\}}{\sum_{n=1}^L \left\{ \int_{x=\frac{2\pi}{L}(n-0.5)}^{\frac{2\pi}{L}(n+0.5)} \left(Q_{k-1} \left[\sin \left(\frac{2\pi}{L} (n-0.5) \right) \right] - \sin \left(x + \frac{2\pi}{L} (n-1) \right) \right)^2 dx \right\}} \right\}$$

wobei $Q_{k-1}[\dots]$ die Quantisierung auf k Bit (inklusive Vorzeichenbit) bedeutet, ergibt die in Tabelle 6 erfaßten Werte.

$SNR_{\text{Wavetable abscheiden}}$ dB		Funktionswertlänge in k Bit						
		8	10	12	14	16	20	24
Speicherfeldgröße L	256	35.9 (-29.9)	36.9 (-31.6)	36.9 (-32.1)	36.9 (-32.2)	36.9 (-32.2)	36.9 (-32.2)	36.9 (-32.2)
	512	40.3 (-34.2)	42.7 (-37.0)	42.9 (-37.9)	42.9 (-38.2)	42.9 (-38.2)	42.9 (-38.2)	42.9 (-38.2)
	1024	42.5 (-37.3)	48.1 (-43.6)	48.9 (-44.1)	48.9 (-44.2)	48.9 (-44.2)	48.9 (-44.2)	48.9 (-44.2)
	2048	43.2 (-39.4)	52.3 (-46.1)	54.6 (-49.0)	54.8 (-50.0)	54.8 (-50.2)	54.8 (-50.3)	54.8 (-50.3)
	4096	43.6 (-40.7)	54.5 (-49.3)	59.9 (-53.9)	60.6 (-55.6)	60.7 (-56.1)	60.7 (-56.3)	60.7 (-56.3)
	8192	43.6 (-41.4)	55.5 (-51.4)	64.0 (-58.0)	66.1 (-61.0)	66.3 (-62.0)	66.3 (-62.3)	66.3 (-62.3)

Tabelle 6: Signal-/Rauschabstand (und größter relativer Fehler) in Dezibel bei Quantisierung des Argumentwertes auf L Werte durch Abschneiden und Quantisierung der Funktionswerte auf k Bit.

3.1.1.2.2 Runden des Phasenwertes

Das Runden des Phasenwertes auf m Bit erfordert eine Addition des $m-1$ -ten Bit an der Stelle m . Tabelle 7 gibt den Signal-/Rauschabstand nach folgender Formel berechnet an:

$$SNR_{\text{Wavetable runden}} = 10 \log \left\{ \frac{\sum_{n=1}^L \left\{ \left(Q_{k-1} \left[\sin \left(\frac{2\pi}{L} (n-0.5) \right) \right] \frac{2\pi}{L} \right)^2 \right\}}{\sum_{n=1}^L \left\{ \int_{x=\frac{2\pi}{L}(n-1)}^{\frac{2\pi}{L}n} \left(Q_{k-1} \left[\sin \left(\frac{2\pi}{L} (n-0.5) \right) \right] - \sin \left(x + \frac{2\pi}{L} (n-1) \right) \right)^2 dx \right\}} \right\}$$

$\frac{SNR_{\text{Wavetable runden}}}{\text{dB}}$		Funktionswertlänge in k Bit						
		8	10	12	14	16	20	24
Speicherfeldgröße L	256	39.9 (-34.0)	42.8 (-37.1)	43.0 (-38.0)	43.0 (-38.2)	43.0 (-38.2)	43.0 (-38.2)	43.0 (-38.2)
	512	42.6 (-37.5)	48.2 (-41.9)	49.0 (-43.6)	49.0 (-44.1)	49.0 (-44.2)	49.0 (-44.2)	49.0 (-44.2)
	1024	43.4 (-39.5)	52.4 (-46.0)	54.8 (-49.0)	55.0 (-50.0)	55.0 (-50.2)	55.0 (-50.3)	55.0 (-50.3)
	2048	43.5 (-40.7)	54.8 (-49.3)	60.3 (-53.9)	61.0 (-55.6)	61.0 (-56.1)	61.0 (-56.3)	61.0 (-56.3)
	4096	43.6 (-41.4)	55.4 (-51.5)	64.4 (-58.0)	66.8 (-61.0)	67.0 (-62.0)	67.0 (-62.3)	67.0 (-62.3)
	8192	43.6 (-41.8)	55.8 (-52.7)	66.8 (-61.2)	72.2 (-65.9)	72.9 (-67.7)	73.0 (-68.3)	72.9 (-68.3)

Tabelle 7: Signal-/Rauschabstand (und größter relativer Fehler) in Dezibel bei Quantisierung des Argumentwertes auf L Werte durch Runden und Quantisierung der Funktionswerte auf k Bit.

Eine Näherung für den optimalen Signal-/Rauschabstand ergibt (siehe [3]):

$$SNR_{\text{Wavetable abschneiden}} = 6(e - 1) \quad [dB]$$

bei einer Funktionswertquantisierung auf $e + 1$ Bit und einer Speicherfeldgröße von $L = 2^e$.

3.1.1.2.3 Interpolationsmethoden

Als Lösungsverfahren zur Interpolation bieten sich zwei Methoden an:

- Interpolationspolynome
- Spline-Interpolation

Interpolationspolynome

Interpolationsmethoden dienen zur Annäherung von Funktionen. Dabei wird eine durch $n + 1$ gegebene Wertepaare $[x_0; y_0], [x_1; y_1], \dots, [x_n; y_n]$ innerhalb eines Intervalles $[a, b]$ definierte Folge mit den Stützwerten y_i an den Stützstellen x_i durch ein Polynom n -ten Grades angenähert, wobei für die Stützstellen $a \leq x_0 < x_1 < x_2 < \dots < x_n \leq b$ gilt.

Für äquidistante Stützstellen, wie dies im Fall der gespeicherten Sinusfolge vorliegt, eignet sich die Interpolationsformel von Gregory-Newton (siehe [4]).

Mit

$$h = x_{k+1} - x_k$$

und dem Bildungsgesetz

$$\Delta^0 y_k := y_k \quad \Delta^n y_k := \Delta^{n-1} y_{k+1} - \Delta^{n-1} y_k \quad , \text{ wobei: } n = 1, 2, \dots, k$$

gilt.

$$I_n(x) = y_0 + \frac{\Delta^1 y_0 (x - x_0)}{1!h} + \frac{\Delta^2 y_0 (x - x_0)(x - x_1)}{2!h^2} + \dots \\ + \frac{\Delta^n y_0 (x - x_0)(x - x_1) \dots (x - x_{n-1})}{n!h^n}$$

Im Fall der Sinusfolgeninterpolation läßt sich das Argument x als Summe eines ganzzahligen Teiles int , welcher der relativen Speicheradresse entspricht, und eines gebrochenen Teiles $frac$ darstellen. Mit $Q_{k-1}[\sin(int)]$ als den auf k Bit quantisierten Funktionswert von int resultieren in Analogie zur Interpolationsformel im Bereich $int \leq x < int + 1$ folgende Ergebnisse:

a.) Für lineare Interpolation ($n = 1$):

$$I_1(x) = I_1(int + frac) \Big|_{int}^{int+1} = Q_{k-1}[\sin(int)] + (Q_{k-1}[\sin(int + 1)] - Q_{k-1}[\sin(int)])frac$$

Der zusätzliche Rechenaufwand beträgt:

- 2 Additionen
- 1 Multiplikation

Die resultierende Signal-/Rauschabstand berechnet sich zu

$$SNR_{\substack{\text{Wavetable} \\ \text{lin.interpol.}}} = 10 \log \left\{ \frac{\sum_{n=1}^L \left\{ \int_{x=n}^{n+1} \left(Q_{k-1} \left[I_1(x) \Big|_n^{n+1} \right] \right)^2 dx \right\}}{\sum_{n=1}^L \left\{ \int_{x=n}^{n+1} \left(Q_{k-1} \left[I_1(x) \Big|_n^{n+1} \right] - \sin \left(\frac{2\pi}{L} (x+n) \right) \right)^2 dx \right\}} \right\}$$

wobei

$$Q_{k-1} \left[I_s(x) \Big|_a^b \right]$$

den nach s -ten Grad zwischen a und b interpolierten und anschließend auf k Bit quantisierten Wert von x wiedergibt. Das Ergebnis enthält Tabelle 8.

$\frac{SNR_{Wavetable}}{lin.interpol.}$ dB		Funktionswertlänge in k Bit						
		8	10	12	14	16	20	24
Speicherfeldgröße L	64	37.5 (-36.1)	49.0 (-46.9)	56.0 (-53.9)	59.8 (-57.4)	60.8 (-58.1)	61.1 (-58.4)	61.1 (-58.4)
	128	38.9 (-36.2)	50.2 (-48.1)	60.7 (-58.9)	68.5 (-66.0)	71.9 (-69.2)	73.1 (-70.4)	73.1 (-70.4)
	256	38.3 (-36.2)	50.6 (-48.5)	61.9 (-60.1)	72.7 (-70.4)	80.5 (-77.6)	84.9 (-82.1)	85.2 (-82.4)
	512	39.3 (-36.4)	50.8 (-48.2)	63.1 (-60.3)	74.4 (-72.4)	85.5 (-82.8)	95.9 (-93.1)	97.2 (-94.4)
	1024	40.7 (-37.4)	50.9 (-48.3)	62.7 (-60.4)	74.7 (-72.4)	86.4 (-84.3)	104.5 (-101.9)	108.9 (-106.2)
	2048	42.0 (-39.5)	51.5 (-48.3)	63.1 (-60.5)	75.1 (-72.5)	87.2 (-84.5)	109.3 (-106.6)	120.0 (-117.3)

Tabelle 8: Signal-/Rauschabstand (und größter relativer Fehler) in Dezibel bei L gegebenen und auf k Bit quantisierten Funktionswerten mit linearer Interpolation und anschließender Quantisierung auf wiederum k Bit.

Eine Näherung für den optimalen Signal-/Rauschabstand bei linearer Interpolation erhält man laut [3] mittels

$$SNR_{\frac{Wavetable}{lin.interpol.}} = 12(e - 1) \quad [dB]$$

bei einer Funktionswertquantisierung auf $e + 1$ Bit und einer Speicherfeldgröße von $L = 2^e$.

b.) Für quadratische Interpolation ($n = 2$):

$$\begin{aligned} I_2(x) &= I_2(int + frac) \Big|_{int}^{int+0.5} = \\ &= Q_{k-1}[\sin(int)] + \\ &+ \left(-\frac{1}{2} Q_{k-1}[\sin(int - 1)] + \frac{1}{2} Q_{k-1}[\sin(int + 1)] \right) frac + \\ &+ \left(\frac{1}{2} Q_{k-1}[\sin(int - 1)] - Q_{k-1}[\sin(int)] + \frac{1}{2} Q_{k-1}[\sin(int + 1)] \right) frac^2 \end{aligned}$$

für $frac < 0.5$

und

$$\begin{aligned}
I_2(x) &= I_2(int + frac) \Big|_{int+0.5}^{int+1} = \\
&= Q_{k-1}[\sin(int)] + \\
&+ \left(-\frac{3}{2} Q_{k-1}[\sin(int)] + 2Q_{k-1}[\sin(int + 1)] - \frac{1}{2} Q_{k-1}[\sin(int + 2)] \right) frac + \\
&+ \left(\frac{1}{2} Q_{k-1}[\sin(int)] - Q_{k-1}[\sin(int + 1)] + \frac{1}{2} Q_{k-1}[\sin(int + 2)] \right) frac^2
\end{aligned}$$

für $frac \geq 0.5$

Der Rechenaufwand beträgt:

- 6 Additionen
- 4 Multiplikationen
- 3 Shift-Operationen
- 1 Bedingungsanweisung

Den Signal-/Rauschabstand nach

$$SNR_{\substack{\text{Wavetable} \\ \text{quadr.interpol.}}} = 10 \log \left\{ \frac{\sum_{n=1}^L \left\{ \int_{x=n}^{n+1} \left(Q_{k-1} \left[I_2(x) \Big|_n^{n+1} \right] \right)^2 dx \right\}}{\sum_{n=1}^L \left\{ \int_{x=n}^{n+1} \left(Q_{k-1} \left[I_2(x) \Big|_n^{n+1} \right] - \sin\left(\frac{2\pi}{L}(x+n)\right) \right)^2 dx \right\}} \right\}$$

berechnet enthält Tabelle 9.

$\frac{SNR_{\substack{\text{Wavetable} \\ \text{quadr.interpol.}}}{dB}}$		Funktionswertlänge in k Bit						
		8	10	12	14	16	20	24
Speicherfeldgröße L	4	16.6 (-15.1)	16.8 (-15.1)	16.8 (-15.1)	16.8 (-15.1)	16.8 (-15.1)	16.8 (-15.1)	16.8 (-15.1)
	8	32.5 (-27.7)	34.2 (-30.7)	34.3 (-31.1)	34.3 (-31.1)	34.3 (-31.1)	34.3 (-31.1)	34.3 (-31.1)
	16	38.3 (-34.6)	48.1 (-42.6)	51.7 (-47.0)	52.3 (-48.4)	52.3 (-48.6)	52.3 (-48.6)	52.3 (-48.6)
	32	38.5 (-35.9)	49.9 (-48.0)	62.8 (-57.4)	69.4 (-63.2)	71.2 (-65.7)	71.3 (-66.5)	71.3 (-66.5)
	64	37.8 (-36.1)	50.9 (-47.9)	61.7 (-59.6)	74.1 (-69.6)	83.0 (-76.4)	85.5 (-80.1)	85.5 (-80.4)
	128	38.9 (-36.4)	50.5 (-48.0)	62.4 (-59.6)	74.1 (-69.8)	82.2 (-76.8)	83.8 (-80.1)	83.8 (-80.4)

Tabelle 9: Signal-/Rauschabstand (und größter relativer Fehler) in Dezibel bei L gegeben und auf k Bit quantisierten Funktionswerten mit quadratischer Interpolation und anschließender Quantisierung auf wiederum k Bit.

c.) Für kubische Interpolation ($n = 3$):

$$\begin{aligned}
 I_3(x) &= I_3(int + frac) \Big|_{int}^{int+1} = \\
 &= Q_{k-1}[\sin(int)] + \\
 &+ \left(-\frac{1}{3} Q_{k-1}[\sin(int-1)] - \frac{1}{2} Q_{k-1}[\sin(int)] + Q_{k-1}[\sin(int+1)] - \frac{1}{6} Q_{k-1}[\sin(int+2)] \right) frac + \\
 &+ \left(\frac{1}{2} Q_{k-1}[\sin(int-1)] - Q_{k-1}[\sin(int)] + \frac{1}{2} Q_{k-1}[\sin(int+1)] \right) frac^2 + \\
 &+ \left(-\frac{1}{6} Q_{k-1}[\sin(int-1)] + \frac{1}{2} Q_{k-1}[\sin(int)] - \frac{1}{2} Q_{k-1}[\sin(int+1)] + \frac{1}{6} Q_{k-1}[\sin(int+2)] \right) frac^3
 \end{aligned}$$

Der Rechenaufwand ist nun:

- 11 Additionen
- 8 Multiplikationen
- 4 Shift-Operationen

Tabelle 10 enthält die nach der Formel

$$SNR_{\substack{\text{Wavetable} \\ \text{kub.interpol.}}} = 10 \log \left\{ \frac{\sum_{n=1}^L \left\{ \int_{x=n}^{n+1} \left(Q_{k-1} \left[I_3(x) \Big|_n^{n+1} \right] \right)^2 dx \right\}}{\sum_{n=1}^L \left\{ \int_{x=n}^{n+1} \left(Q_{k-1} \left[I_3(x) \Big|_n^{n+1} \right] - \sin \left(\frac{2\pi}{L} (x+n) \right) \right)^2 dx \right\}} \right\}$$

berechneten Werte für den Signal-/Rauschabstand.

$\frac{SNR_{\substack{\text{Wavetable} \\ \text{kub.interpol.}}}{\text{dB}}}$		Funktionswertlänge in k Bit						
		8	10	12	14	16	20	24
Speicher- feldgröße	4	20.0 (-18.0)	20.8 (-18.6)	20.9 (-18.7)	20.9 (-18.7)	20.9 (-18.7)	20.9 (-18.7)	20.9 (-18.7)
	8	35.2 (-34.0)	41.7 (-40.0)	43.6 (-41.0)	44.0 (-41.3)	44.2 (-41.4)	44.3 (-41.4)	44.3 (-41.4)
	16	38.6 (-36.2)	49.6 (-48.4)	58.4 (-57.3)	65.4 (-62.6)	67.2 (-64.6)	68.0 (-65.1)	68.0 (-65.2)
	32	38.6 (-36.2)	49.9 (-48.4)	63.4 (-60.7)	73.4 (-69.6)	83.5 (-76.4)	89.6 (-80.2)	90.4 (-80.3)
L	64	37.9 (-36.3)	50.9 (-48.1)	61.7 (-60.2)	74.4 (-69.6)	86.0 (-76.4)	97.8 (-80.1)	98.3 (-80.4)
	128	39.0 (-36.4)	50.5 (-48.2)	62.5 (-60.4)	74.7 (-72.3)	87.1 (-77.7)	100.8 (-80.1)	101.4 (-80.4)

Tabelle 10: Signal-/Rauschabstand (und größter relativer Fehler) in Dezibel bei gegebenen L und auf k Bit quantisierten Funktionswerten mit kubischer Interpolation und anschließender Quantisierung auf wiederum k Bit.

Spline-Interpolation

Da bei der stückweisen Interpolation mittels Polynomen mindestens in der ersten Ableitung Unstetigkeiten an den Stützstellen auftreten, verwendet man vorzugsweise die sogenannte Spline-Interpolation, bei der zusätzlich die Ableitungen des Interpolationspolynoms gleichgesetzt werden. Dadurch wird die Gesamtkrümmung der interpolierenden Kurve minimal.

Es seien wiederum $n+1$ Stützstellen im Intervall $[a, b]$ gegeben. Die gesuchte Funktion $\hat{S}_k(x)$ wird interpolierende Spline-Funktion vom Grad k genannt. Diese erhält man aus der Menge der Spline-Funktionen $S_k(x)$ über K_n , wobei K_n ein System der Stützstellen $a = x_0 < x_1 < x_2 < \dots < x_n = b$ repräsentiert. Für die Menge der Spline-Funktionen über K_n gilt:

- $S_k(x) \in C^{k-1}([a, b])$
- $S_k(x)$ ist für $x \in [x_{j-1}, x_j]$ mit $j = 1, 2, \dots, n$ ein Polynom höchstens vom Grade k

Die interpolierende Spline-Funktion $\hat{S}_k(x)$ erhält man aus $S_k(K_n)$, wenn $S_k(x_j) = y_j = f(x_j)$ für $j = 0, 1, \dots, n$ gilt. In den meisten Anwendungen genügt es, $k = 3$ zu wählen und die sogenannte kubische Spline-Interpolation zu verwenden ([4]):

$$\begin{aligned} h_j &= x_j - x_{j-1} \\ \hat{S}_3(x) &= s(x) \\ s_j^{(i)} &= s^{(i)}(x_j) \quad \text{für } i = 0, 1, 2 \text{ und } j = 0, 1, \dots, n \end{aligned}$$

Im speziellen ergibt sich für die periodische kubische Spline-Interpolation einer Sinusfunktion mit den an L äquidistanten Stützstellen auftretenden Stützwerten

$$y_n = Q_{k-1}[\sin(\frac{2\pi}{L}(n+0.5))] \text{ für } n = 0, 1, \dots, L-1$$

folgendes Gleichungssystem:

$$s_1'' \frac{4\pi}{3L} + s_2'' \frac{\pi}{3L} + s_L'' \frac{\pi}{3L} = \frac{L}{2\pi} (y_2 - 2y_1 + y_0)$$

$$s_{n-1}'' \frac{\pi}{3L} + s_n'' \frac{4\pi}{3L} + s_{n+1}'' \frac{\pi}{3L} = \frac{L}{2\pi} (y_{n+1} - 2y_n + y_{n-1}) \text{ für } n = 1, 2, \dots, L-1$$

$$s_1'' \frac{\pi}{3L} + s_{L-1}'' \frac{\pi}{3L} + s_L'' \frac{4\pi}{3L} = \frac{L}{2\pi} (y_1 - 2y_n + y_{n-1})$$

Die Koeffizienten s_n'' für $n = 0, 1, \dots, L$ ergeben in

$$s(x) = s_{n-1}'' \frac{L}{12\pi} (1 - \text{frac})^3 + s_n'' \frac{L}{12\pi} \text{frac}^3 + \left(\frac{y_n L}{2\pi} - \frac{2\pi s_n''}{6L} \right) \text{frac} - \left(\frac{y_{n-1} L}{2\pi} - \frac{\pi s_{n-1}''}{3L} \right) (1 - \text{frac})$$

eingesetzt die gesuchte interpolierende Spline-Funktion. Der dadurch erreichbare Signal-/Rauschabstand wurde mit nachfolgender Formel berechnet und ist in Tabelle 11 zusammengefaßt.

$$SNR_{\substack{\text{Wavetable} \\ \text{Spline-Interp.}}} = 10 \log \left\{ \frac{\sum_{n=1}^L \left\{ \int_{x=n}^{n+1} \left(\mathcal{Q}_{k-1} \left[s(x) \Big|_n^{n+1} \right] \right)^2 dx \right\}}{\sum_{n=1}^L \left\{ \int_{x=n}^{n+1} \left(\mathcal{Q}_{k-1} \left[s(x) \Big|_n^{n+1} \right] - \sin \left(\frac{2\pi}{L} (x+n) \right) \right)^2 dx \right\}} \right\}$$

$SNR_{\substack{\text{Wavetable} \\ \text{Spline-Interp.}}}$		Funktionswertlänge in k Bit						
		8	10	12	14	16	20	24
dB								
Speicherfeldgröße	2	20.3 (-9.6)	20.3 (-9.6)	20.4 (-9.6)	20.4 (-9.6)	20.4 (-9.6)	20.4 (-9.6)	20.4 (-9.6)
	4	32.0 (-26.4)	33.6 (-28.4)	33.7 (-28.8)	33.7 (-28.8)	33.7 (-28.9)	33.7 (-28.9)	33.7 (-28.9)
	8	38.5 (-35.6)	50.4 (-46.6)	58.4 (-54.3)	60.8 (-57.3)	61.6 (-58.5)	61.9 (-58.9)	61.9 (-58.9)
	16	38.8 (-36.1)	50.4 (-48.3)	61.2 (-60.5)	74.6 (-71.6)	81.5 (-79.6)	87.1 (-83.3)	87.5 (-83.8)
	32	38.6 (-36.0)	49.9 (-48.4)	63.5 (-61.0)	74.1 (-72.2)	86.9 (-84.1)	106.1 (-103.5)	111.6 (-107.7)
	64	37.8 (-36.1)	50.9 (-47.8)	61.7 (-60.0)	74.4 (-72.3)	86.6 (84.4)	110.5 (-107.6)	129.9 (-126.2)
	128	38.6 (-36.3)	50.5 (-48.2)	62.5 (-60.1)	74.7 (-72.1)	87.4 (-83.9)	111.1 (-108.7)	134.5 (-132.5)

Tabelle 11: Signal-/Rauschabstand (und größter relativer Fehler) in Dezibel bei gegebenem L und auf k Bit quantisierten Funktionswerten mit kubischer Spline-Interpolation und anschließender Quantisierung auf wiederum k Bit.

Der Rechenaufwand und Speicherplatzbedarf beträgt:

- 4 Additionen
- 8 Multiplikationen
- $3L$ Speicherplätze, da die Koeffizienten nach dem Lösen des Gleichungssystems gespeichert werden

3.1.1.2.4 Berücksichtigung der Viertelwellensymmetrie

Wie bereits im Abschnitt 3.1.1.1.2 erwähnt, läßt sich durch Berücksichtigung der Viertelwellensymmetrie der Sinusfunktion bei geeigneter Argumentzuweisung der Speicherplatzbedarf auf ein Viertel verringern. Zusätzlich verringert sich die Funktionswertauflösung um ein Bit, da in diesem Fall kein Vorzeichenbit notwendig ist.

3.1.1.2.5 Frequenz- und Phasenmodulation

Um Aliasing zu vermeiden, muß der Inkrementwert I kleiner als die halbe Speicherfeldgröße L sein. Wird der Inkrementwert inklusive des Vorzeichenbits mit l Bit aufgelöst, so lassen sich nur 2^l Inkrementwerte darstellen, wodurch auch die Frequenzauflösung f_Δ auf den Wert

$$f_\Delta = \frac{f_s}{2^l} \quad [\text{Hz}]$$

festgelegt ist. Bei dynamischer Ansteuerung muß das Inkrement wie folgt berechnet werden.

$$I = f_1[n] \frac{L}{f_s} + \Psi[n] \frac{L}{2\pi}$$

Man benötigt also zusätzlich:

- 2 Multiplikationen
- 1 Addition

Eine Methode, die Frequenzauflösung zu verbessern, erhält man durch eine Vergrößerung des Phasenregisters auf p Bit und eine variable Handhabung des Inkrementregisters. Letzteres wird mittels Shift-Arithmetik über das Phasenregister verschoben, wodurch die relative Frequenzauflösung konstant bleibt. Bleibt die Frequenz innerhalb einer Oktave, so muß nur das Inkrementregister zum Phasenregister addiert werden. Bei einem Oktavwechsel wird das Inkrementregister um ein Bit verschoben. Bild 23 illustriert diesen Algorithmus.

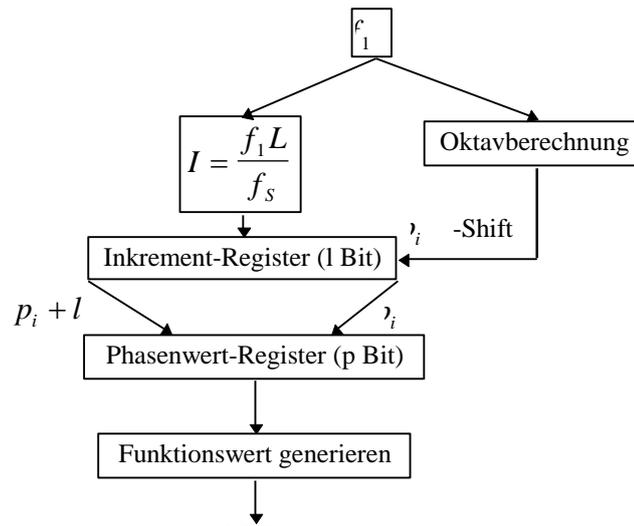


Bild 23: Algorithmus zur Wavetable-Lookup-Methode mit konstanter relativer Frequenzauflösung.

3.1.1.3 Rekursive Algorithmen

Eine weitere Möglichkeit, Sinusfolgen zu erzeugen, besteht durch die Anwendung von rekursiven Algorithmen. Dabei werden die Augenblickswerte der Folge über zeitlich vorausgegangene Werte und Zustandsvariablen des Systems berechnet. Der Vorteil liegt im geringen Speicherplatzbedarf und Rechenaufwand. Zusätzlich läßt sich dieses Verfahren für die Hardware einfach implementieren. Nachteilig bei diesen Verfahren sind jedoch die im Abschnitt 3.0.2.2 erwähnten Probleme durch die endliche Rechengenauigkeit. Um bei solchen Systemen Stabilität zu gewährleisten, ist eine hohe Auflösung der Koeffizienten erforderlich. Weiters ist eine dynamische Ansteuerung von Frequenz und Phase problematisch, da eine Änderung eines Parameters auch die Systemeigenschaften verändern kann, wodurch weitere Zustandsvariablen beeinflusst werden. Im folgenden sollen vier Verfahren ([1],[2]) von rekursiven Algorithmen vorgestellt und bezüglich ihres Rechenaufwandes überprüft werden.

3.1.1.3.1 Gekoppelte Form

Algorithmus

Die gekoppelte Form kann über das Gleichungssystem

$$\begin{aligned} x[n] &= \cos(\Theta)x[n-1] + \sin(\Theta)y[n-1] \\ y[n] &= -\sin(\Theta)x[n-1] + \cos(\Theta)y[n-1] \end{aligned}$$

bzw. in Matrizendarstellung

$$\begin{bmatrix} x[n] \\ y[n] \end{bmatrix} = \begin{bmatrix} \cos(\Theta) & \sin(\Theta) \\ -\sin(\Theta) & \cos(\Theta) \end{bmatrix} \begin{bmatrix} x[n-1] \\ y[n-1] \end{bmatrix} = \begin{bmatrix} \cos(\Theta) & \sin(\Theta) \\ -\sin(\Theta) & \cos(\Theta) \end{bmatrix}^n \begin{bmatrix} x[0] \\ y[0] \end{bmatrix} = \begin{bmatrix} \cos[n\Theta] & \sin[n\Theta] \\ -\sin[n\Theta] & \cos[n\Theta] \end{bmatrix} \begin{bmatrix} x[0] \\ y[0] \end{bmatrix}$$

beschrieben werden.

Mit dem Anfangsvektor

$$\begin{bmatrix} x[0] \\ y[0] \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

ergibt sich für $x[n]$ eine cosinusförmige und für $y[n]$ eine sinusförmige Folge mit der Frequenz Θ . Um Stabilität zu gewährleisten, ist es notwendig, daß der Wert der Determinante konstant 1 ist, was durch die endliche Darstellungsgenauigkeit der Koeffizienten nicht immer gewährleistet ist und in weiterer Folge zur oben erwähnten Instabilität führen kann.

Der Rechenaufwand und Speicherplatzbedarf beträgt:

- 4 Multiplikationen
- 2 Additionen
- 2 Speicherplätze

Frequenzmodulation

Frequenzmodulation ist durch direkte Manipulation von $\cos(\Theta)$ und $\sin(\Theta)$ möglich, jedoch steigt der Rechenaufwand nun um:

- 1 cos-Berechnung
- 1 sin-Berechnung

3.1.1.3.2 Modifizierte gekoppelte Form

Algorithmus

Die modifizierte gekoppelte Form ist eine Erweiterung der gekoppelten Form. Durch Verwendung eines anderen Koeffizientensatzes läßt sich dadurch ein stabiles Verhalten gewährleisten. Das Gleichungssystem lautet

$$\begin{aligned} x[n+1] &= x[n] - \varepsilon y[n] \\ y[n+1] &= \varepsilon x[n+1] + y[n] \end{aligned}$$

und die Matrizendarstellung

$$\begin{bmatrix} x[n+1] \\ y[n+1] \end{bmatrix} = \begin{bmatrix} 1 & -\varepsilon \\ \varepsilon & 1 - \varepsilon^2 \end{bmatrix}^n \begin{bmatrix} x[0] \\ y[0] \end{bmatrix}$$

Man erkennt, daß unabhängig von der Darstellungsgenauigkeit von ε die Determinante in jedem Fall 1 ist. Der Zusammenhang zwischen der Frequenz und dem Koeffizienten ε ist wie folgt:

$$\sin\left(\frac{\Theta}{2}\right) = \frac{\varepsilon}{2} \quad \text{und} \quad \cos\left(\frac{\Theta}{2}\right) = \sqrt{1 - \varepsilon^2 / 4}$$

Eine Aussteuerung von $-2 < \varepsilon < 2$ korrespondiert mit der Frequenz $-\pi < \Theta < \pi$.

Der Rechenaufwand und Speicherplatzbedarf reduziert sich hier auf:

- 2 Multiplikationen
- 2 Additionen
- 2 Speicherplätze

Frequenzmodulation

Um Phasentreue zu gewährleisten, ist es notwendig, den frequenzabhängigen Term $\cos(n\Theta - \Psi)$ in den neuen Term $\cos(m\Theta[n+1] - \Psi[n+1])$ überzuführen. Mittels der trigonometrischen Beziehungen läßt sich auch folgende Form anschreiben (siehe [1]):

$$\cos(n\Theta - \Psi) = \cos(n\Theta) \sin\left(\frac{\Theta[n+1]}{2}\right) + \sin(n\Theta) \cos\left(\frac{\Theta[n+1]}{2}\right)$$

Um jedoch den Term $\sin(n\Theta)$ zu erhalten, ist es notwendig, einen zweiten Generator in modifizierter gekoppelter Form zu betreiben. Von den resultierenden 4 Gleichungen müssen jedoch nur 2 berechnet werden, da sich die x -Werte lediglich aus vorhergehenden x -Werten ergeben. Die Gleichungen lauten:

$$\begin{aligned} x_1[n+1] &= x_1[n] - \varepsilon[n+1] \left(\sin\left(\frac{\Theta[n+1]}{2}\right) x_1[n] + \cos\left(\frac{\Theta[n+1]}{2}\right) x_2[n] \right) \\ x_2[n+1] &= x_2[n] - \varepsilon[n+1] \left(\sin\left(\frac{\Theta[n+1]}{2}\right) x_2[n] - \cos\left(\frac{\Theta[n+1]}{2}\right) x_1[n] \right) \end{aligned}$$

Der Rechenaufwand erhöht sich nun auf:

- 6 Multiplikationen
- 4 Additionen
- 1 Berechnung von $\sin\left(\frac{\Theta[n+1]}{2}\right)$
- 1 Berechnung von $\cos\left(\frac{\Theta[n+1]}{2}\right)$

3.1.1.3.3 Digitaler Resonator

Algorithmus

Einen einfachen digitalen Resonator erhält man aus einem zweipoligen rekursiven Filter, dessen Dämpfung auf 0 gesetzt wird.

$$y[n] = 2R \cos(\Theta)y[n-1] - R^2 y[n-2] + x[n] \quad (3)$$

Durch $R = 1$ werden die Pole auf den Einheitskreis geschoben, woraus der Fall der ungedämpften Schwingung resultiert. Das Filter benötigt als Eingangssignal einen Impuls, welcher mit dem Faktor $\sin(\Theta)$ skaliert wird, um die Ausgangsfolge auf den Bereich ± 1 einzuschränken. Gleichung (3) liefert bereits eine Sinusschwingung. Durch eine Erweiterung auf

$$y[n] = 2R \cos(\Theta)y[n-1] - R^2 y[n-2] + x[n] - R \cos(\Theta)x[n-1]$$

ergibt sich eine Cosinusfolge. Der Rechenaufwand beträgt:

- 1 Multiplikation
- 1 Addition

Der Nachteil bei der Verwendung eines digitalen Resonators besteht im geringen Signal-/Rauschabstand besonders bei tiefen Frequenzen (< 10 Dezibel bei Verwendung von 14 Bits für den gebrochenen Teil der Festpunktzahl [1]).

Frequenzmodulation

Wie bei der modifizierten gekoppelten Form ist auch hier eine Überführung des Terms $\cos((n-1)\Theta)$ in $\cos((m-1)\Theta[n+1])$ notwendig (siehe oben).

3.1.1.3.4 Digitaler Oszillator mittels Gitterstruktur

Ein weiterer Sinusoszillator kann durch Physical Modelling gewonnen werden (siehe [2]). Dabei werden zwei akustische Röhren miteinander verbunden, welche nach außen keine Übertragung ermöglichen. Die Energie innerhalb des Systems bleibt konstant. Bild 24 zeigt ein Modell dieses Systems und den resultierenden Signalfußgraphen.

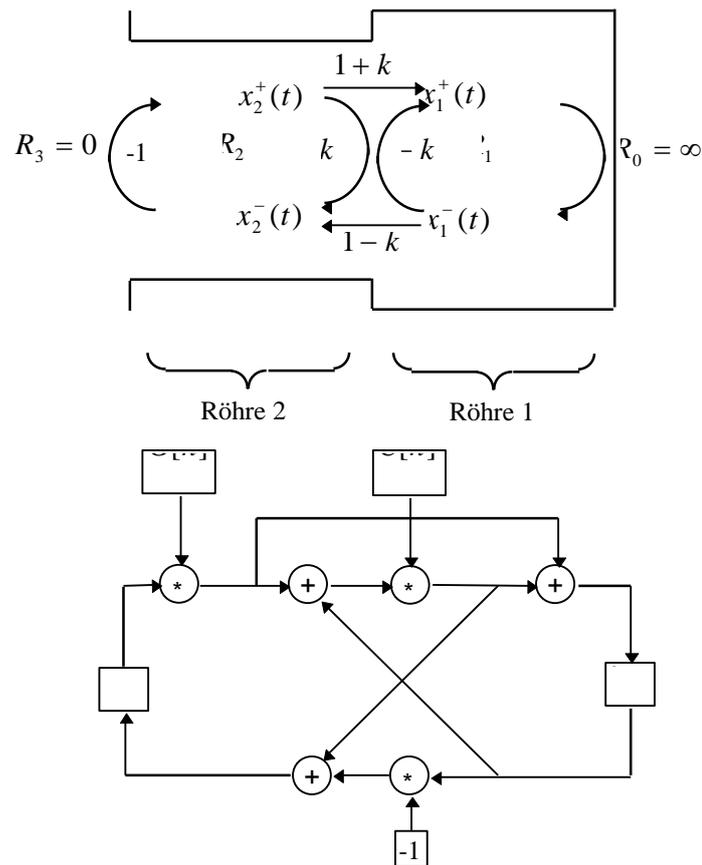


Bild 24: Modell einer akustischen Doppelröhre und Signalflußgraph einer normierten Gitterstruktur zur Erzeugung von Sinusfolgen.

Die Koeffizienten lauten:

$$G[n] = \frac{r[n]g[n]}{g[n-1]} \quad C[n] = \cos(\Theta)$$

wobei:

$$r[n] = e^{-\frac{T}{\tau_n}} \quad \text{und} \quad g[n] = \sqrt{\frac{1-C[n]}{1+C[n]}}$$

Wenn die Frequenz konstant bleibt, wird $G[n] = 1$ und der Rechenaufwand beträgt:

- 1 Multiplikation
- 3 Additionen
- 3 Speicherplätze

Desweiteren ist auch eine dynamische Frequenzsteuerung möglich. Da eine Frequenzänderung auch eine Amplitudenänderung zur Folge hat, muß diese mittels $r[n]$ in $G[n]$ kompensiert werden.

3.1.2 Impuls-, Rechteck-, Dreieck- und Sägezahngeneratoren

3.1.2.0 Allgemeines

Im folgenden Kapitel sollen die von Analogsynthesizern vorwiegend angebotenen Kurvenformen in Hinblick auf eine digitale Realisierung untersucht werden. Dazu zählen Impuls-, Rechteck-, Dreieck- und Sägezahnfunktionen.

Die Impulsfunktion setzt sich aus unendlich vielen Harmonischen gleicher Amplitude zusammen. Funktionen mit un stetigen Stellen (Rechteck und Sägezahn) zeichnen sich durch ein Spektrum aus, welches pro Oktave um nicht mehr als 6 Dezibel abnimmt. Das Spektrum von Funktionen mit stetigen Verlauf aber nicht differenzierbaren Stellen (Dreieck) fällt mit 12 Dezibel pro Oktave ab.

Theoretisch haben diese Funktionen eine unendliche Bandbreite, wodurch das Problem des Aliasing in den folgenden Betrachtungen in den Vordergrund tritt. Die Algorithmen werden wiederum auf ihren Rechenaufwand untersucht. Zusätzlich wird eine Ansteuerung des Tastverhältnisses betrachtet.

3.1.2.1 Additive Synthese im Zeitbereich

3.1.2.1.1 Additive Synthese mittels Sinusgeneratoren

Die einfachste Form der Erzeugung von tiefpaßgefilterten Impulssignalen ist die Realisierung mittels additiver Synthese. Dabei werden die einzelnen harmonischen Schwingungen des Signals mittels Sinusgeneratoren, wie sie im Kapitel 3.1 besprochen wurden, generiert und deren Signale addiert.

Zur Erzeugung einer Schwingung mit der Grundfrequenz f_1 werden

$$N = \text{int}\left(\frac{f_s}{2f_1}\right)$$

Sinusgeneratoren benötigt.

Die Berechnung des Ausgangssignals erfolgt allgemein über:

$$y[n] = \sum_{k=1}^N a_k[n] \cos(\Phi_k[n])$$

wobei:

$$\Phi_k[n] = \Phi_k[n-1] + \Theta_k[n] + \psi_k[n] - \psi_k[n-1]$$

Um eine aliasingfreie tiefpaßgefilterte Impulsfolge mit N Harmonischen zu generieren, ergibt sich mit

$$y[n] = \frac{1}{N} \sum_{k=0}^N \cos(k(\Phi[n-1] + \Theta[n]))$$

folgender Rechenaufwand:

- N trigonometrische Berechnungen (siehe Kapitel 3.1.1)
- N Multiplikationen
- $2N - 1$ Additionen

Aufgrund des hohen Rechenaufwandes ist diese Methode für niedrige Grundfrequenzen ungeeignet.

3.1.2.1.2 Wavetable-Synthese

Wie im Kapitel 3.1.1.2 besprochen, lassen sich Sinusfolgen aus einer Wavetable synthetisieren. Dieselben Überlegungen gelten auch für beliebige Wellenformen, welche sich aus einer Superposition einzelner Sinusfolgen ergeben. Die resultierenden Kurvenformen lassen sich somit direkt in die Wavetable schreiben. Dabei muß jedoch für das Phaseninkrement folgende Bedingung erfüllt sein, um Aliasing zu vermeiden:

$$I \leq \frac{L f_1}{f_s n_h}$$

wobei:

f_1 ... Grundfrequenz

n_h ... Nummer der höchstfrequenten Harmonischen ($f_{\max} = n_h f_1$)

L ... Größe der Wavetable

Daher ist dieses Verfahren in Hinblick auf Frequenzmodulation nur für kleine Frequenzänderungen geeignet, da ansonsten die Wavetable dynamisch berechnet werden müßte bzw. mehrere Wavetables notwendig wären.

3.1.2.1.3 Periodische Impulsfunktion

Allgemeines

Eine ideale periodische Impulsfunktion läßt sich im Zeit- und Frequenzbereich wie folgt anschreiben:

$$\sum_{n=-\infty}^{\infty} \delta(t - nT_1) \Leftrightarrow \frac{1}{T} \sum_{n=-\infty}^{\infty} \delta(\Omega - n\Omega_1) \quad \text{mit } \Omega_1 = \frac{2\pi}{T_1}$$

Die Multiplikation dieser Folge mit der Abtastfunktion liefert nur Werte ungleich null, wenn die Impulse der abzutastenden Funktion genau auf Werten der Abtastfolge liegen. Durch die theoretisch unendlich schmalen Impulse sind diese Abtastwerte nur zufällig verteilt und liefern keine zufriedenstellende Repräsentation des Nutzsignals.

Zeitlich gerundete periodische Impulsfunktion

Diese Problematik lässt sich wie folgt lösen (siehe [6]): Durch eine Verbreiterung der Impulse der abzutastenden Funktion auf

$$p(t) = \begin{cases} 1 & |t| \leq T_s / 2 \\ 0 & |t| > T_s / 2 \end{cases} \Leftrightarrow P(\Omega) = T_s \frac{\sin(\pi f T_s)}{\pi f T_s} = T_s \operatorname{sinc}(f T_s)$$

erhält man anstatt der periodischen Impulsfunktion nun:

$$x(t) = \sum_{n=-\infty}^{\infty} p(t - nT_1) \Leftrightarrow X(\Omega) = P(\Omega) \sum_{n=-\infty}^{\infty} \delta(\Omega - n\Omega_1)$$

Die Impulsbreite wird also auf die Länge der Abtastperiode vergrößert. Durch Abtasten dieser Funktion werden die Impulse des Nutzsignals zeitlich auf den nächstgelegenen Abtastwert gerundet. Man erhält schließlich:

$$x(kT_s) = \sum_{n=-\infty}^{\infty} p(kT_s - nT_1) \Leftrightarrow \sum_{k=-\infty}^{\infty} X(\Omega + k\Omega_s) = \sum_{k=-\infty}^{\infty} P(\Omega + k\Omega_s) \sum_{n=-\infty}^{\infty} \delta(\Omega + k\Omega_s - n\Omega_1)$$

Die k-Terme verursachen starke Aliasing-Frequenzen, wie aus Bild 25 entnommen werden kann. Lediglich für tiefe Nutzsignalfrequenzen ergeben sich annehmbare Ergebnisse (siehe [6]).

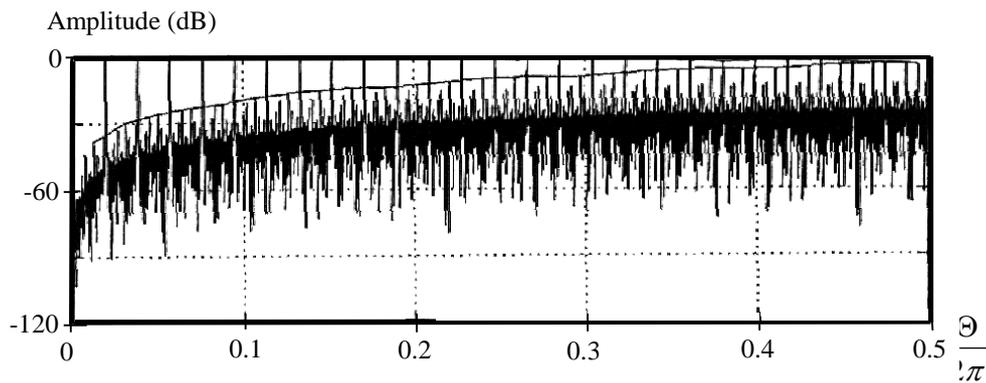


Bild 25: Spektrum einer Impulsfunktion, deren Impulse bei der Abtastung auf den nächstgelegenen Abtastwert gerundet wurden. Die durchgezogene Linie zeigt die Einhüllende. Bild nach [6].

Linear interpolierte periodische Impulsfunktion

Bessere Ergebnisse liefert eine lineare Interpolation der Funktionswertimpulse auf die beiden benachbarten Impulse der Abtastfunktion. Die Impulse der Abtastfunktion werden entsprechend ihrer Nähe zum Impuls des Nutzsignales gewichtet, wobei die Summe beider Impulse 1 ergibt. Bild 26 zeigt das Spektrum einer nach dieser Methode generierten Impulsfunktion.

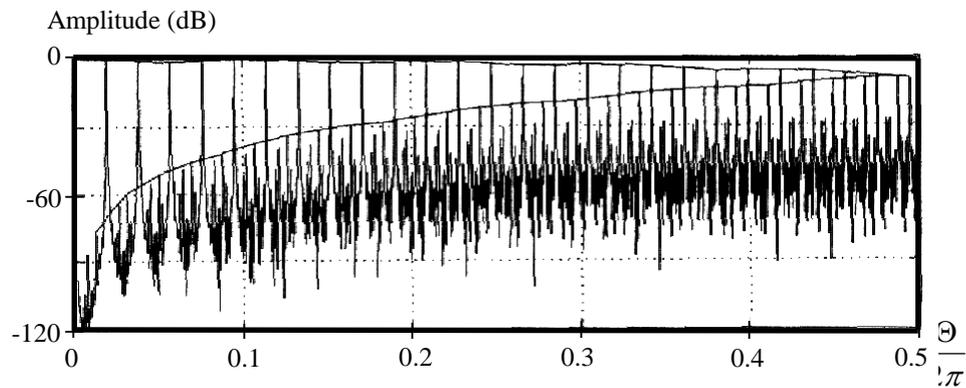


Bild 26: Spektrum einer Impulsfunktion mit linear interpolierter Abtastung. Die durchgezogene Linie zeigt die Einhüllende. Bild nach [6].

3.1.2.1.4 Tiefpaßgefilterte periodische Impulsfunktion

Allgemeines

In diesem Abschnitt wird die Erzeugung einer tiefpaßgefilterten periodischen (also aliasingfreien) Impulsfunktion betrachtet, deren Spektrum auf N Harmonische begrenzt ist.

$$X(\Omega) = \frac{1}{N} \sum_{n=-N}^N \delta(\Omega - n\Omega_1)$$

Dies repräsentiert eine mit der Grenzfrequenz $f_{\max} = Nf_1$ ideal tiefpaßgefilterte Impulsfunktion.

Aus dieser läßt sich schließlich durch einfache mathematische Beziehungen die Rechteckfunktion (durch Integration) und die Dreieck- bzw. Sägezahnfunktionen (durch zweifache Integration) herleiten. Durch eine Phasenverschiebung der Impulse läßt sich auch die Steuerung des Tastverhältnisses auf einfachem Wege realisieren. Bild 27 illustriert das Prinzip.

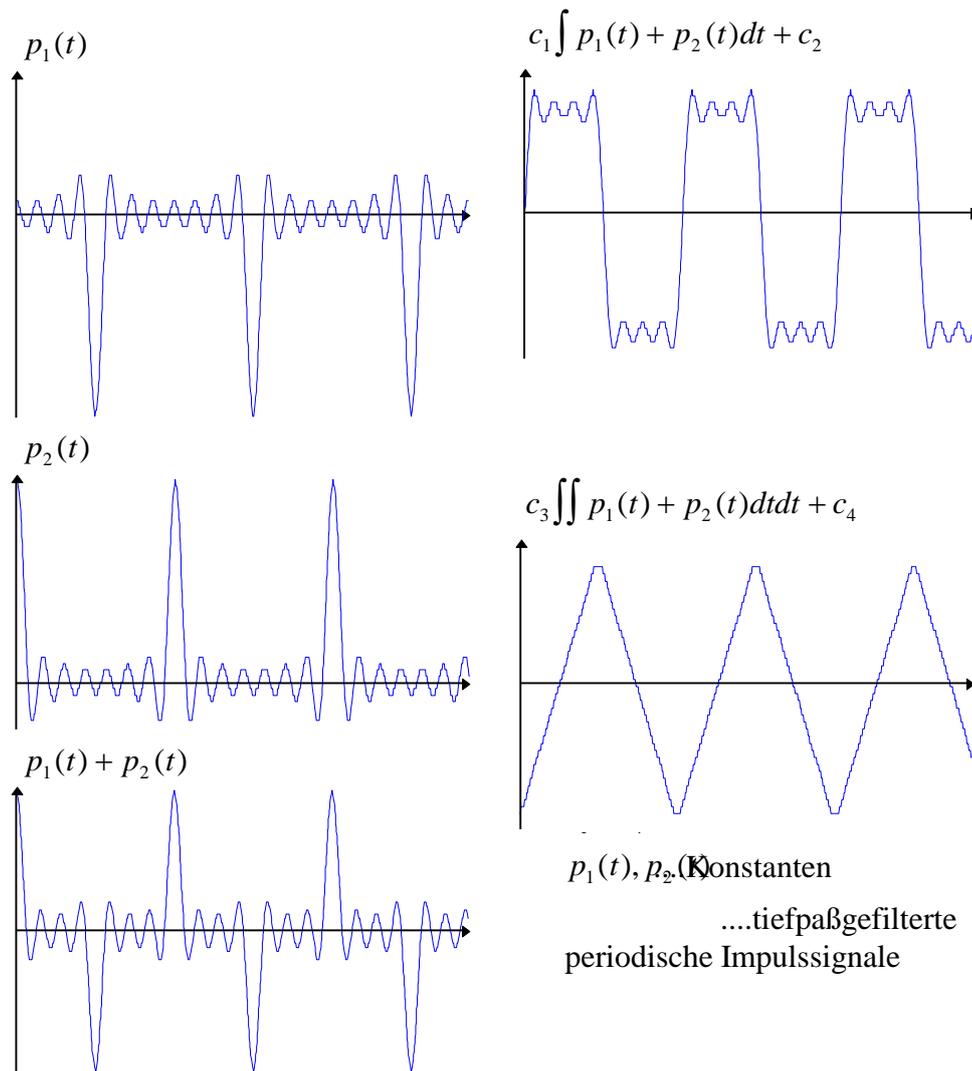


Bild 27: Herleitung einer aliasingfreien Rechteck- und Dreieckfunktion aus zwei tiefpaßgefilterten periodischen Impulsfunktionen.

Diskrete Summenformel

Moorer [7] verwendet zur Berechnung von Summen von Sinusschwingungen in geschlossener Form die diskrete Summenformel. Basierend auf der geometrischen Reihe

$$\sum_{k=0}^{N-1} z^k = \frac{1 - z^N}{1 - z}$$

ergibt sich mit der Identität $2 \cos(x) = e^{jx} + e^{-jx}$:

$$\begin{aligned} \frac{1}{N} \sum_{k=0}^{N-1} a^k \cos(k\Theta n + \varphi) &= \\ &= \frac{1}{N} \frac{\cos(\varphi) - a \cos(\Theta n + \varphi) - a^N \cos(N\Theta n + \varphi) + a^{N+1} \cos((N-1)\Theta n + \varphi)}{1 - 2a \cos(\Theta n) + a^2} \end{aligned}$$

und mittels $2j \sin(x) = e^{jx} - e^{-jx}$:

$$\begin{aligned} \frac{1}{N} \sum_{k=0}^{N-1} a^k \sin(k\Theta n + \varphi) &= \\ &= \frac{1}{N} \frac{\sin(\varphi) - a \sin(\Theta n + \varphi) - a^N \sin(N\Theta n + \varphi) + a^{N+1} \sin((N-1)\Theta n + \varphi)}{1 - 2a \cos(\Theta n) + a^2} \end{aligned}$$

Hieraus lassen sich mittels trigonometrischen Beziehungen weitere vereinfachte Formeln in geschlossener Form zur Erzeugung von tiefpaßgefilterten Impulsfunktionen herleiten. Im folgenden werden die entsprechenden Realisierungen und deren notwendiger Rechenaufwand angegeben.

a.) tiefpaßgefilterte Impulsfolge mit Gleichanteil:

$$\frac{1}{N} \sum_{k=0}^{N-1} \cos(k\Theta n) = \frac{1}{2N} \frac{1 + \cos((N-1)\Theta n) - \cos(N\Theta n) - \cos(\Theta n)}{1 - \cos(\Theta n)}$$

Rechenaufwand:

- 3 trigonometrische Berechnungen
- 4 Multiplikationen
- 5 Additionen
- 1 Division
- 1 Shift-Operation

b.) tiefpaßgefilterte Impulsfolge ohne Gleichanteil ([8]):

$$\frac{1}{N} \sum_{k=1}^N \cos(k\Theta n) = \frac{1}{2N} \left(\frac{\sin((2N+1)\Theta n / 2)}{\sin(\Theta n / 2)} - 1 \right)$$

Um eine Division durch Null zu verhindern, muß der Divisor vor jeder Berechnung untersucht werden. Meist genügt es, das Ergebnis für kleine Werte des Divisors gleich 1 zu setzen. Eine genauere Berücksichtigung kann über folgende Näherungsformel erfolgen ([9]):

$$\frac{\sin(kx)}{\sin(x)} \approx \frac{k \cos(kx)}{\cos(x)} \quad \text{für } x \ll 1$$

Rechenaufwand:

- 2 trigonometrische Berechnungen
- 3 Multiplikationen
- 2 Additionen
- 1 Division
- 2 Shift-Operationen

c.) tiefpaßgefilterte bipolare Impulsfolge mit einem Tastverhältnis von 50 Prozent:

$$\frac{1}{N} \left(\sum_{k=0}^{N-1} \cos(k\Theta n) - \sum_{k=0}^{N-1} (-1)^k \cos(k\Theta n) \right) = \frac{\cos((N-2)\Theta n) - \cos(N\Theta n)}{N(1 - \cos(2\Theta n))}$$

N ...ungerade

Mit $a = -1$ erhält man eine Phasenverschiebung um 180 Grad für jede ungeradzahli-
ge Harmonische, wodurch ein um die halbe Periodendauer zeitversetzter Impulszug rea-
lisiert wird. Dieser führt durch Subtraktion von ersterem zu einer bipolaren Impulsfolge
mit einem Tastverhältnis von 50 Prozent. Für ein geradzahliges N erhält man wieder-
um eine unipolare Impulsfolge mit der doppelten Frequenz.

Rechenaufwand:

- 3 trigonometrische Berechnungen
- 4 Multiplikationen
- 3 Additionen
- 1 Division
- 1 Shift-Operation

d.) tiefpaßgefilterte bipolare Impulsfolge mit beliebigem Tastverhältnis:

$$\frac{1}{N} \sum_{k=0}^{N-1} e^{jk\alpha} \sin(k\Theta n) = 2 \cdot \operatorname{Im} \left\{ \frac{e^{jN\alpha} \sin((N-1)\Theta n) - \sin(\Theta n) - e^{j(N-1)\alpha} \sin(N\Theta n)}{N(\cos(\Theta n) + \cos(\alpha))} \right\}$$

Mittels komplexer Arithmetik läßt sich über den Wert a auch das Tastverhältnis der
bipolaren Impulsfunktion beeinflussen. Der Zusammenhang zwischen dem Tastverhält-
nis t_v in Prozent und α gibt folgende Beziehung wieder:

$$\alpha = -\frac{\pi}{100} t_v$$

Rechenaufwand:

- 5 trigonometrische Berechnungen
- 8 Multiplikationen (davon 2 komplex)
- 4 Additionen (davon 1 komplex)
- 1 Division
- 1 Shift-Operation

Zum Vergleich benötigt die Realisierung einer bipolaren Impulsfolge mit Tastverhältnissteuerung die Berechnung von zwei unipolaren Impulsfolgen, welche um n_{t_v} Abtastwerte zeitverschoben sind und voneinander subtrahiert werden, folgenden Rechenaufwand:

$$\begin{aligned} \frac{1}{N} \sum_{k=1}^N \cos(k\Theta n) - \frac{1}{N} \sum_{k=1}^N \cos(k\Theta(n - n_{t_v})) &= \\ &= \frac{1}{2N} \left(\frac{\sin((2N+1)\Theta n / 2)}{\sin(\Theta n / 2)} - \frac{\sin((2N+1)\Theta(n - n_{t_v}) / 2)}{\sin(\Theta(n - n_{t_v}) / 2)} \right) \end{aligned}$$

- 4 trigonometrische Berechnungen
- 5 Multiplikationen
- 3 Additionen
- 2 Divisionen
- 4 Shift-Operationen

Gefensterte Sinc-Funktion

Neben den geschlossenen Formen lassen sich tiefpaßgefilterte Impulsfolgen auch mittels Interpolationstechniken generieren ([6],[11]). Die Grundlage hierzu bildet der FIR-Filterentwurf nach der Fenstermethode.

Die Basisoperation lautet:

$$\text{Abtastung}[x(t) * \text{TP}(t)]$$

wobei $\text{TP}(t)$ eine gefensterte Sinc-Funktion ist und "*" die Faltung symbolisiert.

Die Eingangsfunktion $x(t)$ bildet die ideale Impulsfolge mit der gewünschten Periodendauer T_1 :

$$x(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_1)$$

Dieses Signal wird nun auf die Abtastperiode T_s übergeführt und dabei durch Faltung mit der Impulsantwort eines Tiefpaßfilters interpoliert. Die Grenzfrequenz des Tiefpaßfilters liegt - um Aliasing zu unterbinden - bei $f_g \leq f_s / 2$. Die Impulsantwort des zeitkontinuierlichen Filters wird mit einer Periode $T_2 < T_s$ abgetastet und auf L Werte gefenstert, wobei durch die Symmetrie der Impulsantwort (Sinc-Funktion) nur $L/2$ Werte gespeichert werden müssen. Die Faltung erfolgt nun derart, daß die gefensterte Sinc-Funktion auf das zu berechnende Ausgangssample gesetzt wird, und die sich in diesem Bereich befindlichen Impulse der Eingangsfunktion mit den entsprechenden Werten der Sinc-Funktion gewichtet aufsummiert werden. Um die Rechengenauigkeit weiter zu erhöhen, können die Impulse innerhalb der Sinc-Funktionswerte ebenfalls noch interpoliert werden. Bild 28 veranschaulicht dieses Prinzip.

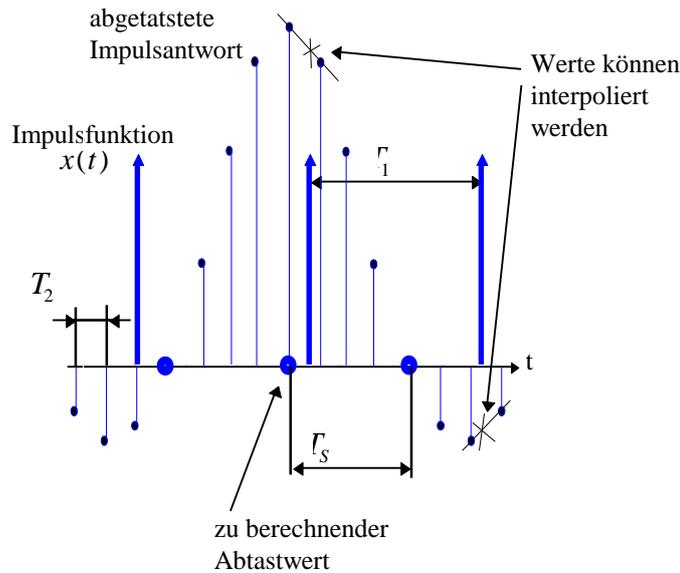


Bild 28: Prinzip der Abtastratenwandlung zur Erzeugung tiefpaßgefilterter Impulsfolgen.

Zur Berechnung selbst wird eine ideale periodische Impulsfolge mit der Periode T_1 in ein Speicherfeld der Länge L geschrieben und schließlich für jedes zu berechnende Ausgangssample mit der Impulsantwort des Filters gefaltet. Danach wird der Inhalt des Speicherfeldes um die Länge eines Samples T_s weitergeschoben und eventuell neue Impulse mit der aktuellen Frequenz hinzugefügt. Dabei wird in einem Register die Lage des zuletzt ins Speicherfeld geschriebenen Impulses ebenfalls als Zeitwert (T_p) gespeichert, um die Position des nächsten Impulses berechnen zu können.

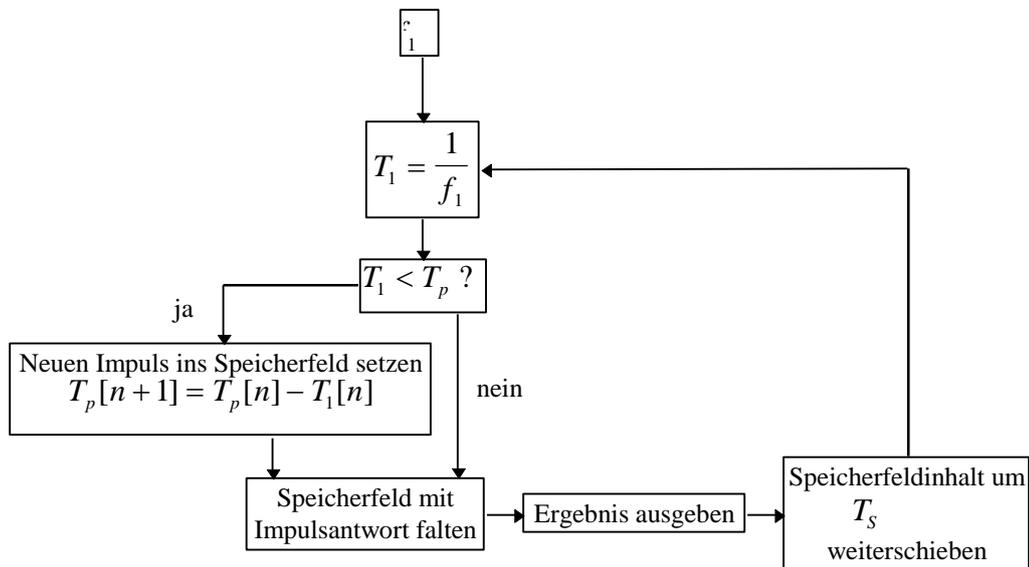


Bild 29: Algorithmus zur Erzeugung einer tiefpaßgefilterten Impulsfolge nach der Methode mittels gefensterter Sinc-Funktion.

Bild 29 illustriert den Algorithmus. Das Ergebnis gibt Bild 30 wieder. Beim ersten Spektrum wurden 8 Nulldurchgänge der Sinc-Funktion bei der Fensterung berücksichtigt. Das zweite Spektrum berücksichtigt 32 Nulldurchgänge der Sinc-Funktion. Zusätzlich wurde die Grenzfrequenz des Filters herabgesetzt.

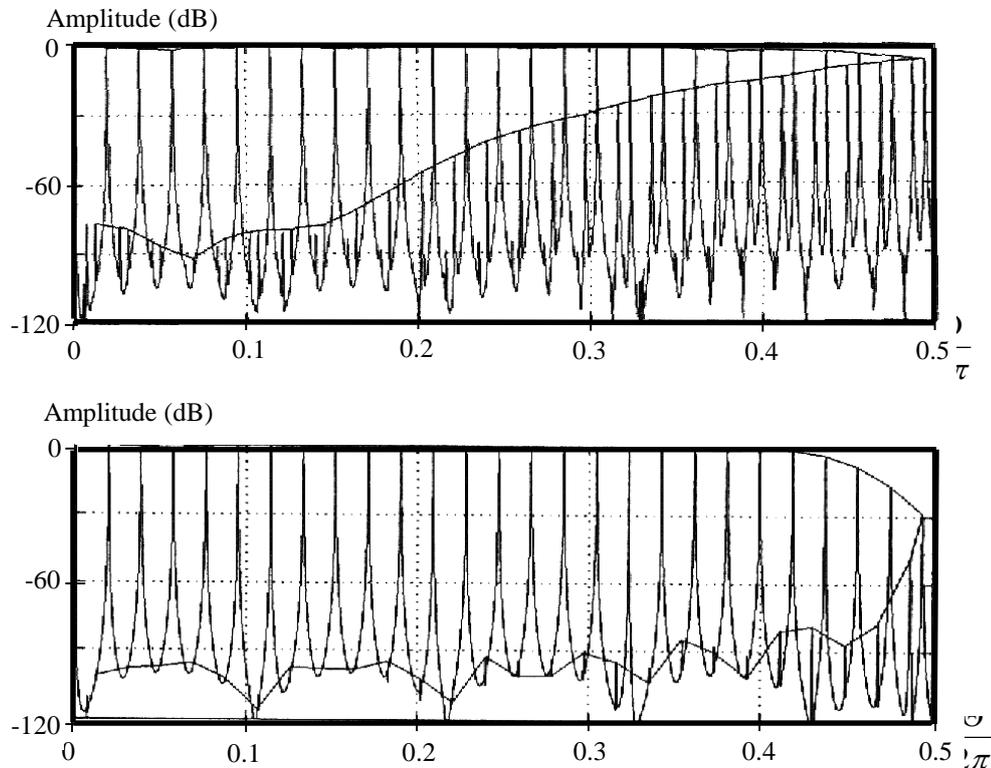


Bild 30: Spektrum einer durch die Fenstermethode tiefpaßgefilterten Impulsfunktion. Die Sinc-Funktion wurde über 8 (oben) bzw. 32 (unten) Nulldurchgänge gefenstert. Die durchgezogene Linie zeigt die Einhüllende. Bild nach [6].

3.1.2.1.5 Erweiterungen zur Generierung von Rechteck-, Dreieck- und Sägezahnfolgen

Wie bereits erwähnt wurde, läßt sich aus der bipolaren tiefpaßgefilterten Impulsfolge mittels Integration eine Rechteckfolge und durch eine weitere Integration eine Dreieckfolge erzeugen, ohne daß Aliasing auftritt. Dabei sind jedoch zusätzliche Algorithmen notwendig, um eine von Frequenz, Phase und Tastverhältnis unabhängige Amplitude zu gewährleisten. Um einen Überlauf zu verhindern, dürfen die Integratoren keine Gleichanteile erhalten. In der Praxis werden jedoch die Pole der Integratoren leicht innerhalb des Einheitskreises gesetzt, um Gleichanteile durch Rundungsfehler zu eliminieren.

Ausgehend von einer tiefpaßgefilterten Impulsfolge $pulse[n]$ sollen nun die erweiterten Kurvenformen hergeleitet werden.

Bipolare Impulsfolge

Die Erzeugung einer bipolaren Impulsfolge $bppulse[n]$ lässt sich mittels

$$bppulse[n] = C_0 \left(pulse[n] - pulse[n - n_{t_v}] \right)$$

realisieren, wobei sich mit n_{t_v} über die Beziehung

$$n_{t_v} = \text{int} \left(\frac{f_s}{f_1} \frac{t_v}{100} \right)$$

eine Steuerung des Tastverhältnisses vornehmen lässt. Aufgrund der Überlagerung der beiden Impulse kommt es zu einer Amplitudenverzerrung in Abhängigkeit vom Tastverhältnis, was in Bild 31 gezeigt wird. Die Periode der Oszillation hängt von der Grundfrequenz ab. Dieses Phänomen tritt vor allem bei bipolaren Impulsfolgen, die mittels diskreter Summenformel generiert wurden auf. Bei der Methode mit der gefensternten Sinc-Funktion beschränkt sich dieser Effekt auf hohe Grundfrequenzen, da es erst dort zur Überlagerung der benachbarten Impulse kommt.

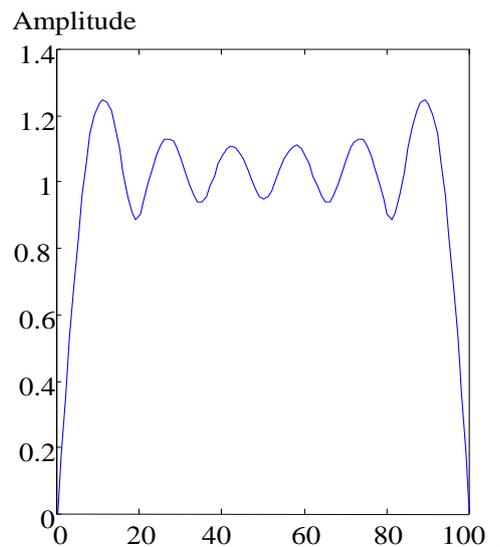


Bild 31: Amplitudenmaximum der bipolaren Impulsfolge in Abhängigkeit vom Tastverhältnis bei einer Berücksichtigung von 9 Harmonischen.

Eine genaue Korrektur kann durch Multiplikation mit dem Faktor

$$C_0 = \frac{1}{\frac{4}{\pi} \sum_{k=0}^{\text{int}(\frac{f_s}{4f_1})} \frac{1}{(2k+1)} \sin\left((2k+1) \frac{\pi t_v}{100}\right)}$$

vorgenommen werden. Da der Effekt jedoch erst bei hohen Grundfrequenzen und extremen Tastverhältnissen um $t_v \approx 0$ bzw. $t_v \approx 100$ in Erscheinung tritt, kann auf eine Korrektur auch verzichtet werden. (Ausnahme: siehe Sägezahnfolge)

Rechteckfolge

Eine Rechteckfolge $rect[n]$ läßt sich über die Beziehung

$$rect[n] = \frac{1}{C_1[n]} \sum_{k=0}^n pulse[k - k_0] - pulse[k - k_{t_v} - k_0] - C_2[n]$$

generieren. k_0 ist die Anzahl von Abtastwerten, um welche die Impulsfolge zeitversetzt beginnt. Für $k_0 = 0$ wird angenommen, daß die Impulsfunktion mit dem positiven Maximum startet. Da die beiden Impulsfolgen in Summe 0 ergeben, kann der Integrator nicht übersteuert werden. Die Amplitude ändert sich in Abhängigkeit von der Frequenz und muß mittels C_1 normiert werden. Je nach Größe von k_0 ergibt sich ein Gleichanteil am Ausgang, welcher mittels C_2 kompensiert werden kann. Für die Koeffizienten gilt:

$$C_1 = \sum_{n=0}^{\text{int}\left(\frac{f_s}{2f_1}\right)} pulse[n]$$

$$C_2 = \sum_{n=0}^{k_0} pulse[n]$$

Dreieck- und Sägezahnfolge

Die Dreieckfolge $tri[n]$ läßt sich durch eine Integration der Rechteckfolge regenerieren:

$$tri[n] = \frac{1}{C_3[n]} \sum_{k=0}^n rect[k] - C_4[n]$$

Um die Extrema der Folge wieder auf 1 zu normieren, muß eine frequenz- und tastverhältnisabhängige Korrektur vorgenommen werden.

$$C_3 = (t_v / 100)(1 - t_v / 100) \frac{f_s}{f_1}$$

$$C_4 = \sum_{n=0}^{\text{int}\left(\frac{f_s}{2f_1}\right)} rect[n] + \sum_{n=0}^{k_0} rect[n]$$

Auch die Sägezahnfolge kann durch eine Dreieckfolge realisiert werden, wobei das Tastverhältnis nahezu 100 beträgt.

Um die oben erwähnten Amplitudenverzerrungen zu reduzieren, ist eine näherungsweise Korrektur der bipolaren Impulsfolge mit dem Faktor

$$C_0 = \begin{cases} \frac{80f_1}{f_s} \dots f_1 > \frac{f_s}{80} \\ 1 \dots \dots \dots \text{sonst} \end{cases}$$

notwendig, wobei ein Tastverhältnis von $t_v = 99$ angenommen wurde.

Bild 32 veranschaulicht die Erzeugung der erwähnten Kurvenformen, wobei als Eingangssignal zwei tiefpaßgefilterte Impulsfolgen verwendet wurden.

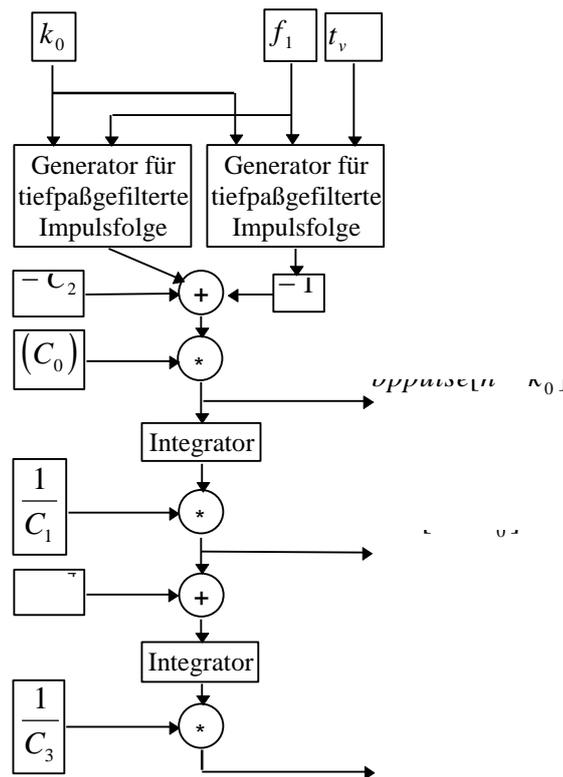


Bild 32: Algorithmus zur Erzeugung einer Rechteck- und Dreieckfolge aus einer tiefpaßgefilterten Impulsfolge.

3.1.2.1.6 Approximation im Zeitbereich

Eine Methode zur Annäherung einer tiefpaßgefilterten Rechteckfolge besteht darin, lediglich den Bereich um die Unstetigkeitsstelle an eine tiefpaßgefilterte Rechteckfolge anzugleichen. Dabei wird jener Bereich durch Addition mit einem Differenzsignal, welches aus der Subtraktion der tiefpaßgefilterten von der idealen Rechteckfolge gewonnen wurde, geglättet. Bild 33 erläutert das Prinzip.

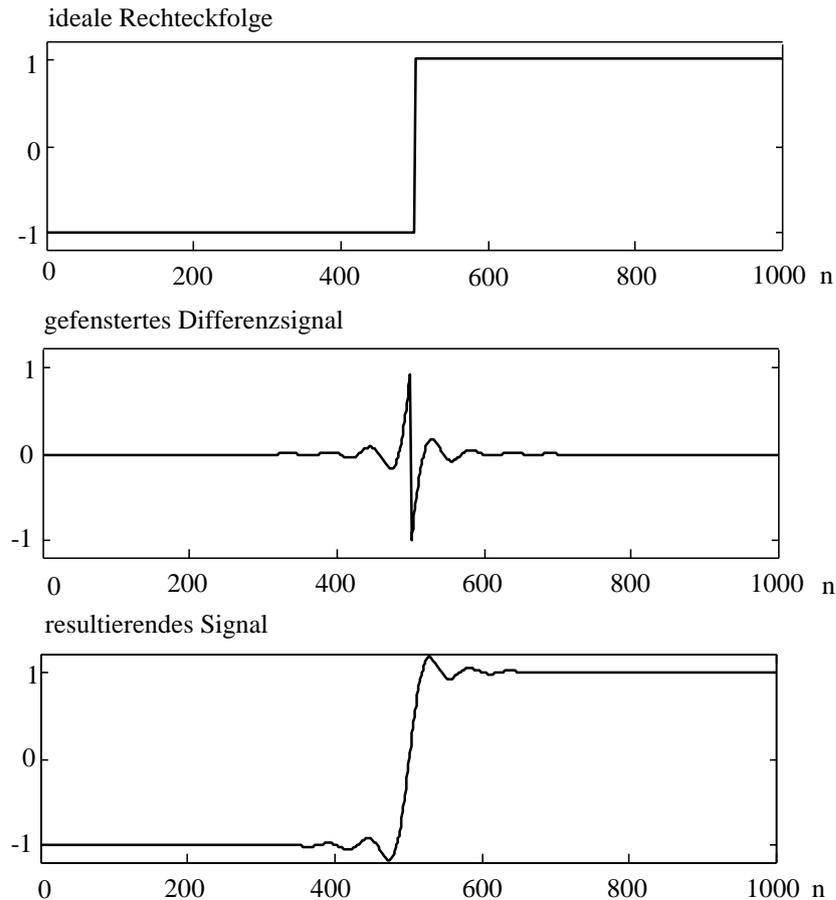


Bild 33: Prinzip der Erzeugung eines tiefpaßgefilterten Rechtecksignals mittels Approximation im Zeitbereich.

Um einen stetigen Übergang zu gewährleisten, wird das Differenzsignal mit einem Hanningfenster multipliziert. Bei hohen Frequenzen kann man dadurch zusätzlich die Überlappung der Differenzsignale verhindern, indem die Länge des Hanningfensters mit der Periodendauer gekoppelt wird.

Der Algorithmus läßt sich auf weitere Kurvenformen wie Dreieck- und Sägezahnfolgen erweitern. Schließlich läßt sich auch die Frequenzcharakteristik des resultierenden Signals durch die Wahl des Fensters beeinflussen. Bild 34 zeigt den Algorithmus. Die resultierende Folge wird einfach durch Addition zweier Wavetables, wovon die zweite zeitversetzt startet, berechnet. Zusätzlich können wiederum die in Kapitel 3.1.1.2 besprochenen Interpolationstechniken verwendet werden. Der Rechenaufwand beträgt für eine Periode des Ausgangssignals bei einer Interpolationstabelllänge von L :

- L Additionen
- $(T_1 - L) / 2$ -Warteschleife
- 1 Division
- Generation einer k -Punkte-Wavetable für idealen Funktionsgenerator (siehe Kap. 3.1.1.2)
- $k + L + 2$ Speicherplätze

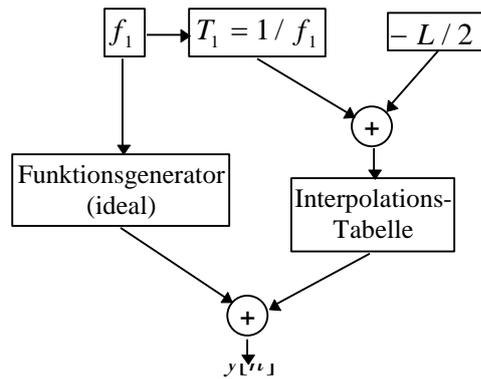


Bild 34: Algorithmus zur Berechnung eines tiefpaßgefilterten Signals mittels Approximation im Zeitbereich.

3.1.2.2 Additive Synthese im Frequenzbereich

Inverse Fourier-Transformation

Da die spektralen Komponenten der gewünschten Wellenformen bekannt sind, lassen sich diese auch im Frequenzbereich erzeugen.

Ausgehend von der Short-Time Fourier-Analyse wird das Signal im Zeitbereich in einzelne Frames unterteilt und dessen Spektrum berechnet. Durch die Fensterung wird das Signal im Frequenzbereich mit der Fourier-Transformierten der Fensterfunktion gefaltet.

$$w[n] \sum_{k=1}^N a_k \cos(\Theta_k n + \psi_k) \xrightarrow{FFT} W(\Theta) * \sum_{k=1}^N a_k e^{j\psi_k} \delta(\Theta - \Theta_k)$$

wobei $w[n]$...Fensterfunktion im Zeitbereich

$W(\Theta)$..Fensterfunktion im Frequenzbereich

Aus dieser Überlegung läßt sich auch umgekehrt im Frequenzbereich ein beliebiges Spektrum durch Faltung mit der Fensterfunktion erzeugen und anschließend mittels inverser Fourier-Transformation in ein Zeitsignal umwandeln (siehe [12]).

Überlappende Synthese

Bei der Short-Time Fourier-Transformation wird angenommen, daß sich die Signaleigenschaften während der Dauer eines Frames nicht ändern, was sich jedoch als unzureichend erweist und Verzerrungen zur Folge hat. Um Verzerrungen aufgrund der Amplitudenunterschiede zu vermindern, werden die Fenster im Zeitbereich überlappend angeordnet und ineinander übergeblendet [11]. Das Fenster soll somit zwei wesentliche Eigenschaften erfüllen; einerseits soll es im Frequenzbereich eine ausgeprägte Hauptkeule aufweisen und andererseits im Zeitbereich im sich überlappenden Intervall die Summe 1 ergeben.

In der Praxis werden jedoch zwei Fenster verwendet:

Zuerst wird ein Fenster $w[n] \leftrightarrow W(\Theta)$ mit der ersten Eigenschaft für die Faltung im Frequenzbereich verwendet. Im Zeitbereich werden die Frames mittels Dreieckfenstern $\Delta[n]$ - welche letztere Eigenschaft erfüllen - überlappend angeordnet, wodurch sich in der Summe immer der Wert 1 ergibt, sofern die Überlappung $L/2$ Abtastwerte bei einer Fensterlänge von L Abtastwerten beträgt. Um die Verzerrungen von $w[n]$ zu beseitigen, muß das Signalframe nach der inversen Fourier-Transformation mit dem Fenster

$$w'[n] = \frac{\Delta[n]}{w[n]}$$

gewichtet werden.

Phasenkorrektur

Durch die unterschiedlichen Frequenzen zweier benachbarter Frames kann es bei der Überblendung zu Amplitudenverzerrungen aufgrund von Phasenauslöschungen kommen.

Durch Gleichsetzen der Phase der Schwingungen benachbarter Frames in der Mitte der Überlappung kann dieser Fehler minimiert werden (siehe [13]). Mathematisch formuliert lautet die Bedingung

$$\cos\left(\Theta_{k,i} \frac{3L}{4} + \psi_{k,i}\right) = \cos\left(\Theta_{k,i+1} \frac{L}{4} + \psi_{k,i+1}\right)$$

wobei i den Frameindex symbolisiert. Die resultierende Phase beträgt dann:

$$\psi_{k,i+1} = \psi_{k,i} + \frac{L}{4} (3\Theta_{k,i} - \Theta_{k,i+1})$$

Chirp-Synthese

Eine weitere Möglichkeit, die Verzerrungen aufgrund der unterschiedlichen Frequenzen zu vermindern, besteht darin, die Frequenz innerhalb eines Frames nicht als konstanten Wert, sondern als eine lineare Funktion anzusehen (Chirp). Ein Frame enthält also für jeden Teilton einen Chirp, welcher eine bezüglich der Frequenz linear steigende oder fallende Sinusschwingung repräsentiert, bzw.:

$$y_i[n] = \sum_{k=1}^N \cos((\Theta_{1k,i} + \alpha_{k,i}n)n + \psi_{k,i}) = \sum_{k=1}^N \cos(\Theta_{1k,i}n + \alpha_{k,i}n^2 + \psi_{k,i})$$

$\Theta_{1k,i}$ entspricht der Frequenz am Anfang des Frames und α der jeweiligen Frequenzänderung pro Abtastwert. α läßt sich aus den Anfangsfrequenzen der beiden benachbarten Frames herleiten:

$$\alpha_{k,i} = \frac{\Theta_{1k,i+2} - \Theta_{1k,i}}{L}$$

Schließlich ergibt sich für die zu berücksichtigende Phase folgende Beziehung:

$$\psi_{k,i+1} = \psi_{k,i} + \frac{L}{4} (3\Theta_{1k,i} - \Theta_{1k,i+1}) + \left(\frac{L}{4}\right)^2 (9\alpha_{k,i} - \alpha_{k,i+1})$$

Durch die Annahme der sich linear ändernden Frequenzen der Teiltöne ergibt sich auch für das im Frequenzbereich zur Faltung verwendete Fenster $W(\Theta)$ eine Abhängigkeit von α . In der Praxis wird $W(\Theta, \alpha)$ für einen Satz von $\{\alpha_p\}$ Werten berechnet und gespeichert. Das resultierende Fenster selbst wird durch lineare Interpolation der Werte von $W(\Theta, \alpha_p)$ und $W(\Theta, \alpha_{p+1})$ berechnet.

Zur Regenerierung periodischer Signale vereinfacht sich der Algorithmus derart, daß sich die Werte von α für die einzelnen Harmonischen nur durch einen ganzzahligen Faktor unterscheiden. Darüberhinaus können die Formeln zur Berechnung der Fourierkoeffizienten für die verschiedenen Kurvenformen in geschlossener Form gespeichert werden. Bild 35 zeigt das Flußdiagramm eines Algorithmus zur Berechnung eines periodischen Ausgangssignals.

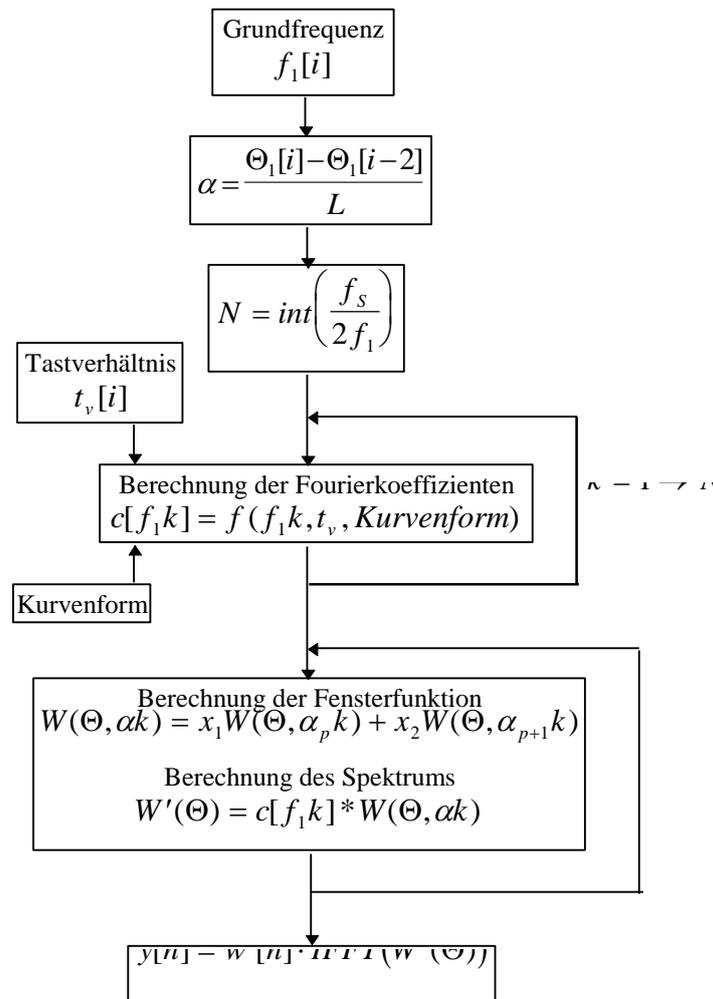


Bild 35: Algorithmus zur Erzeugung eines periodischen Signals mittels inverser Fourier-Transformation.

3.1.2.3 Subtraktive Synthese im Frequenzbereich

Algorithmus

[14] präsentiert eine Methode, eine Sägezahnswingung aus einer Sinusschwingung zu approximieren.

Das gewünschte Spektrum der Sägezahnswingung hat folgende Fourier-Reihe:

$$y_{sz}(x) = \frac{1}{\pi} \left(\frac{\sin(x)}{1} + \frac{\sin(2x)}{2} + \frac{\sin(3x)}{3} + \dots \right)$$

Ausgehend von der Betragsfunktion einer Sinusschwingung, deren Fourier-Reihe gegeben ist durch

$$\left| \sin\left(\frac{x}{2}\right) \right| = \frac{2}{\pi} - \frac{4}{\pi} \left(\frac{\cos(x)}{3} + \frac{\cos(2x)}{3 \cdot 5} + \frac{\cos(3x)}{5 \cdot 7} + \dots \right)$$

werden nun deren spektrale Komponenten durch ein dynamisches Hochpaßfilter erster Ordnung verändert. Die Grenzfrequenz f_g des Filters wird $f_g = 16f_1$ gewählt. Dadurch werden die ersten sechzehn Harmonischen der Betragsfunktion um 6 Dezibel pro Oktave angehoben, da sie in der steigenden Flanke des Filters liegen. Als Ergebnis erhält man folgende Fourier-Reihe:

$$\begin{aligned} HPF \left\{ \left| \sin\left(\frac{x}{2}\right) \right| \right\} &= \frac{f_1}{\pi f_g} \left(\frac{1 \sin(x)}{(1+1/2)(1-1/2)} + \frac{2 \sin(2x)}{(2+1/2)(2-1/2)} + \frac{3 \sin(3x)}{(3+1/2)(3-1/2)} + \dots \right) \approx \\ &\approx \frac{f_1}{\pi f_g} \left(\frac{\sin(x)}{1} + \frac{\sin(2x)}{2} + \frac{\sin(3x)}{3} + \dots \right) \end{aligned}$$

Das Hochpaßfilter muß also dynamisch mit der Grundfrequenz mitgesteuert werden. Die Dämpfung des Filters wird durch eine Verstärkung um den Faktor 16 ausgeglichen. Um für die digitale Implementierung die Aliasing-Komponenten weiter zu unterdrücken, wird zusätzlich noch ein Tiefpaßfilter mit konstanter Grenzfrequenz eingefügt. Bild 36 illustriert den Algorithmus.

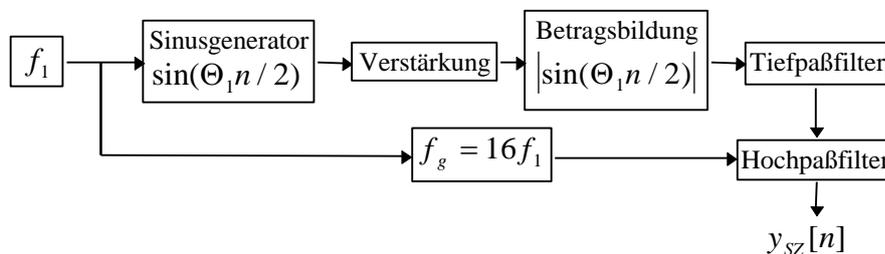


Bild 36: Algorithmus zur Approximation einer Sägezahnfolge mittels subtraktiver Synthese.

Hochpaßfilter

Das Hochpaßfilter erster Ordnung wird durch folgende Differenzgleichung beschrieben:

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

Die Filterkoeffizienten berechnen sich zu:

$$a = \frac{1+b}{2} \quad b = \frac{\cos(\Theta_g)}{1 + \sin(\Theta_g)}$$

Die Filterung erfordert also folgenden Rechenaufwand:

- 2 trigonometrische Berechnungen
- 2 Multiplikationen
- 4 Additionen
- 1 Shift-Operation
- 1 Division

Um den Rechenaufwand für das dynamische Filter zu verringern, wird in [14] eine Approximation der Filterkoeffizienten mittels Taylor-Reihe vorgenommen. Die Entwicklung erfolgt für die zwei Grundfrequenzbereiche $0 \leq \Theta_g < \pi/2$ und $\pi/2 \leq \Theta_g < \pi$ mit den entsprechenden Entwicklungspunkten $\Theta_g = \pi/4$ und $\Theta_g = 3\pi/4$.

Es werden vier Glieder berücksichtigt, wodurch sich die Filterkoeffizienten wie folgt berechnen lassen:

$$a = \frac{1+b}{2} \quad b = c_0 + c_1\Theta_g + c_2\Theta_g^2 + c_3\Theta_g^3$$

Der Rechenaufwand verringert sich auf:

- 8 Multiplikationen
- 5 Additionen
- 1 Shift-Operation
- 1 if-Anweisung

Erweiterungen für Rechteck- und Dreieckfolgen

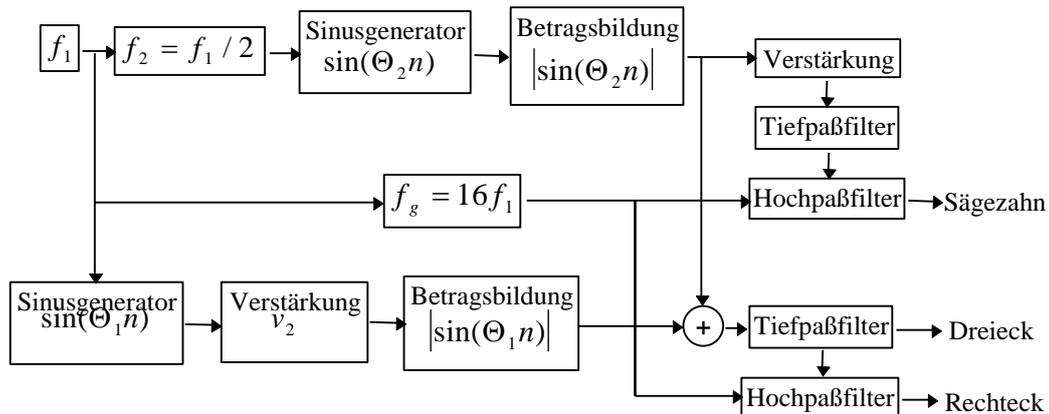


Bild 37: Algorithmus zur Approximation einer Sägezahn-, Dreieck- und Rechteckfolge mittels subtraktiver Synthese.

Bild 37 zeigt eine Erweiterung des Verfahrens zur zusätzlichen Erzeugung von Dreieck- und Rechteckfolge. Durch die Hinzunahme eines zweiten Sägezahngenerators mit der doppelten Grundfrequenz läßt sich durch Addition beider Folgen eine Beeinflussung auf die geradzahigen Harmonischen vornehmen. Über den Verstärkungsfaktor des zweiten Generators kann die Stärke dieser Komponenten variiert werden, wobei sich bei einer Auslöschung (Verstärkungsfaktor $v_2 \approx -0.2$) der geradzahigen Harmonischen eine Rechteckfolge realisieren läßt. Zusätzlich ergibt sich durch Auskoppeln des Signals vor dem Hochpaß eine Annäherung an eine Dreieckfolge.

Durch geringes Hinzufügen von geradzahigen Harmonischen läßt sich auch eine Pulsweitenmodulation simulieren (siehe [14]).

Da die Betragswerte der Sinusfolge einfach durch Ignorieren des Vorzeichens erhalten werden, wird hierfür kein Rechenaufwand angenommen. Für die Hochpaßberechnung wird die Taylor-Reihenentwicklung verwendet. Der gesamte Algorithmus benötigt somit:

- 2 Sinusberechnungen
- 2 Tiefpaßberechnungen
- 18 Multiplikationen
- 11 Additionen
- 7 Shift-Operationen
- 1 if-Anweisung

3.2 Rauschgenerator

Rauschgeneratoren liefern Zufallssignale, deren Charakteristik mit statistischen Mittelwerten beschrieben wird. Die für Audiosignale bedeutsamste Eigenschaft ist hierbei neben der Leistung das Spektrum.

Die Generierung solcher Signale gliedert sich in zwei Abschnitte:

- Generierung einer zufälligen Zahlenfolge (weißes Rauschen)
- Beeinflussung des Leistungsdichtespektrums (farbiges Rauschen)

3.2.1 Weißes Rauschen

Eine Zahlenfolge wird als zufällig bezeichnet, wenn deren Werte unabhängig voneinander auftreten, d. h., von den bisher bekannten Werten der Folge kann nicht auf den nächsten Wert geschlossen werden.

Die Wahrscheinlichkeit, mit der ein bestimmter Wert auftritt, soll für alle Werte gleich sein; es wird also eine Amplitudengleichverteilung vorausgesetzt.

Aufgrund der digitalen Implementierung ergibt sich für die zu berechnende Zahlenfolge ein endlicher Wertebereich, welcher zwischen zwei Extremwerten liegt.

Das Leistungsdichtespektrum solcher Zahlenfolgen zeichnet sich durch eine frequenzunabhängige Charakteristik aus, welche als weißes Rauschen bezeichnet wird.

3.2.1.1 Abtasten eines Zufallssignals

Zufallsfolgen können durch Abtasten eines externen Zufallssignals, welches über einen physikalischen Zufallsprozeß (z. B. thermisches Rauschen) generiert wird, gewonnen werden. Nachteilig ist hierbei jedoch der zusätzliche Hardware-Aufwand.

3.2.1.2 Pseudo-Zufallszahlen-Generator

Eine weitere Möglichkeit bietet die Erzeugung von sogenannten Pseudo-Zufallszahlen, welche die Zahlenfolge direkt in der digitalen Ebene generieren. Da diese Generatoren auf einem deterministischen Algorithmus basieren, und die Zahlenfolge somit vorhersehbar ist, spricht man von Pseudo-Zufallszahlen-Generatoren.

Lineare Kongruenzmethode

Ein Verfahren dieser Art ermöglicht die lineare Kongruenzmethode. Dabei wird die Zahlenfolge $x[n]$ nach folgender Vorschrift berechnet:

$$x[n+1] = (ax[n] + b) \text{ modulo}(c)$$

mit $n \geq 0$...Index

$a \geq 0$...Multiplikator

$b \geq 0$...Inkrement

$c \geq x[0]$...Modulus

weitere muß gelten: $(c > a) \wedge (c > b)$

Diese Berechnung liefert eine periodische Zahlenfolge, deren Periodendauer und Wertebereich von der speziellen Wahl der Konstanten abhängt. Insbesondere wird die maximale Länge der Sequenz und die Anzahl der maximal möglichen Werte durch c vorgegeben. Der Wert $x[0]$ repräsentiert dabei den Startwert der Sequenz.

Durch ein großes c (z. B.: $c = 2^{31}$) und eine geeignete Wahl der übrigen Konstanten läßt sich somit eine periodische Zahlenfolge generieren, deren Grundfrequenz so klein ist, daß deren spektralen Komponenten so dicht gestaffelt sind, daß man näherungsweise von einem kontinuierlichen Spektrum sprechen kann, wie es für weißes Rauschen definiert ist. Der Rechenaufwand zur Generierung von weißem Rauschen beträgt also:

- 1 Division
- 1 Multiplikation
- 1 Addition
- 4 Speicherplätze

3.2.2 Farbiges Rauschen

3.2.2.1 Brownsches Rauschen

Das Leistungsdichtespektrum von Brownschem Rauschen hat eine $1/f^2$ -Charakteristik. Der Wert nimmt also um 6 Dezibel pro Oktave ab, wodurch sich dieses Rauschen einfach durch einen Integrator erzeugen läßt. In der Praxis wird ein Tiefpaß 1. Ordnung mit einer Grenzfrequenz im Infrarotbereich ($f_g < 16\text{Hz}$) verwendet.

Mit der Tiefpaßberechnung

$$y[n] = ax[n] - by[n-1]$$

wobei sich die Koeffizienten wie folgt berechnen lassen:

$$a = 1 + b$$

$$b = \sqrt{(2 - \cos(\Theta_g))^2 - 1 - 2 + \cos(\Theta_g)}$$

Man benötigt für das Brownsche Rauschen also folgenden Rechenaufwand:

- 1 Division
- 3 Multiplikationen
- 2 Additionen
- 5 Speicherplätze

3.2.2.2 Rosa Rauschen

Das Leistungsdichtespektrum des rosa Rauschens zeichnet sich durch eine $1/f$ -Charakteristik aus. Die Leistungsdichte nimmt also um 3 Dezibel pro Oktave ab.

3.2.2.2.1 Approximation von rosa Rauschen

Gardner-Methode

Ein annähernde $1/f$ -Charakteristik läßt sich durch Summation von mehreren Rauschgeneratoren, welche mit unterschiedlichen Taktraten getriggert werden erreichen. Bild 38 illustriert das Prinzip.

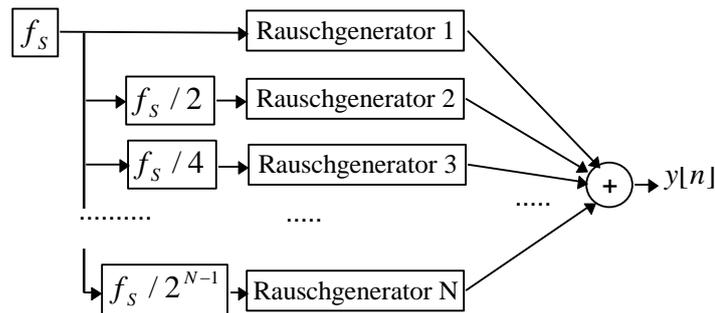


Bild 38: Approximation von rosa Rauschen.

Durch die unterschiedlichen Taktraten bleibt der Einfluß der langsamer erzeugten Zufallswerte auf das Gesamtergebnis für längere Zeit erhalten, wodurch sich die für die $1/f$ -Charakteristik typische zeitliche Abhängigkeit von vorhergehenden Werten ergibt ([15]).

Die Realisierung der N Generatoren kann mittels einem Generator mit der linearen Kongruenzmethode erfolgen, wobei voneinander unabhängige Sequenzen durch verschiedene Startwerte erreicht werden. Der durchschnittliche Rechenaufwand pro Sample beträgt bei N Generatoren:

- k Divisionen
- k Multiplikationen
- $k+N-1$ Additionen
- $3+N$ Speicherplätze

wobei: $k = \frac{2^{N+1} - 1}{N}$

IIR-Filter

Eine weitere Möglichkeit zur Approximation von rosa Rauschen besteht in der Verwendung eines IIR-Filters.

Hierbei werden die Pol- und Nullstellen auf der reellen Achse abwechselnd positioniert. Man beginnt in der Nähe des Punktes $z = 1$ mit einer Polstelle. Dabei ergibt sich im Bereich der Polstelle eine flache Frequenzcharakteristik, welche durch die dahinter positionierte Nullstelle eine leichte negative Steigung erfährt. Diese Nullstelle verursacht jedoch im höheren Frequenzbereich eine zu große Steigung, welche wiederum durch eine Polstelle kompensiert wird, usw...

Durch eine geeignete Wahl der Pol- und Nullstellenpositionen läßt sich auf diese Weise die gewünschte Frequenzcharakteristik approximieren.

3.2.2.2 Rosa Rauschen mittels Modifikation im Frequenzbereich

Eine weitere Methode ergibt sich durch Bearbeiten des Signals im Frequenzbereich. Dabei wird das Signal von der Zeitdarstellung in eine spektrale Leistungsdichte umgerechnet und dort mit der gewünschten Spektralverteilungsfunktion multipliziert. Schließlich wird das Signal wieder in den Zeitbereich übergeführt (siehe [15]).

Während der Rechenaufwand relativ hoch ist, bietet diese Methode den Vorteil, Rauschen mit beliebiger spektraler Dichteverteilung zu generieren. Mit der Vorschrift

$$y[n] = w[n] \cdot \text{IFFT} \left\{ \text{Zielspektrum} \cdot (\cos(x[n]) + j \sin(x[n])) \right\}$$

wobei $x[n]$ weißes Rauschen und $w[n]$ das für die überlappende Synthese verwendete Fenster im Zeitbereich ist, ergibt sich bei einer Framelänge von L Abtastwerten ein Rechenaufwand von:

- Erzeugung von weißem Rauschen (siehe 3.3.1)
- $2L$ trigonometrische Berechnungen
- $2L$ Multiplikationen
- $2L$ Additionen
- IFFT-Algorithmus

3.3 Linear-Exponential-Konverter

Der Zusammenhang der vom Menschen empfundenen Lautheit eines Klages mit der Signalamplitude ist nichtlinear. Als Näherung liefert der Lautstärkepegel L in Dezibel eine Größe, welche mit der Signalamplitude x_1 und der Bezugsgröße x_0 über den dekadischen Logarithmus verbunden ist:

$$L = 20 \log_{10} \left(\frac{x_1}{x_0} \right) \quad [\text{dB}]$$

Ein ähnlicher Zusammenhang besteht in der Empfindung der Tonhöhe. Hier ist die relative Tonhöhe in Cent mit der Frequenz f_1 und der Bezugsgröße f_0 über den dualen Logarithmus verbunden.

$$\text{Rel. Tonhöhe} = 1200 \log_2 \left(\frac{f_1}{f_0} \right) = \frac{1200}{\log_{10}(2)} \log_{10} \left(\frac{f_1}{f_0} \right) \quad [\text{Cent}]$$

Die Transferfunktionen für Lautstärke und wahrnehmbare Tonhöhe unterscheiden sich also nur durch einen konstanten Faktor.

Linear-Exponential-Konverter dienen dazu, Steuersignale in ein exponentielles Maß überzuführen und damit die Signalamplitude bzw. die Frequenz entsprechend zu verändern. Das ursprüngliche Steuersignal repräsentiert durch diese Umwandlung die Lautstärke bzw. die relative Tonhöhe.

Die Konvertierung kann über die Table-Lookup-Methode realisiert werden. Dabei werden die Funktionswerte der Transferfunktion mit einer Wortbreite, die der Signalauflösung entspricht, in eine Tabelle geschrieben. Die Größe der Tabelle wird durch die gewünschte Auflösung der Pegelwerte bestimmt. Die Tabelle kann mit den in Kapitel 3.1.1.2 erwähnten Verfahren interpoliert werden. Bei einer Wortbreite der Funktionswerte von 16 Bit würde beispielsweise eine Tabellenlänge von 256 Einträgen zu einer Pegelauflösung von $96\text{dB} / 256 = 0.375\text{dB}$ führen. Bild 39 illustriert die Konvertierung einer Sinusschwingung an der Transferkennlinie, welche sich durch die Umkehrung der Logarithmusfunktion zu einer Exponentialfunktion ergibt.

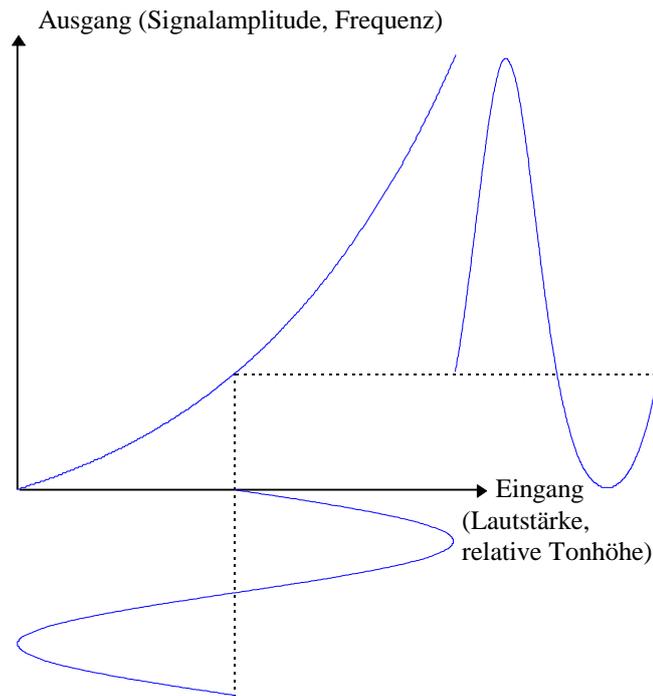


Bild 39: Transferfunktion zur Berechnung des Exponentialwertes bei Ansteuerung mit einem Sinussignal.

3.4 Ringmodulator

3.4.1 Grundprinzip

Ringmodulatoren multiplizieren zwei Signale, wodurch im Ausgangssignal neue Frequenzen entstehen, die sich durch die Faltung der beiden Eingangsspektren ergeben. Sind beide Eingangssignale bandbegrenzt, so entspricht der höchste Frequenzanteil des Ausgangssignals der Summe der beiden höchsten Frequenzanteile der Eingangsspektren. Im allgemeinen haben beide Eingangssignale dieselbe Maximalfrequenz, nämlich die halbe Abtastfrequenz, wodurch sich für das Ausgangssignal Frequenzen bis zur Abtastfrequenz ergeben, was starkes Aliasing verursachen würde.

Daher müssen die beiden Signale vor der Multiplikation einer Abtastraterhöhung - mindestens um den Faktor 2 - unterzogen werden, um Aliasing zu verhindern. Damit das Signal wieder auf die ursprüngliche Abtastrate heruntergesampelt werden kann, muß das Signal nach der Multiplikation mit einem Tiefpaßfilter auf mindestens ein Viertel der höheren Abtastrate bandbegrenzt werden. Bild 40 zeigt das Prinzip und die Spektren der Signale nach den verschiedenen Bearbeitungsschritten.

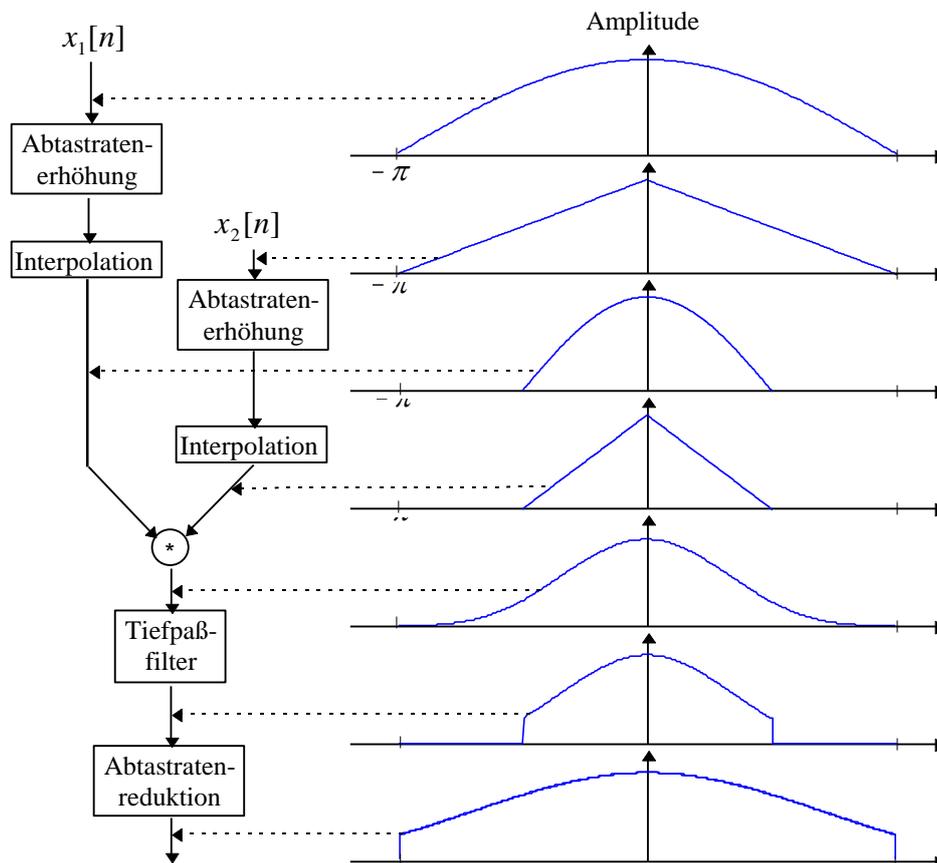


Bild 40: Prinzip eines digitalen Ringmodulators und Beeinflussung der Frequenzspektren.

3.4.2 Ringmodulator mit Halbbandfilter

Im folgenden wird ein Algorithmus vorgestellt, welcher die Ringmodulation auf effiziente Weise mittels Halbbandfiltern realisiert. Aufgrund der nichtidealen Filter ergeben sich hierbei Aliasing-Komponenten, welche jedoch bei ausreichend hohen Abtastraten vernachlässigt werden können. Das Prinzip ist dasselbe, wie es in Bild 40 dargestellt ist, jedoch ergeben sich durch die Verwendung von Halbbandfiltern Vereinfachungen im Algorithmus.

3.4.2.1 Halbbandfilter

Wird die Abtastrate eines Systems um den Faktor q erhöht, so muß die Folge zwischen den ursprünglichen Abtaststellen interpoliert werden. Für die Impulsantwort dieser Filter muß gelten:

$$\begin{aligned} h[m = nq] &= 0 && \text{für } n = 1, 2, 3, \dots \\ h[m = nq] &= 1 && \text{für } n = 0 \end{aligned}$$

Dadurch wird gewährleistet, daß das Signal an den ursprünglichen Abtaststellen seinen Wert beibehält. Solche Filter werden als Interpolatorfilter und - für den speziellen Fall von $q = 2$ - als Halbbandfilter bezeichnet. Obige Bedingung kann nur von FIR-Filtern erfüllt werden. Halbbandfilter sind linearphasige Tiefpaßfilter, deren Übertragungsfunktion punktsymmetrisch zu $\Theta = \pi/2$ ist. Dadurch wird die Impulsantwort für alle geraden Werte gleich Null mit Ausnahme für $m = 0$.

Bei einer Abtastratenerhöhung um den Faktor 2 wird zuerst die Eingangsfolge zwischen den Werten mit Nullen ergänzt, was in Bild 41 der Folge $x[m]$ entspricht. Diese Folge wird nun mit der Impulsantwort des Halbbandfilters gefaltet, wobei sich zwei Fälle unterscheiden lassen:

Wird die Faltung an der Stelle eines ursprünglichen Signalwertes berechnet, so erhält man als Ergebnis genau diesen Wert. Die effektive Impulsantwort mit der gefaltet wurde entspricht also dem Einheitsimpuls (im Bild 41 als $h_1[n]$ bezeichnet).

Wird jedoch die Faltung an einer Stelle berechnet, bei der das Signal mit einer Null ergänzt wurde, so erhält man als effektive Impulsantwort eine symmetrische Folge ($h_2[n]$). Die Berechnung kann also im Takt der niedrigeren Abtastrate erfolgen.

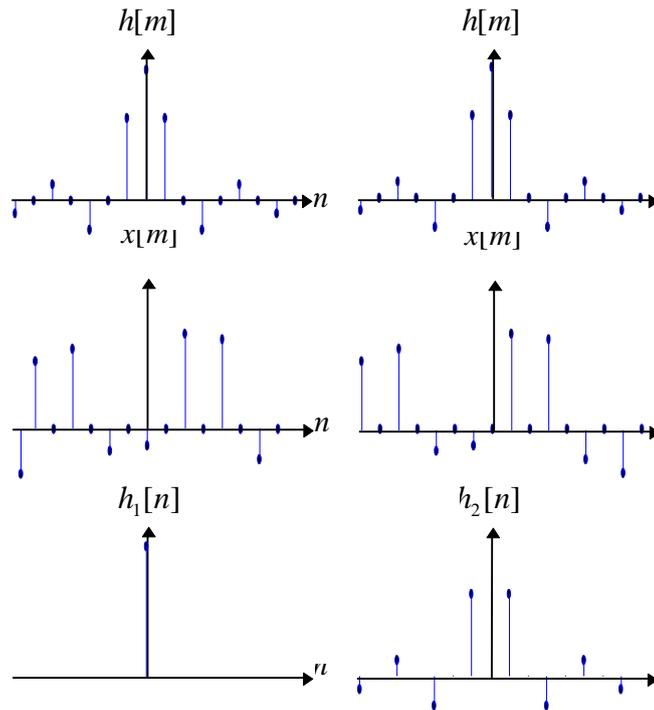


Bild 41: Impulsantwort eines Halbbandfilters $h[m]$ und deren Verwendung in der Faltung mit einer um den Faktor 2 abtastratenerhöhten Eingangsfolge $x[m]$ zu den Zeitpunkten $m = 2n$ und $m = 2n + 1$, wobei $n = 1, 2, 3, \dots$

Die Filterung läßt sich somit auch als die Summe des Originalsignals und der Faltungssumme des Originalsignals mit der Impulsfolge $h_2[n]$ realisieren, wobei letztere Signalwerte zwischen das Originalsignal eingefügt werden. Bild 42 zeigt die Struktur dieses Algorithmus. Um Phasenfehler zu vermeiden, muß das Originalsignal um $N/2$ Abtastwerte verzögert werden, wobei N der Filterordnung entspricht.

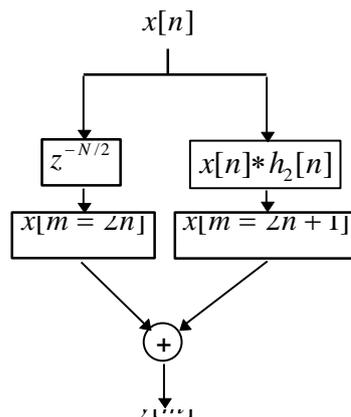


Bild 42: Algorithmus zur Abtastratenerhöhung um den Faktor 2.

3.4.2.2 Algorithmus des Ringmodulators

Unter Verwendung des oben erwähnten Prinzips zur Abtastraterhöhung läßt sich auch der Ringmodulator realisieren. Der Algorithmus ist in Bild 43 dargestellt.

Die Signalwerte der beiden Eingangsfolgen an den Stellen $n = 2m$ und $n = 2m + 1$ werden miteinander multipliziert und anschließend mit einem Halbbandfilter tiefpaßgefiltert. Dabei werden die beiden Eingangsfolgen in ihrer Abtastrate nicht verändert, da bei der nachfolgenden Abtastratenreduktion nur jeder zweite Wert der Folge als Ausgangssignal dient. Das Halbbandfilter vor der Addition berechnet also nur jene Faltungssumme der Folge für die Werte $m = 2n + 1$, welche für die Werte an den Stellen $m = 2n$ einen Beitrag liefern, die schließlich dieser Folge zeitgleich hinzuaddiert werden. Die Gesamtverzögerung beträgt N Abtastwerte, wobei N durch die Länge der Impulsantwort des Halbbandfilters $h_2[n]$ bestimmt wird.

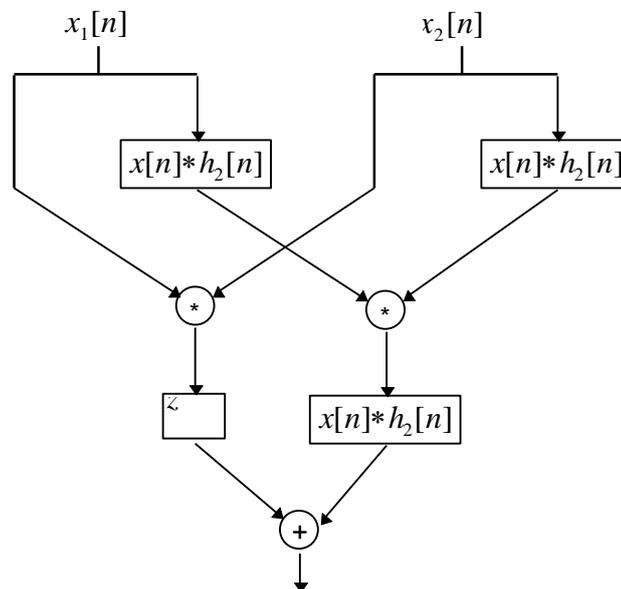


Bild 43: Algorithmus eines Ringmodulators mit Halbbandfilter.

3.5 Spannungsgesteuerter Verstärker (VCA)

Spannungsgesteuerte Verstärker unterscheiden sich im allgemeinen nicht vom Ringmodulator, da die Multiplikation mit dem Verstärkungsfaktor im Frequenzbereich wieder eine Faltung der Spektren ergibt.

VCAs werden aber im speziellen zur Amplitudenmodulation mittels Hüllkurven und langsam schwingenden Oszillatoren (LFOs) verwendet, um die Lautstärke des Signals zu beeinflussen. Da solche Signale einen Gleichanteil besitzen, bleibt am Ausgang das Eingangsspektrum erhalten, was einen weiteren Unterschied zum Ringmodulator darstellt.

Unter der Voraussetzung, daß die höchste Frequenz im Ausgangssignal kleiner als die halbe Abtastfrequenz bleibt, läßt sich diese Funktion mit einer einfachen Multiplikation realisieren, wobei aufgrund des nichtnegativen Steuersignales nur eine 2-Quadranten-Multiplikation erforderlich ist.

3.6 Spannungsgesteuertes Filter (VCF)

Im folgenden werden Algorithmen zur Implementierung von spannungsgesteuerten Filtern gezeigt. Kapitel 3.6.0 gibt einen Überblick über digitale Filterstrukturen. Kapitel 3.6.1 behandelt die Elimination von Transienten im Ausgangssignal des Filters, welche sich durch die zeitlich variierenden Filterkoeffizienten ergeben. Kapitel 3.6.2 bis 3.6.5 befassen sich mit den Filteralgorithmen selbst. Dabei wurde im wesentlichen das spannungsgesteuerte Tiefpaßfilter behandelt, da es in musikalischen Anwendungen am häufigsten verwendet wird und durch Frequenztransformation in andere Filtercharakteristika übergeführt werden kann. Die übrigen Filtercharakteristika (Kapitel 3.6.3-3.6.5) behandeln deshalb nur ausgewählte Algorithmen, die sich für eine zeitvariable Steuerung der Kontrollparameter besonders gut eignen.

3.6.0 Allgemeines

3.6.0.1 FIR-Filter

FIR-Filter (FIR = finite impulse response, endliche Impulsantwort) werden durch folgende Differenzgleichung beschrieben:

$$y[n] = \sum_{k=0}^M a_k x[n-k]$$

Die Differenzgleichung entspricht einer zeitdiskreten Faltung der Impulsantwort $h[n]$ des Filters mit $x[n]$, wobei sich die Impulsantwort zu

$$h[n] = \begin{cases} a_n & n = 0, 1, \dots, M \\ 0 & \text{sonst} \end{cases}$$

ergibt. Durch die rückkopplungsfreie Struktur des Filters erhält man für die Systemfunktion (mit Ausnahme von $z = 0$) lediglich Nullstellen:

$$H(z) = \sum_{n=0}^M h[n]z^{-n} \quad (4)$$

Dadurch wird auch der Vorteil von FIR-Filtern deutlich: Sie sind in jedem Fall stabil. Der Nachteil liegt in der hohen Ordnung der Filter, die notwendig ist, um die selben Anforderungen zu erfüllen, wie sie durch IIR-Filter (siehe Kapitel 3.6.0.2) erzielt werden können, wodurch schließlich auch der Rechenaufwand größer ist.

3.6.0.1.1 Rundungsfehler

Durch die Koeffizientenquantisierung eines FIR-Systems verändert sich dessen Übertragungsfunktion zu

$$H'(z) = H(z) + \Delta H(z)$$

Dabei ist $H'(z)$ die resultierende Systemfunktion und $\Delta H(z)$ die durch den Quantisierungsfehler verursachte Systemfunktion. Diese steht aufgrund

$$\Delta H(z) = \sum_{n=0}^M \Delta h[n] z^{-n}$$

in einem linearen Zusammenhang mit den Quantisierungsfehlern der Koeffizienten $\Delta h[n]$.

3.6.0.1.2 Filtertopologien

Direkte Form

Die direkte Form eines FIR-Filters entspricht der Realisierung von Gleichung (4).

Reihenschaltungsform

Die Reihenschaltungsform erhält man durch eine Faktorisierung der polynomialen Systemfunktion:

$$H(z) = \prod_{k=1}^{M_S} (a_{0k} + a_{1k} z^{-1} + a_{2k} z^{-2}) \quad \text{wobei: } M_S = (M + 1) \text{ modulo}(2)$$

Linearphasige FIR-Systeme

Bei linearphasigen Systemen läßt sich die Differenzgleichung aufgrund der Achsen- bzw. Punktsymmetrie der Impulsantwort vereinfachen. Die Anzahl der Multiplikationen reduziert sich dadurch bei einer Länge der Impulsantwort von $M + 1$ Abtastwerten auf

$$(M - 1) / 2 \quad \text{wenn: } M \dots \text{ungerade}$$

und auf

$$M / 2 - 1 \quad \text{wenn: } M \dots \text{gerade}$$

Dabei sind jedoch ebensoviele zusätzliche Additionen notwendig.

3.6.0.2 IIR-Filter

Die Differenzgleichung für IIR-Filter (IIR = infinite impulse response, unendliche Impulsantwort) lautet

$$y[n] = \sum_{k=1}^N b_k y[n-k] + \sum_{k=0}^M a_k x[n-k]$$

und besitzt die dazugehörige Systemfunktion:

$$H(z) = \frac{\sum_{k=0}^M a_k z^{-k}}{1 - \sum_{k=1}^N b_k z^{-k}}$$

3.6.0.2.1 Rundungsfehler

Durch den Rekursivteil der Differenzgleichung muß das Ausgangssignal zurückgeführt werden. Da aber die internen Rechenregister eine konstante Länge aufweisen, muß das Signal vor der Multiplikation in seiner Auflösung verringert werden. Dieser Quantisierungsvorgang erzeugt zusätzliches Rauschen. Im allgemeinen korreliert dieses Rauschen mit dem Nutzsignal und wird durch den Rückkopplungsweig des Filters spektral verändert. Verschiedene Verfahren - je nach verwendeter Topologie - verringern das so entstandene Rauschen (siehe [16]).

3.6.0.2.2 Skalierung

Um eine Übersteuerung eines internen Zweiges des Systems zu verhindern und andererseits die Dynamik des Eingangssignals nicht unnötig zu verringern, versucht man das Eingangssignal möglichst optimal an die interne Struktur des Filters mit seiner erweiterten Bitauflösung anzupassen. Dabei wird entweder das Eingangssignal mit einem entsprechenden Wert skaliert, oder man stellt den internen Knotenpunkten des Systems einen entsprechenden Headroom an Bits zur Verfügung. Verschiedene Skalierungsstrategien, welche auf der Annahme von diversen "worsed-case"-Eingangssignalen beruhen, führen zu unterschiedlichen Skalierungsfaktoren (siehe [23]).

3.6.0.2.3 Filtertopologien

Es folgt ein Überblick über verschiedene IIR-Filtertopologien. Dabei werden nur Systeme 2. Ordnung erörtert. Neben der entsprechenden Differenzgleichung bzw. Systemfunktion wird auf diverse Vor- und Nachteile der jeweiligen Topologie hingewiesen. Umfangreiche Untersuchungen finden sich in [17] und [20].

Direkte Form I

Differenzgleichung: (Realisierung siehe Bild 44)

$$y[n] = a_0x[n] + a_1x[n-1] + a_2x[n-2] - b_1y[n-1] - b_2y[n-2]$$

Systemfunktion:

$$H(z) = \frac{a_0 + a_1z^{-1} + a_2z^{-2}}{1 + b_1z^{-1} + b_2z^{-2}}$$

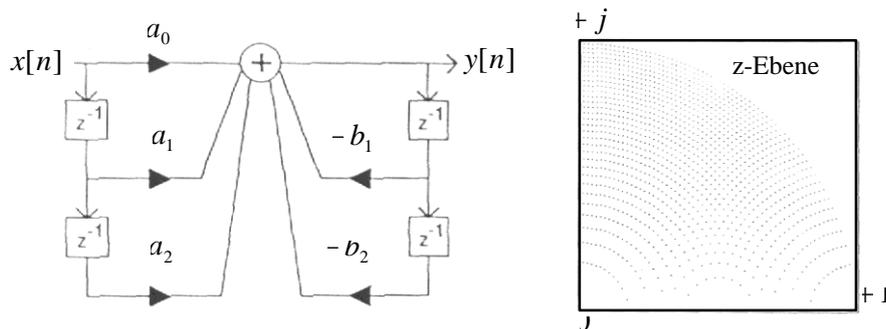


Bild 44: Realisierung der direkten Form I und Polstellenverteilung bei direkter Form I und II sowie transponierter direkter Form I und II. Bild nach [16] und [18].

Rechenaufwand:

- 5 Multiplikationen
- 4 Additionen

Vorteile:

- Es ist keine Skalierung des Eingangssignals notwendig (bei Ausnutzung der Modulo2-Arithmetik).
- Das Rauschen durch Rundungsfehler lässt sich mittels "Error Spectrum Shaping"-Techniken weitgehendst eliminieren (siehe [16]).

Nachteil:

- Durch die Koeffizientenquantisierung ergibt sich im Bereich $z=1$ eine geringe Dichte der möglichen Polstellen (siehe Bild 44).

Transponierte direkte Form I

Differenzgleichung: (Realisierung siehe Bild 45)

$$y[n] = a_0 w[n] + a_1 w[n-1] + a_2 w[n-2]$$

wobei: $w[n] = x[n] - b_1 w[n-1] - b_2 w[n-2]$

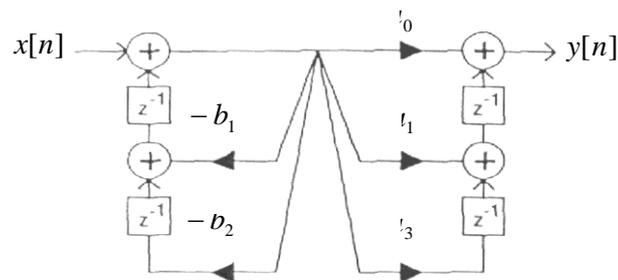


Bild 45: Realisierung der transponierten direkten Form I. Bild nach [16].

Rechenaufwand:

- wie bei direkter Form I.

Nachteil:

- Eingangssignal muß aufgrund der Zwischengröße $w[n]$ skaliert werden.
- Polstellenverteilung wie bei direkter Form I (siehe Bild 44).

Direkte Form II

Differenzgleichung: (Realisierung siehe Bild 46)

$$y[n] = a_0 w[n] + a_1 w[n-1] + a_2 w[n-2]$$

wobei: $w[n] = x[n] - b_1 w[n-1] - b_2 w[n-2]$

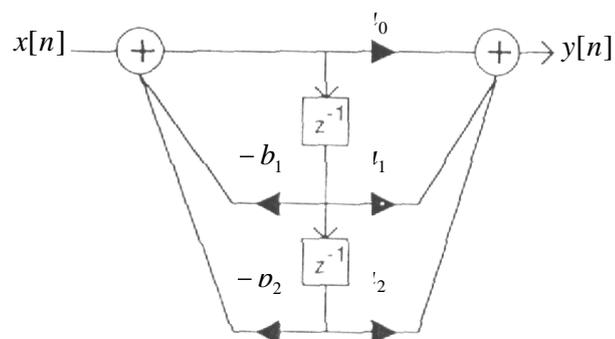


Bild 46: Realisierung der direkten Form II. Bild nach [16].

Rechenaufwand:

- wie bei direkter Form I.

Vorteil:

- benötigt am wenigsten Speicherplätze.

Nachteile:

- Eingangssignal muß skaliert werden.
- Polstellenverteilung wie bei direkter Form I (siehe Bild 44).

Transponierte direkte Form II

Differenzgleichung: (Realisierung siehe Bild 47)

$$y[n] = a_0x[n] + a_1x[n-1] + a_2x[n-2] - b_1y[n-1] - b_2y[n-2]$$

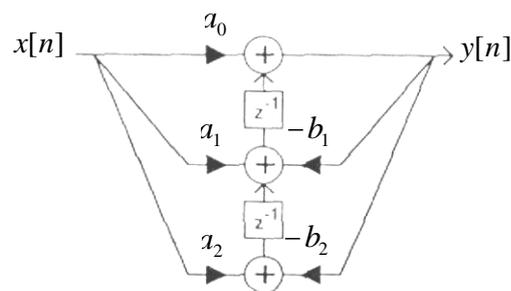


Bild 47: Realisierung der transponierten direkten Form II. Bild nach [16].

Rechenaufwand:

- wie bei direkter Form I.

Vorteile:

- benötigt am wenigsten Speicherplätze.
- Eingangssignal muß nicht skaliert werden.

Nachteil:

- Polstellenverteilung wie bei direkter Form I (siehe Bild 44).

Gold-Rader-Struktur

Differenzgleichung des Rekursivteils: (Realisierung siehe Bild 48)

$$y[n] = r \cos(\varphi) y[n-1] + r \sin(\varphi) w[n-1]$$

wobei: $w[n] = x[n-1] + r \cos(\varphi) w[n-1] - r \sin(\varphi) y[n-1]$

Systemfunktion:

$$H(z) = \frac{N(z)}{1 + 2r \cos(\varphi) z^{-1} + r^2 z^{-2}}$$

Dabei entspricht r dem Radius und φ dem Winkel der Polstelle in der z -Ebene.

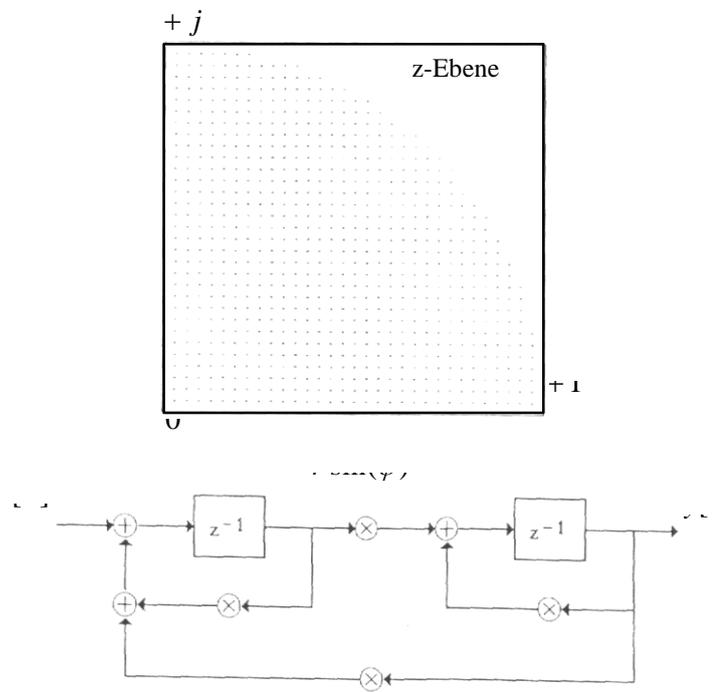


Bild 48: Polstellenverteilung und Realisierung des Rekursivteils der Gold-Rader-Struktur. Bild nach [18].

Rechenaufwand:

- 7 Multiplikationen
- 6 Additionen

Vorteil:

- gleichmäßige Polstellenverteilung (siehe Bild 48).

Nachteil:

- Eingangssignal muß skaliert werden.

Kingsbury-Struktur

Differenzgleichung des Rekursivteils: (Realisierung siehe Bild 49)

$$y[n] = y[n-1] - k_1(w[n-1] + k_2 y[n-1])$$

$$\text{wobei: } w[n] = w[n-1] + x[n] + k_1 y[n]$$

Systemfunktion:

$$H(z) = \frac{N(z)}{1 - (2 - k_1 k_2 - k_1^2)z^{-1} - (1 - k_1 k_2)z^{-2}}$$

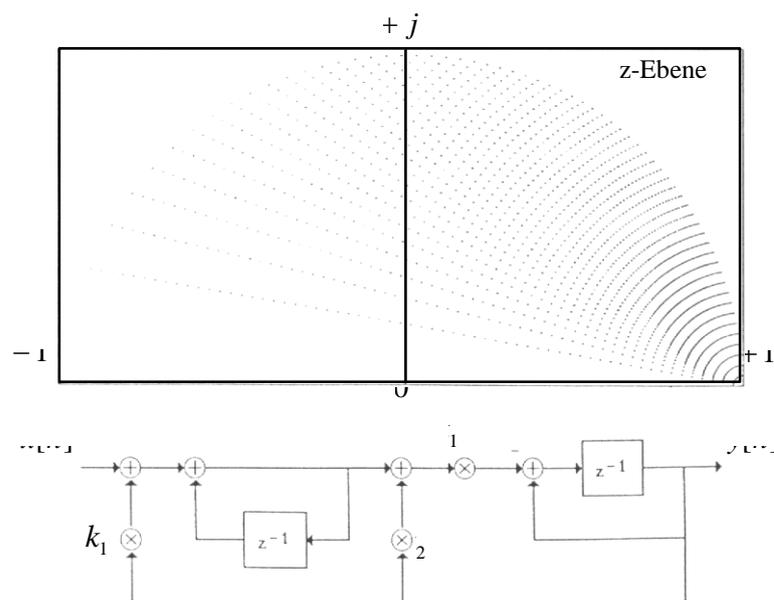


Bild 49: Polstellenverteilung und Realisierung des Rekursivteils der Kingsbury-Struktur. Bild nach [18].

Rechenaufwand:

- 6 Multiplikationen
- 7 Additionen

Vorteil:

- hohe Dichteverteilung der Polstellen bei $z = 1$ (siehe Bild 49).

Nachteil:

- Eingangssignal muß skaliert werden.

Gitterstrukturen

Gitterstrukturen dienen vorwiegend zur Realisierung von Allpol- und Allpaßfiltern. Es gibt eine Vielzahl von weiteren Topologien für Gitterstrukturen mit verschiedenen Eigenschaften.

Systemfunktion eines Allpasses: (Realisierung siehe Bild 51)

$$H(z) = \frac{k_2 + k_1(1 + k_2)z^{-1} + z^{-2}}{1 + k_1(1 + k_2)z^{-1} + k_2z^{-2}}$$

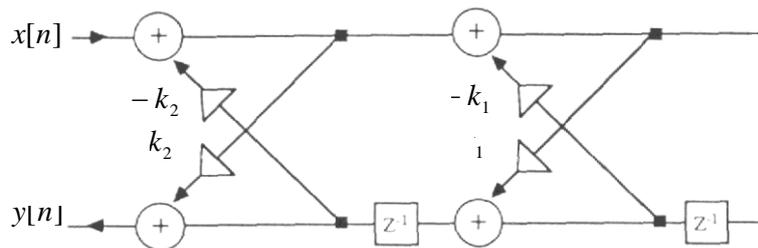


Bild 51: Realisierung einer Gitterstruktur. Bild nach [23].

3.6.1 Elimination von Transienten

3.6.1.1 Elimination von Transienten bei FIR-Filtern

Da FIR-Filter eine endliche Impulsantwort besitzen, kann die Entstehung von Transienten vollständig eliminiert werden.

Dazu werden bei einer Länge der Filterimpulsantwort von N Abtastwerten die letzten N Eingangswerte gespeichert. Wenn sich nun die Filterkoeffizienten verändern, so werden diese Eingangswerte erneut mit der neuen Impulsantwort gefaltet und so der aktuelle Ausgangswert berechnet.

Wenn sich zusätzlich zu den Koeffizienten auch die Ordnung des Filters ändert, so müssen so viele Eingangswerte zwischengespeichert werden, wie es der höchsten gewünschten Filterordnung entspricht.

3.6.1.2 Elimination von Transienten bei IIR-Filtern

Durch die rekursive Struktur des IIR-Filters liefern alle vorhergehenden Eingangswerte einen Beitrag zum aktuellen Ausgangswert. Wenn sich die Filterkoeffizienten ändern, so müßten theoretisch alle Eingangswerte zur Berechnung des neuen Ausgangswertes zur Verfügung stehen. Dadurch lassen sich die Transienten bei IIR-Filtern lediglich reduzieren und nicht vollständig beseitigen.

Überlappende Synthese

Wenn sich die Koeffizienten mit einer niedrigeren Taktrate als das Audiosignal verändern, so kann man das Prinzip der überlappenden Synthese, wie es bereits in Kap. 3.1.2.2 erläutert wurde, verwenden. Wenn sich die Filterkoeffizienten ändern, so wird einerseits das Eingangssignal mit den alten Koeffizienten weiterberechnet und ausgeblendet, gleichzeitig wird aber auch das aktuelle Ausgangssignal mit den neuen Koeffizienten berechnet und eingeblendet. Um die Transienten weiter zu unterdrücken, kann dieses Prinzip auch zeitversetzt erfolgen, damit die Transienten nicht schon im Einblendungsvorgang hörbar werden. Jedoch muß das Signal hierfür über einen längeren Zeitraum berechnet werden.

Interpolation der Koeffizienten

Hierbei werden die bisherigen Filterkoeffizienten zu den aktuellen Koeffizienten mittels Interpolationsverfahren übergeführt. Um die Transienten ausreichend zu unterdrücken, sind jedoch kleine Interpolationsschritte erforderlich, was einen hohen Rechenaufwand erfordert. Außerdem erhält das Ausgangssignal durch Änderung der frequenzbestimmenden Parameter glissando-ähnlichen Charakter.

Berechnung der aktuellen Zustandsvariablen

Eine akustisch zufriedenstellende Methode erhält man, wenn man neben den neuen Filterkoeffizienten auch die Zustandsvariablen des Filters mit den neuen Koeffizienten und dem bisherigen Eingangssignal berechnet und aktualisiert (siehe [19])

Die Impulsantwort wird hierzu nach N Abtastwerten abgeschnitten, wobei N so gewählt wird, daß die Signalenergie der resultierenden Impulsantwort ausreichend groß ist, um die noch vorhandenen Transienten vernachlässigen zu können. Zur Berechnung der aktuellen Zustandsvariablen müssen ebensoviele Eingangswerte zwischengespeichert, bzw. das Ergebnis um diesen Abtastwert zeitversetzt aktualisiert werden. Bild 52 illustriert das Prinzip dieser Technik.

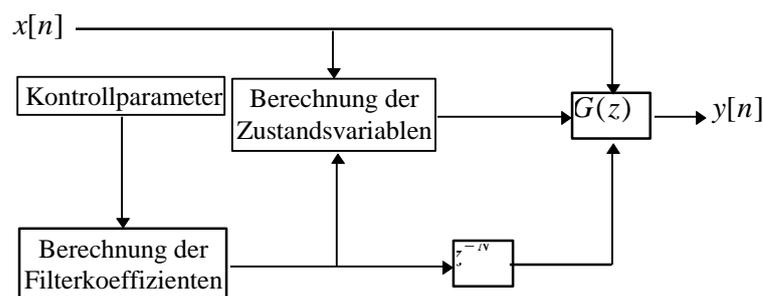


Bild 52: Prinzip der Elimination von Transienten durch Berechnung der Zustandsvariablen.

Nachdem die neuen Filterkoeffizienten berechnet wurden, werden mit den N folgenden Eingangswerten die neuen Zustandsvariablen berechnet. Schließlich werden die Zustandsvariablen und die aktuellen Koeffizienten dem Filteralgorithmus, um N Abtastwerte verzögert, übergeben.

Durch diese Methode lassen sich die Filterkoeffizienten jedoch nur alle N Abtastwerte durch neue Werte ersetzen.

Eine effizientere Methode - jedoch auf Kosten zusätzlichen Rechenaufwandes - erreicht man durch Zwischenspeichern der letzten N Eingangswerte und Berechnung der Zustandsvariablen innerhalb einer Abtastperiode. Der Rechenaufwand wird dadurch N -mal so groß, die Koeffizienten können dafür mit der Signalarate aktualisiert werden.

3.6.2 Tiefpaßfilter

3.6.2.1 Analyse eines analogen Tiefpaßfilters

Bild 53 zeigt die Struktur eines analogen VCF, wie er von Robert Moog vorgeschlagen wurde. Durch die Rückkopplung kommt es im Bereich der Grenzfrequenz zu einem Resonanzverhalten, dessen Güte über den Parameter k variiert werden kann. Die Grenzfrequenz und die Güte des Filters lassen sich unabhängig voneinander steuern. Um Stabilität zu gewährleisten, muß $0 \leq k < 4$ gelten.

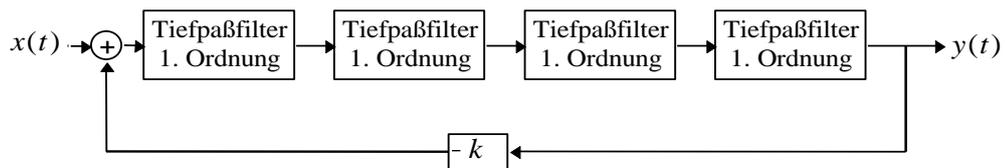


Bild 53: Struktur eines analogen Tiefpaßfilters mit Resonanzverhalten und einer Flankensteilheit von 24dB/Oktave.

Die Gesamtsystemfunktion ergibt sich zu:

$$H(s) = \frac{H_{TP}^4(s)}{1 + kH_{TP}^4(s)}$$

wobei: $H_{TP}(s) = \frac{\Omega_C}{\Omega_C + s}$

Die Abhängigkeit der Polstellen von den Parametern k und Ω_C läßt sich anhand des Wurzelortskurvenverfahrens analysieren (siehe [20]).

Die Polstellen ergeben sich zu:

$$s_p = \Omega_C(-1 \pm \sqrt[4]{k}e^{\pm j\pi/4})$$

Bild 54 zeigt die Wurzelortskurven bei Variation des jeweiligen Parameters in der s -Ebene.

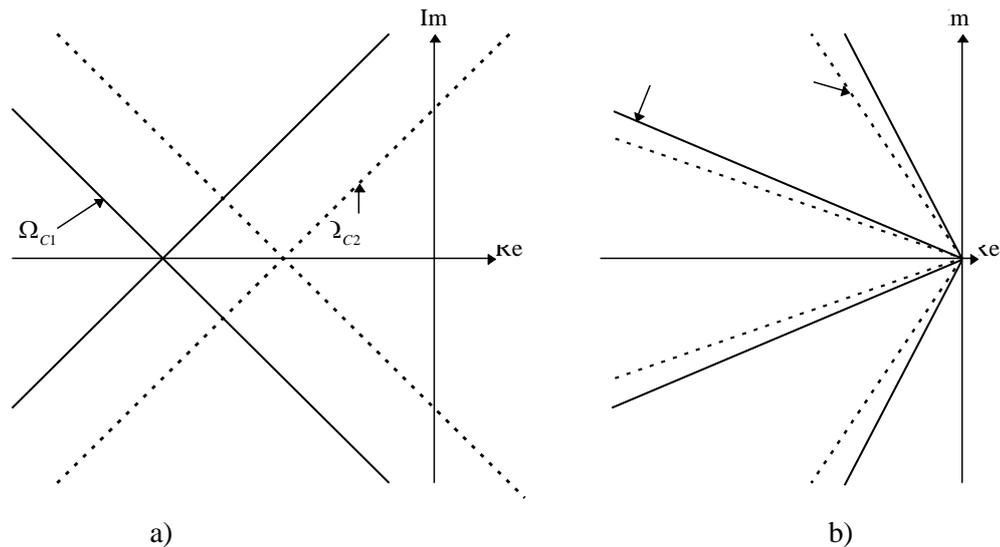


Bild 54: Wurzelortskurve des Analogfilters: a) kontinuierliches Verändern des Parameters k bei zwei verschiedenen Grenzfrequenzen; b) kontinuierliches Verändern der Grenzfrequenz Ω_C bei zwei verschiedenen Werten von k .

3.6.2.2 Implementierungen des VCF

3.6.2.2.1 Entwurf mittels Analog-Digital-Transformationen

Um die Unabhängigkeit der beiden Parameter Ω_C und k auch in der digitalen Ebene zu gewährleisten, wird die Struktur der Filterkaskade und der Rückkopplungszweig beibehalten. Es wird also lediglich das Tiefpaßfilter laut Transformationsvorschrift in die digitale Ebene übergeführt und in die vorhandene Struktur eingesetzt.

Bilineare Transformation

Die analoge Tiefpaßsystemfunktion

$$H_{TP}(s) = \frac{a}{a + s}$$

ergibt mit der bilinearen Transformation

$$s \leftarrow \frac{2(z-1)}{T_s(z+1)}$$

die Tiefpaßfunktion im z -Bereich:

$$H_{TP}(z) = \frac{(p+1)(z+1)}{2(z+p)} \quad \text{wobei:} \quad p = \frac{a-2}{a+2}$$

Der Zusammenhang zwischen digitaler Grenzfrequenz Θ_C und Parameter p ist in Bild 55 dargestellt.

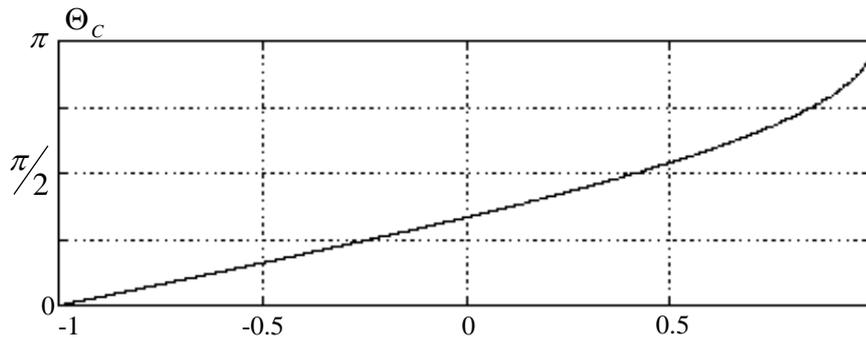


Bild 55: Zusammenhang zwischen Θ_C und p bei der bilinearen Transformation. Bild nach [20].

Die Gesamtsystemfunktion ergibt sich zu:

$$H(z) = \frac{H_{TP}^4(z)}{1 + kH_{TP}^4(z)}$$

Diese Systemfunktion enthält einen geschlossenen Pfad ohne Verzögerungselement, wodurch eine Implementierung unmöglich wird. Um eine Realisierung zu ermöglichen, wird in den Rückkopplungszweig ein Verzögerungselement eingefügt, wodurch sich folgende Systemfunktion ergibt:

$$H(z) = \frac{H_{TP}^4(z)}{1 + kz^{-1}H_{TP}^4(z)}$$

Dadurch geht jedoch die Unabhängigkeit zwischen dem Frequenzparameter p und der Güte k verloren. Bild 56 zeigt die gegenseitige Abhängigkeit der beiden Parameter bei verschiedenen Güten. Die oberste Funktion entspricht dabei dem Grenzfall zur Instabilität. Man sieht, daß $k < 1$ sein muß, um auch bei hohen Frequenzen Stabilität zu gewährleisten. Weiters ist zu erkennen, daß die Kurvenschar annähernd skalierte Versionen der obersten Funktion darstellen, wodurch diese zur Entkopplung der beiden Parameter verwendet werden kann. Dabei wird zuerst der Skalierungsfaktor aus einer Wertetabelle für die entsprechende Güte bestimmt und schließlich mit dem für den Frequenzparameter p erhaltenen Funktionswert multipliziert, um den resultierenden Wert für k zu erhalten.

Das Ergebnis dieses Verfahrens ist in Bild 57 dargestellt.

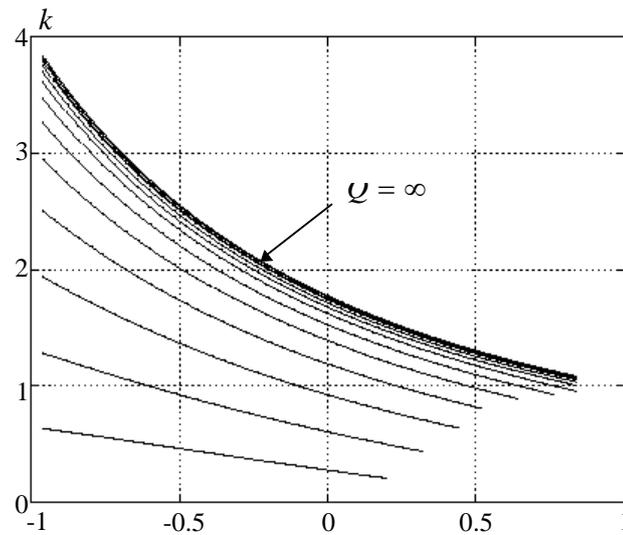


Bild 56: Gegenseitige Beeinflussung der beiden Parameter k und p bei verschiedenen Güten bei der bilinearen Transformation. Bild nach [20].

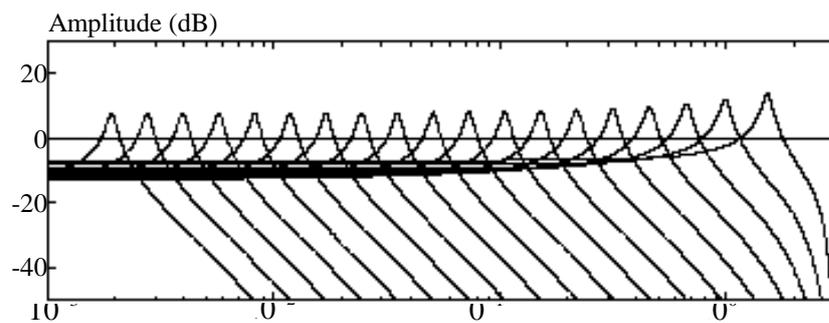


Bild 57: Frequenzantwort des Tiefpaßfilters bei konstanter Güte und verschiedenen Grenzfrequenzen bei der bilinearen Transformation. Bild nach [20].

Die Differenzgleichung eines Tiefpasses lautet:

$$y[n] = \frac{(x[n] + x[n-1])(p+1)}{2} - py[n-1]$$

Der Rechenaufwand des Gesamtsystems beträgt somit:

- 1 Wavetable-Lookup zur Umrechnung von Θ_c auf p .
- 2 Wavetable-Lookups zur Entkopplung von p und k (Funktionswerte und Skalierungsfaktoren).
- 9 Multiplikationen
- 10 Additionen
- 4 Shift-Operationen

Transformation mittels Rückwärtsdifferenzen

Über die Transformationsvorschrift mittels Rückwärtsdifferenzen

$$s \leftarrow \frac{z-1}{T_S z}$$

erhält man die Tiefpaßfunktion im z -Bereich zu:

$$H_{TP}(z) = \frac{(p-1)z}{zp-1} \quad \text{wobei: } p = a+1$$

Bild 58 illustriert wiederum den Zusammenhang zwischen Θ_c und p .

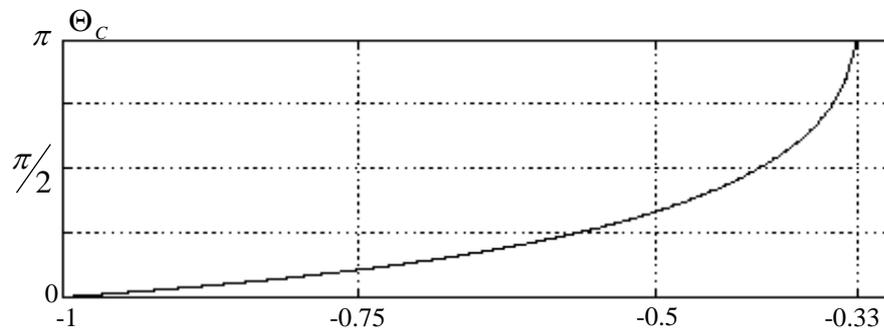


Bild 58: Zusammenhang zwischen Θ_c und p bei der Transformation mittels Rückwärtsdifferenzen. Bild nach [20].

Die Gesamtsystemfunktion benötigt wiederum ein zusätzliches Verzögerungselement im Rückkopplungszweig und lautet:

$$H(z) = \frac{H_{TP}^4(z)}{1 + kz^{-1}H_{TP}^4(z)}$$

Bild 59 zeigt den Vorteil dieser Methode: Während zwar der Zusammenhang zwischen k und p nach wie vor nichtlinear ist, bleibt aber das System unabhängig vom Wert von p stabil. Eine Korrekturtabelle ist daher nicht zwingend notwendig, wodurch der Rechenaufwand vermindert wird. Bild 60 zeigt verschiedene Frequenzantworten bei konstantem Parameter k mit dieser Methode ohne Korrekturtabelle.

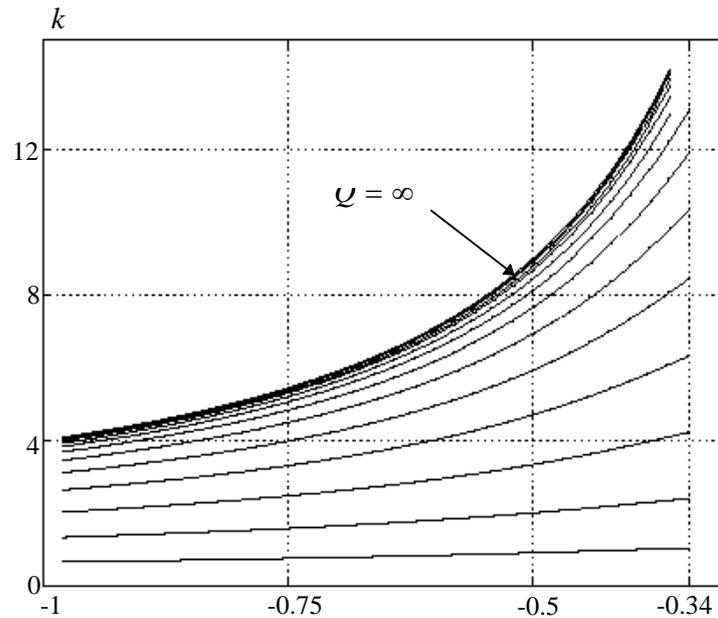


Bild 59: Gegenseitige Beeinflussung der beiden Parameter k und p bei verschiedenen Güten und Transformation mittels Rückwärtsdifferenzen. Bild nach [20].

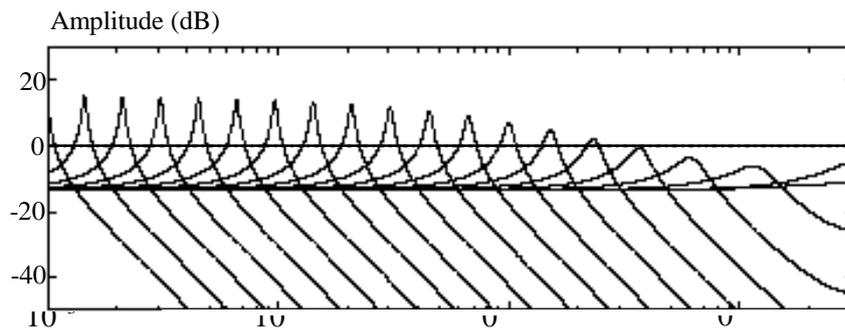


Bild 60: Frequenzantwort des Tiefpaßfilters bei konstant gehaltener Güte und verschiedenen Grenzfrequenzen bei Transformation mittels Rückwärtsdifferenzen. Bild nach [20].

Die Differenzgleichung eines Tiefpasses lautet:

$$y[n] = \frac{x[n](p-1) + y[n-1]}{p}$$

Der Rechenaufwand des Gesamtsystems erfordert folgende Operationen:

- 1 Wavetable-Lookup zur Umrechnung von Θ_c auf p .
- 9 Multiplikationen
- 6 Additionen
- 1 Division

Approximation des Tiefpaßfilters

Ein Vergleich von Bild 56 und Bild 59 zeigt ein gegensätzliches Verhalten der Abhängigkeit der beiden Parameter p und k . Dies ist auf die Nullstelle des Tiefpaßfilters zurückzuführen. Während sich diese bei der bilinearen Transformation am Punkt $z = -1$ befindet, liegt sie bei der Transformation mittels Rückwärtsdifferenzen am Punkt $z = 0$. Eine Kompromißlösung der beiden Methoden läßt sich durch Verlagerung der Nullstelle zwischen diese beiden Werte erreichen. Mit der neuen Nullstelle c wird die Systemfunktion des Tiefpaßfilters zu:

$$H_{TP}(z) = \frac{(p-1)(z-c)}{(1-c)(zp-1)}$$

Die Systemfunktion des gesamten Regelkreises ist wieder:

$$H(z) = \frac{H_{TP}^4(z)}{1 + kz^{-1}H_{TP}^4(z)}$$

Bild 61 zeigt den Zusammenhang zwischen Θ_c und p .

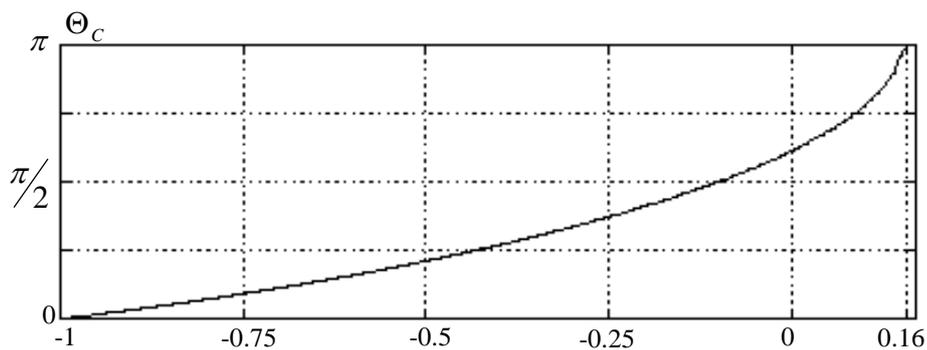


Bild 61: Zusammenhang zwischen Θ_c und p bei der Approximationsmethode. Bild nach [20].

Bild 62 illustriert die deutliche Verbesserung der Unabhängigkeit der beiden Parameter k und p bei verschiedenen Güten. Bild 63 zeigt verschiedene Frequenzantworten dieser Kompromißlösung bei konstanter Güte. Die Nullstelle des Tiefpasses wurde dabei auf $c = 0.3$ positioniert. Auch hier wurde auf eine Entkopplung der beiden Parameter über eine Korrekturtabelle verzichtet.

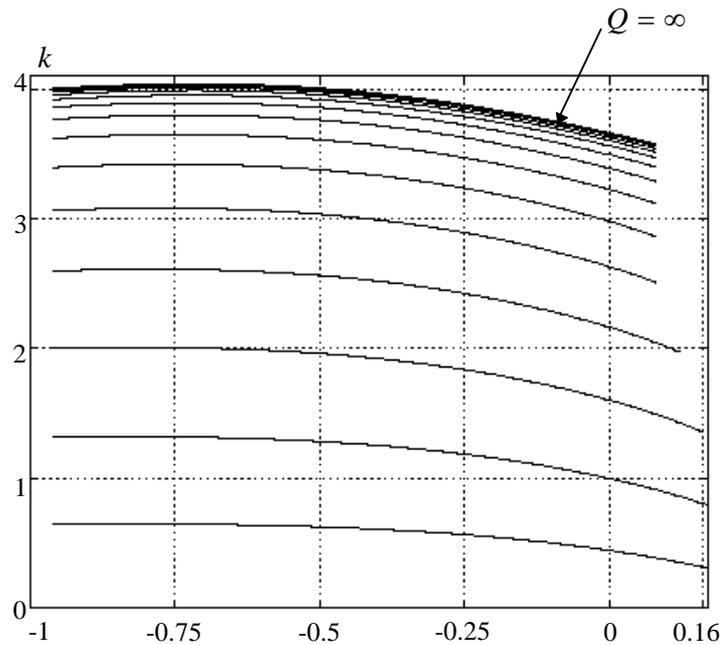


Bild 62: Gegenseitige Beeinflussung der beiden Parameter k und p bei verschiedenen Güten bei der Approximationsmethode. Bild nach [20].

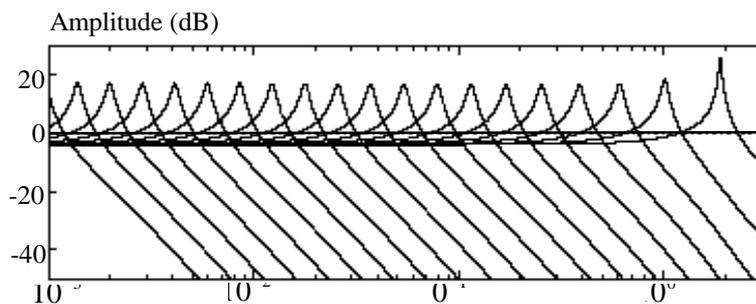


Bild 63: Frequenzantwort des Tiefpaßfilters bei konstant gehaltener Güte und verschiedenen Grenzfrequenzen bei der Approximationsmethode. Bild nach [20].

Die Differenzgleichung des Tiefpasses lautet:

$$y[n] = \frac{(p-1)(x[n] - cx[n-1]) + (1-c)y[n-1]}{p(1-c)}$$

und der Rechenaufwand des Gesamtsystems ergibt sich zu:

- 1 Wavetable-Lookup zur Umrechnung von Θ_c auf p .
- 14 Multiplikationen
- 10 Additionen
- 1 Division

3.6.2.2 Entwurf von digitalen Filtern

Approximation einer konstanten Güte

Eine weitere Möglichkeit, das in Kap. 3.6.1.1 analysierte Filter zu implementieren, kann auch durch den direkten Entwurf eines digitalen Filter erfolgen [20].

Ausgehend vom Entwurf eines Filters mit konstanter Bandbreite läßt sich durch gezieltes Hinzufügen von weiteren Pol- und Nullstellen ein Filter mit konstanter Güte approximieren.

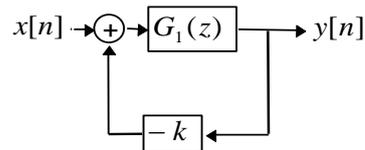


Bild 64: Struktur eines digitalen Tiefpaßfilters mit Resonanzverhalten.

Dabei ergibt die nach Bild 64 skizzierte Struktur mit der Gesamtsystemfunktion

$$G(z) = \frac{G_1(z)}{1 + kG_1(z)}$$

und der Teilsystemfunktion

$$G_1(z) = \frac{z}{(z-r)^2}$$

das erwähnte Filter mit konstanter Bandbreite. Der Zusammenhang zwischen den Kontrollparametern Grenzfrequenz Θ_c und Polradius r mit dem Rückkopplungsfaktor k ist gegeben durch die Beziehung:

$$k = 2r(1 - \cos(\Theta_c))$$

Bild 65 zeigt die Wurzelortskurve dieser Anordnung in der z -Ebene, wobei die strichlierten Linien die bilinear transformierten Wurzelortskurven des Analogfilters darstellen.

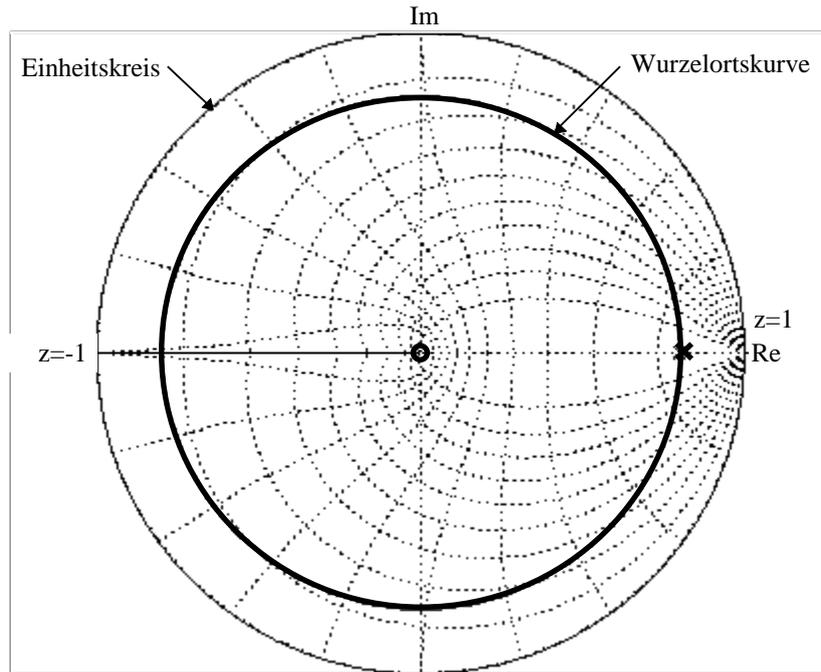


Bild 65: Wurzelortskurve in der z -Ebene eines Filters mit konstanter Bandbreite bei $r = 0.8$ und Variation von Θ_c . Bild nach [20].

Um nun ein Filter mit konstanter Güte zu realisieren, wird nach folgender Überlegung vorgegangen:

Zuerst wird die Wurzelortskurve in der z -Ebene nach rechts verschoben, so daß die Pole den Punkt $z = 1$ schneiden. Da sich dadurch bei einem Gleichsignalanteil eine unendliche Güte ergibt, werden die Polradien geringfügig reduziert. Die resultierende Teilsystemfunktion lautet:

$$G_1(z) = \frac{(z - c)}{(z - (1 - \varepsilon))^2} \quad \text{mit } \varepsilon \ll 1$$

Durch weiteres Hinzufügen von Pol- und Nullstellen läßt sich das Ergebnis auf Kosten höheren Rechenaufwandes optimieren. Bild 66 zeigt schließlich die approximierte Wurzelortskurve mit den hinzugefügten Polstellen p bei $p_1 = 0.5$, $p_2 = 0.9$, $p_3 = 0.97$, $p_4 = 0.9975$ und $p_5 = p_6 = 1$, sowie den Nullstellen c bei $c_1 = 0.1$, $c_2 = 0.55$, $c_3 = 0.92$, $c_4 = 0.975$ und $c_5 = 0.9983$.

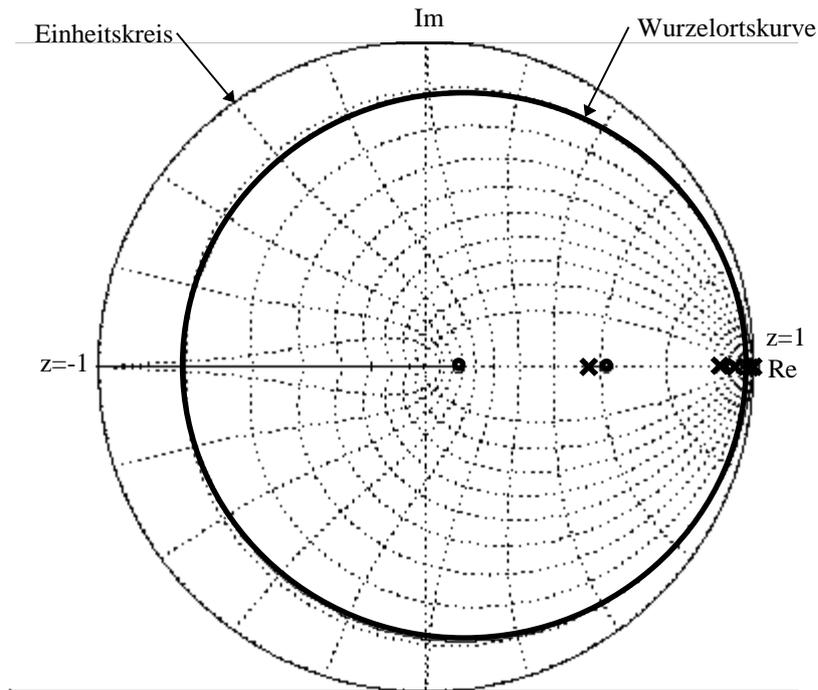


Bild 66: Wurzelortskurve in der z -Ebene eines Filters mit approximierter konstanter Güte bei Variation von Θ_c . Bild nach [20].

Bild 67 zeigt den Verlauf der Güte in Abhängigkeit von der Grenzfrequenz. Während das Ergebnis - zumindest bei hohen Grenzfrequenzen - zufriedenstellend erscheint, liegt der Nachteil dieser Methode im hohen Rechenaufwand durch die hohe Ordnung des Filters; weiters müßten die Pole und Nullstellen für verschiedene Güten berechnet und im Table-Lookup-Prinzip in die Filterberechnung einbezogen werden.

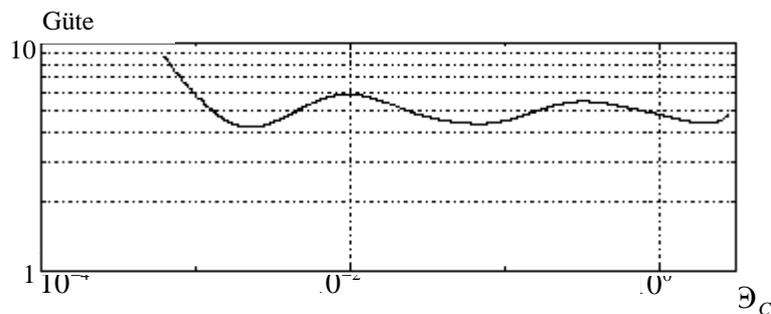


Bild 67: Abhängigkeit der Güte von der Grenzfrequenz Θ_c des Filters bei Approximation einer konstanten Güte durch Hinzufügen von 6 weiteren Pol- und 4 weiteren Nullstellen. Bild nach [20].

Tiefpaßfilter 1. Ordnung

Wird auf die Güte als Kontrollparameter verzichtet, so lassen sich Tiefpaßfilter 1. Ordnung in Kaskadenrealisierung verwenden. Einfache Strukturen ergeben sich durch Verwendung von Allpaßfiltern (siehe [21]).

Die Differenzgleichung für ein Filter 1. Ordnung lautet:

$$y[n] = \frac{1}{2}(x[n] + w[n])$$

$$\text{wobei: } w[n] = b(x[n] - w[n-1]) + x[n-1]$$

Der Zusammenhang zwischen b und Θ_c ist gegeben durch:

$$b = \frac{\tan(\Theta_c / 2) - 1}{\tan(\Theta_c / 2) + 1}$$

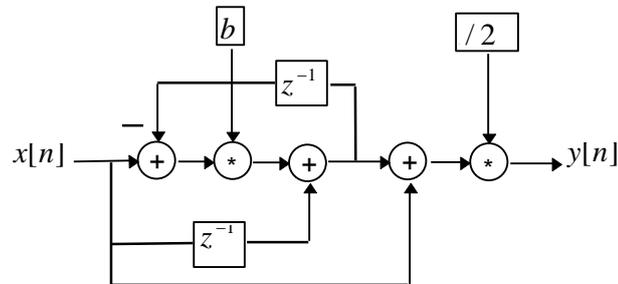


Bild 68: Struktur eines Tiefpaßfilters 1. Ordnung.

Die Struktur des Filters ist in Bild 68 ersichtlich und der Rechenaufwand beträgt:

- 1 trigonometrische Berechnung
- 1 Multiplikation
- 5 Additionen
- 1 Division
- 1 Shift-Operation

3.6.3 Hochpaßfilter

Hochpaßfilter 1. Ordnung

Auch das Hochpaßfilter läßt sich mittels Allpaßfiltern realisieren. Wie in Bild 69 dargestellt unterscheidet sich die Struktur vom Tiefpaß 1. Ordnung lediglich durch die Invertierung des Allpaßausganges. Der Rechenaufwand ist also gleich wie beim Tiefpaß 1. Ordnung (siehe oben).

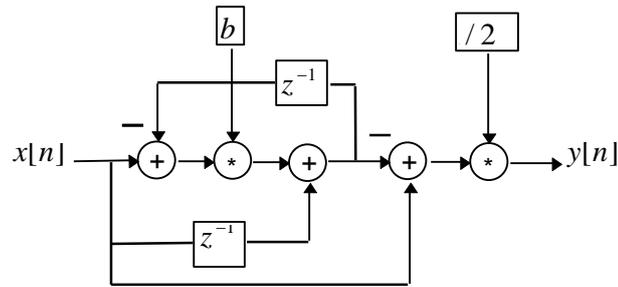


Bild 69: Struktur eines Hochpaßfilters 1. Ordnung.

3.6.4 Bandpaßfilter

Kaskadenrealisierung

Das trivialste Bandpaßfilter läßt sich aus einer Reihenschaltung von Tief- und Hochpässen realisieren. Je nach Anzahl der verwendeten Tief- und Hochpässe lassen sich verschiedene Flankensteilheiten erzeugen. Während der Rechenaufwand für die Filterstruktur linear mit der Anzahl der Filter steigt, bleibt der Rechenaufwand für die Koeffizienten konstant. (Rechenaufwand siehe daher Tief- und Hochpaß 1. Ordnung).

Zustandsvariablen-Filter

In [22] wird ein einfaches Zustandsvariablen-Filter vorgestellt, welches gleichzeitig als Tief-, Hoch- und Bandpaß agiert. Die Differenzgleichungen für die jeweilige Filtercharakteristiken $y_{TP}[n]$, $y_{HP}[n]$ und $y_{BP}[n]$ lauten:

$$\begin{aligned}
 y_{TP}[n] &= b_1[n]y_{BP}[n] + y_{TP}[n-1] \\
 y_{HP}[n] &= x[n] - y_{TP}[n-1] - b_2[n]y_{BP}[n-1] \\
 y_{BP}[n] &= b_1[n]y_{HP}[n] + y_{BP}[n-1]
 \end{aligned}$$

Bild 70 zeigt die Struktur des Filters.

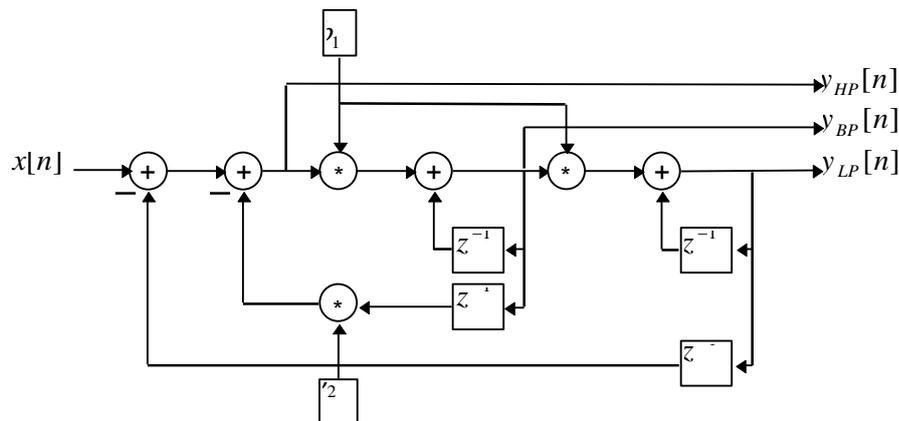


Bild 70: Struktur des Zustandsvariablen-Filters mit Tief-, Hoch- und Bandpaßcharakteristik.

Der Zusammenhang zwischen den Koeffizienten b_1 und b_2 , sowie den Kontrollparametern Grenz- bzw. Mittenfrequenz Θ_c und Dämpfungsfaktor ζ lautet:

$$b_1 = 2 \sin(\Theta_c / 2)$$

$$b_2 = 2\zeta$$

Die Beziehung zwischen der Dämpfung ζ und der Güte Q ist gegeben mit:

$$Q = \frac{1}{2\zeta\sqrt{1-\zeta^2}} \quad \text{bzw.:} \quad Q \approx \frac{1}{2\zeta} \quad \text{für } \zeta \ll 1$$

Mit der Näherungsformel wird schließlich

$$b_2 = \frac{1}{Q}$$

und der Rechenaufwand beträgt insgesamt:

- 1 trigonometrische Berechnung
- 3 Multiplikationen
- 4 Additionen
- 2 Shift-Operationen
- 1 Division

Der Nachteil dieses Filters liegt in der begrenzten Stabilität. Damit die Pole innerhalb des Einheitskreises bleiben muß für die Güte gelten:

$$Q > \frac{\tan(\Theta_c / 2)}{\cos(\Theta_c / 2)}$$

Die Pole sind konjugiert komplex, wenn gilt:

$$Q > \frac{1}{2(1 - \sin(\Theta_c / 2))}$$

Bild 71 illustriert diese Zusammenhänge und zeigt, daß das Filter lediglich für kleine Güten und hohe Grenz- bzw. Mittenfrequenzen instabil ist. Dieser Bereich wird in Audioanwendungen selten benötigt, wodurch dieses Filter - verglichen mit dem geringen Rechenaufwand und den drei möglichen Filtercharakteristiken - einen guten Kompromiß darstellt.

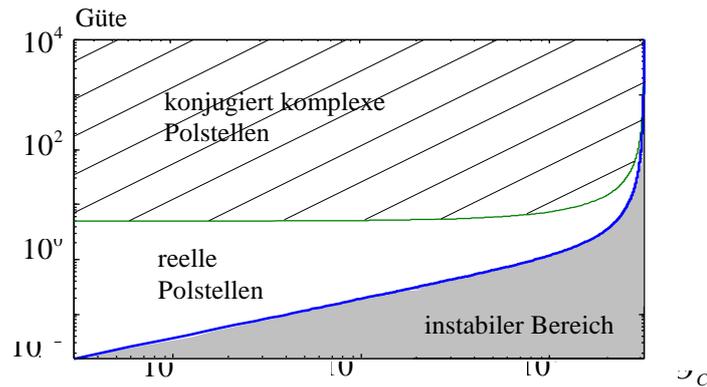


Bild 71: Instabiler Bereich und Bereich der reellen und komplexen Polstellen des Zustandsvariablen-Filters in Abhängigkeit von der Güte Q und der Grenz- bzw. Mittenfrequenz Θ_c .

3.6.5 Bandsperre

Parallelrealisierung

In Analogie zur Kaskadenrealisierung eines Bandpasses aus Tief- und Hochpaßfiltern läßt sich die Bandsperre durch eine Parallelschaltung von Tief- und Hochpässen konstruieren.

Realisierung mittels Gitterstrukturen

Der in [23] vorgestellte parametrische Equalizer läßt sich aufgrund der Unabhängigkeit der Kontrollparameter Mittenfrequenz Θ_c , Bandbreite B und Dämpfungsfaktor d auch als Bandsperre mit variabler Dämpfung einsetzen.

Die Struktur des Systems zeigt Bild 72, wobei das Allpaßsystem $A(z)$, welches ebenfalls in Bild 72 dargestellt ist, folgende Systemfunktion besitzt:

$$A(z) = \frac{z^{-2} + b_1(1 - b_2)z^{-1} + b_2}{1 + b_1(1 - b_2)z^{-1} + b_2z^{-2}}$$

Die Koeffizienten b_1 und b_2 sind wie folgt mit dem jeweiligen Kontrollparameter verknüpft:

$$b_1 = -\cos(\Theta_c)$$

$$b_2 = \frac{1 - \tan(B/2)}{1 + \tan(B/2)}$$

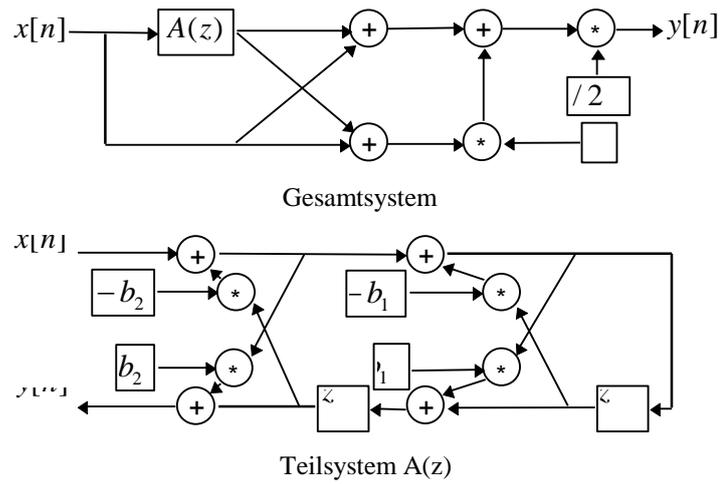


Bild 72: System einer Bandsperre mit einstellbarer Dämpfung d .

3.7 Flanger und Phaser

Flanging und Phasing sind Effekte, die mittels Kammfiltereffekt die Klangfarbe des Signals verändern. Während diese Frequenzminima beim Flanger bei äquidistanten Frequenzen auftreten, können sie beim Phaser beliebige Frequenzintervalle aufweisen. Im folgenden wird der Flanging- und Phasing-Effekt durch digitale Algorithmen realisiert. Abschnitt 3.7.1.1 erklärt die Funktionsweise des Flangers. Die zur Realisierung notwendigen zeitvariablen Verzögerungselemente werden in Abschnitt 3.7.1.2 behandelt. In Abschnitt 3.7.2.1 wird der Phaser beschrieben und in Kapitel 3.7.2.2 die dafür notwendige Realisierung des Allpaßfilters.

3.7.1 Flanger

3.7.1.1 Allgemeines

Bild 73 zeigt die Struktur eines Flangers. Die Übertragungsfunktion ist in Bild 14 ersichtlich. Das Eingangssignal wird um D Abtastwerte verzögert, wobei D auch nicht-ganzzahlige Werte annehmen kann. Durch die Addition des verzögerten Signales mit dem Originalsignal kommt es bei den Frequenzen

$$\Theta_n = \frac{(2n-1)\pi}{D} f_s \quad n = 1, 2, 3, \dots$$

zu Minima in der Übertragungsfunktion. Der Faktor m , wobei $0 < m \leq 1$, bestimmt die Stärke der Minima und wird als Modulationstiefe bezeichnet. Die Verzögerungszeit beträgt zwischen 0 und 10 Millisekunden und kann über ein Kontrollsignal in diesem Bereich zeitlich variiert werden (siehe [24]).

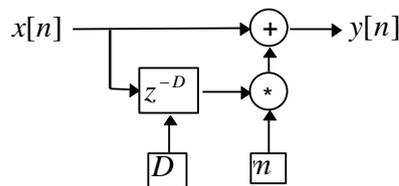


Bild 73: Strukturschaltbild eines Flangers.

Die Differenzgleichung des Flangers lautet:

$$y[n] = x[n] + mx[n - D[n]]$$

3.7.1.2 Zeitvariables Verzögerungselement

Das Verzögerungselement in Bild 73 wird durch Digitalfilter realisiert. Dabei ergibt sich wiederum die Möglichkeit der Verwendung von FIR- und IIR-Filtern, was im folgenden behandelt wird. Weitere Entwurfsmethoden werden in [25] vorgestellt.

3.7.1.2.1 FIR-Delay-Filter

Sinc-Interpolation

Eine Verzögerung des Signals um einen nichtganzzahligen Wert D ergibt mit der Systemfunktion

$$H(z) = z^{-D}$$

die Impulsantwort:

$$h[n] = \frac{\sin(\pi(n-D))}{\pi(n-D)}$$

Eine ideale Realisierung ist aufgrund der unendlich langen Impulsantwort nicht möglich. Durch Abschneiden dieser Funktion gelangt man zu:

$$h[n] = \begin{cases} \frac{\sin(\pi(n-D))}{\pi(n-D)} & 0 \leq n \leq N \\ 0 & \text{sonst} \end{cases} \quad \text{wobei: } N \dots \text{Filterordnung}$$

Damit der auftretende Fehler minimiert wird, soll für die Filterordnung

$$N \geq 2 \text{int}(D)$$

gelten.

Um das dadurch auftretende Gibbs-Phänomen in der Übertragungsfunktion zu unterdrücken, verwendet man Fensterfunktionen, die den unstetigen Übergang im Zeitbereich durch eine stetige Funktion ersetzen. Die resultierende Impulsantwort mit der Fensterfunktion $W[n]$ lautet:

$$h[n] = \begin{cases} W[n-D] \frac{\sin(\pi(n-D))}{\pi(n-D)} & 0 \leq n \leq N \\ 0 & \text{sonst} \end{cases}$$

Die Koeffizienten der Impulsantwort entsprechen beim FIR-Filter direkt den Filterkoeffizienten und können somit direkt in die Implementierung eingebunden werden. Bild 74 zeigt das entsprechende Verzögerungselement als FIR-Filter.

Lagrange-Interpolation

Hier wird die Übertragungsfunktion durch ein Polynom der Ordnung N angenähert, wodurch sich im Frequenzbereich von $\Theta = 0$ eine maximal flache Übertragungsfunktion ergibt und somit bei tiefen Frequenzen eine hohe Genauigkeit erzielt werden kann. Die Filterkoeffizienten berechnen sich zu:

$$h[n] = \prod_{\substack{k=0 \\ k \neq n}}^N \frac{D-k}{n-k} \quad \text{für } n = 0, 1, 2, \dots, N$$

Die Implementierung ist ebenfalls in Bild 74 zu sehen.

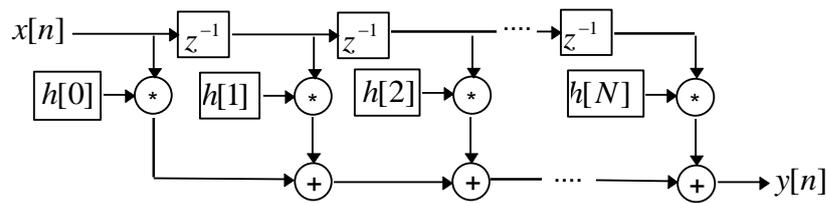


Bild 74: Implementierung des Verzögerungselementes als FIR-Filter.

3.7.1.2.2 Allpaßfilter

Der Entwurf von Verzögerungselementen mittels IIR-Filterdesign führt zu Allpaßfiltern, welche den Vorteil eines idealen Amplitudenganges aufweisen. Ihre Systemfunktion lautet mit der Ordnung N des Filters:

$$H(z) = \frac{a_n + a_{n-1}z^{-1} + \dots + a_1z^{-(N-1)} + z^{-N}}{1 + a_1z^{-1} + \dots + a_{N-1}z^{-(N-1)} + a_Nz^{-N}}$$

In [25] wird eine Methode vorgestellt, die sich durch eine maximal flache Gruppenlaufzeit bei der Frequenz $\Theta = 0$ auszeichnet und eine Berechnung der Filterkoeffizienten in geschlossener Form ermöglicht. Die Koeffizienten berechnen sich zu

$$a_k = (-1)^k \frac{N!}{k!(N-k)!} \prod_{n=0}^N \frac{D-N+n}{D-N+k+n} \quad \text{für } k = 0, 1, 2, \dots, N$$

wobei für die Filterordnung $N = \text{int}(D)$ gelten muß.

3.7.1.2.3 Steueralgorithmus

Da die Verzögerungszeit zeitlich variiert, muß ein Auftreten von Transienten beim Austauschen der Filterkoeffizienten vermieden werden. Eine Möglichkeit bildet die Verwendung einer Farrow-Struktur ([25]). Sie ist sowohl für FIR- als auch für IIR-Filter geeignet.

Dabei werden die Filterkoeffizienten durch ein Polynom P -ter Ordnung von D angenähert, bzw. ergibt dies für die Impulsantwort eines FIR-Filters:

$$h([n], D) = \sum_{m=0}^P c_m[n] D^m \quad \text{mit: } n = 0, 1, 2, \dots, N$$

Die Systemfunktion ergibt sich somit in Abhängigkeit von z und D zu:

$$H(z, D) = \sum_{n=0}^N h([n], D) z^{-n} = \sum_{n=0}^N \left(\sum_{m=0}^P c_m[n] D^m \right) z^{-n}$$

Durch Vertauschen der Indizes erhält man:

$$H(z, D) = \sum_{m=0}^P \left(\sum_{n=0}^N c_m[n] z^{-n} \right) D^m = \sum_{m=0}^P C_m(z) D^m \quad \text{wobei: } C_m(z) = \sum_{n=0}^N c_m(n) z^{-n}$$

Die resultierende Struktur läßt sich durch eine Parallelschaltung der Systeme $C_m(z)$, welche mit dem Faktor D^m gewichtet werden, realisieren. Bild 75 zeigt diese sogenannte Farrow-Struktur.

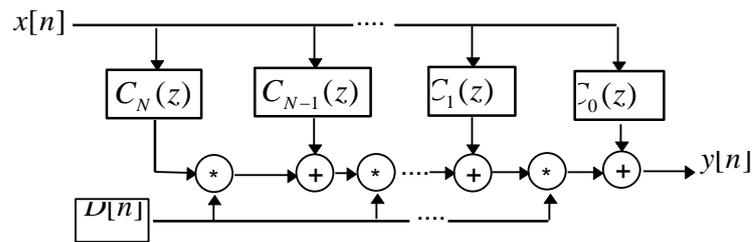


Bild 75: Farrow-Struktur zur Realisierung eines zeitabhängigen FIR-Filters.

In der Praxis wird der gewünschte Bereich $D_{\min} \leq D \leq D_{\max}$ in mehrere äquidistante Intervalle unterteilt. Schließlich werden mit einer wählbaren Approximationsmethode die entsprechenden Filterkoeffizienten $C_m(n)$ für diese Werte berechnet. Die einzelnen Koeffizienten werden dann über Polynome angenähert.

3.7.2 Phaser

3.7.2.1 Allgemeines

Bild 76 zeigt die Struktur eines Phasers mit einem Allpaß 8. Ordnung und Bild 16 dessen Übertragungsfunktion. Das Signal wird durch ein Allpaßfilter mit steuerbarer Mittenfrequenz Θ_c und Bandbreite B verändert und dem Originalsignal hinzuaddiert. Die Bandbreite ist ein Maß für den Abstand der Funktionsminima in der Übertragungsfunktion.

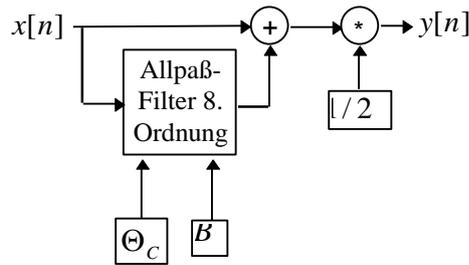


Bild 76: Strukturschaltbild eines mittels Allpaßfilter 8. Ordnung realisierten Phasers.

3.7.2.2 Allpaßfilter

Das im Kapitel 3.6.5 vorgestellte Allpaßfilter 2. Ordnung zur Realisierung eines Bandpaßfilters, welches sich unabhängig in Grenzfrequenz und Bandbreite steuern läßt, eignet sich auch für den Einsatz im Phaser (siehe [23]).

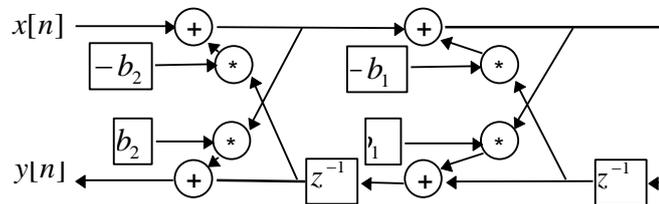


Bild 77: Allpaßfilter zur Realisierung des Phasers mit unabhängiger Steuerung zwischen Mittenfrequenz Θ_c und Bandbreite B .

Bild 77 zeigt die Struktur des Allpaßfilters. Die Koeffizienten b_1 und b_2 sind wie folgt mit dem jeweiligen Kontrollparameter verknüpft:

$$b_1 = -\cos(\Theta_c)$$

$$b_2 = \frac{1 - \tan(B/2)}{1 + \tan(B/2)}$$

Die in Kapitel 3.6.1.2 vorgeschlagenen Algorithmen zur Vermeidung von Transienten bei Änderung der Filterkoeffizienten gelten auch für das Allpaßfilter.

Der Rechenaufwand eines einstufigen Phasers beträgt:

- 2 trigonometrische Berechnungen
- 4 Multiplikationen
- 7 Additionen
- 1 Division
- 2 Shift-Operationen

4 Realisierungen

4.0 IRCAM Signal Processing Workstation und MAX

IRCAM Signal Processing Workstation

Die "IRCAM Signal Processing Workstation" (ISPW) bietet die Möglichkeit, Algorithmen zur Audiosignal-Verarbeitung zu implementieren.

Das System besteht aus einer Signalprozessor-Karte mit zwei Intel i860 RISC-Prozessoren und einem Motorola DSP 56001 Signalprozessor. Zusätzlich beinhaltet die ISPW eine "SoundSmith"-Schnittstelle, welche neben vier analogen und acht digitalen (AES/EBU) Ein- und Ausgängen auch über ein MIDI-Interface verfügt. Das Betriebssystem "Faster Than Sound" (FTS) ist eigens für die ISPW zuständig.

Die Rechenleistung für die Bearbeitung der Daten in Echtzeit wird von der ISPW bzw. vom FTS zur Verfügung gestellt, wobei die Soundfiles jedoch über einen NeXT-Host verwaltet werden.

Die Programmierung der ISPW erfolgt durch MAX - einer grafischen Programmierumgebung, welche auf dem Host läuft.

Mit MAX hat man die Möglichkeit, den Ablauf der Signal- und Steuerdaten für die Computermusik-Anwendungen grafisch zu erstellen und in Echtzeit zu beeinflussen.

MAX

Die einzelnen sog. "Patches" bilden dabei die grafischen Grundbausteine. Diese können elementare Funktionen wie Oszillator, MIDI-Daten-Operationen, Filter, etc. erfüllen, aber auch durch die Einbindung als "Subpatches" zu komplexeren Bausteinen verbunden werden. Zusätzlich besteht die Möglichkeit, C-Programme zu schreiben, welche als sog. "MAX externals" wiederum als grafische Grundbausteine in MAX zur Verfügung stehen.

Das Netzwerk dieser Grundbausteine bzw. MAX-Objekte funktioniert über die Versendung von Nachrichten entlang der Verbindung von Ausgängen zu Eingängen.

Dabei erfolgt eine Unterscheidung, ob es sich bei der Nachricht um Steuer- oder Signaldaten handelt. Dadurch wird der Rechenaufwand beträchtlich reduziert. Während die Signaldaten mit der Audio-Abtastrate (32 oder 44,1 kHz) bearbeitet werden, werden die Steuerdaten nur behandelt, wenn sich diese tatsächlich ändern. Die Kontrollrate beträgt 1/64 der Audio-Abtastrate.

Diese strenge Unterscheidung führt zu zwei MAX-Objekt-Klassen:

- Signal-Objekte:

In Signal-Objekten werden jeweils 64 Signalsamples zu einem Vektor zusammengefaßt und berechnet. Diese Vektoren werden von Signal-Objekt zu Signal-Objekt weitergereicht und entsprechend der Objekteigenschaften modifiziert. Dadurch ist aber ein rekursiver Aufbau des Netzwerkes nur unter Einschränkungen möglich; um beispielsweise ein rekursives Filter 1. Ordnung zu realisieren, müßte das Signal um ein Sample verzögert am Eingang wieder zur Verfügung stehen, was aber durch die 64-Sample-Verzögerung nicht möglich ist. In diesem Fall muß also ein MAX-external in C geschrieben werden.

- Message-Objekte:

Die Steuerdaten werden mit der niederen Taktrate verarbeitet, d. h., sie werden im Takt der 64-Sample-Vektoren der Signaldaten zu den nächsten Objekten weitergeschickt und lösen dort die entsprechenden Veränderungen aus.

4.1 VCO-Realisierungen

4.1.1 Impulsgenerator

Die Realisierung entspricht dem im Kap. 3.1.2.1.4 - "Gefensterte Sinc-Funktion" vorgestellten Algorithmus. Dabei wird eine ideale Impulsfunktion mit einer Sinc-Funktion gefaltet. Die Sinc-Funktion wird auf das zu berechnenden Sample gesetzt und die sich innerhalb der Sinc-Funktion befindenden idealen Impulse mit dem entsprechenden Funktionswerten gewichtet addiert.

Zur Darstellung der idealen Impulsfunktion dient eine Datenstruktur, die über Pointer zu einem Ringspeicher realisiert werden.

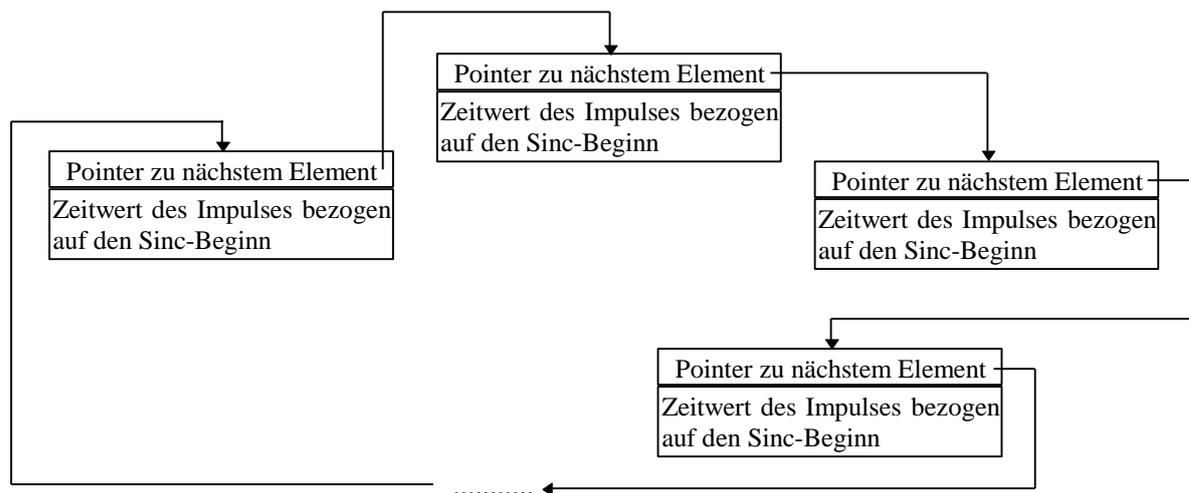


Bild 78: Ringspeicherstruktur mit Datenelementen.

Die Anzahl der Datenelemente in diesem Ringspeicher hängt von der zeitlichen Länge T_{Sinc} der Sinc-Funktion ab. Der Maximalwert I_{max} ergibt sich bei der höchstzulässigen Impulsfrequenz von $f_s / 2$ zu:

$$I_{max} = \frac{T_{Sinc} f_s}{2}$$

Beim Aufrufen des Patches wird der Ringspeicher mit zwei Impulsen initialisiert, welche sich am Anfang und am Ende des Sinc-Impulses befinden. Jedes Element der Struktur entspricht also einem Impuls der idealen Impulsfunktion. Zusätzlich verweist ein Pointer auf die jeweilige Struktur, um den Beginn (anf) und das Ende (end) der sich im Ringspeicher befindenden Impulse zu kennzeichnen, welche für die Berechnung des aktuellen Abtastwertes notwendig sind.

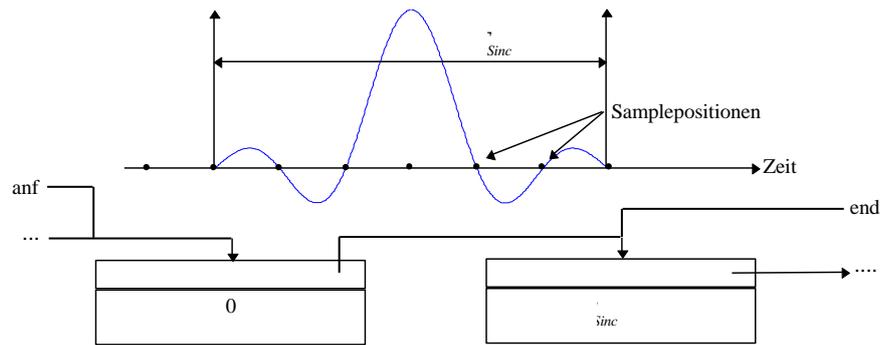


Bild 79: Initialisierung des Ringspeichers beim Patchaufruf. Der Zeitwert des ersten Elementes "zeigt" auf den Sinc-Impulsbeginn, das zweite Element auf dessen Ende.

Nach der Initialisierung erfolgt nun der rekursive Teil des Algorithmus. Dabei wird der Sinc-Impuls um einen Abtastwert T_s weiterverschoben, was durch die Subtraktion dieses Wertes von den Zeitwerten in der Datenstruktur erfolgt.

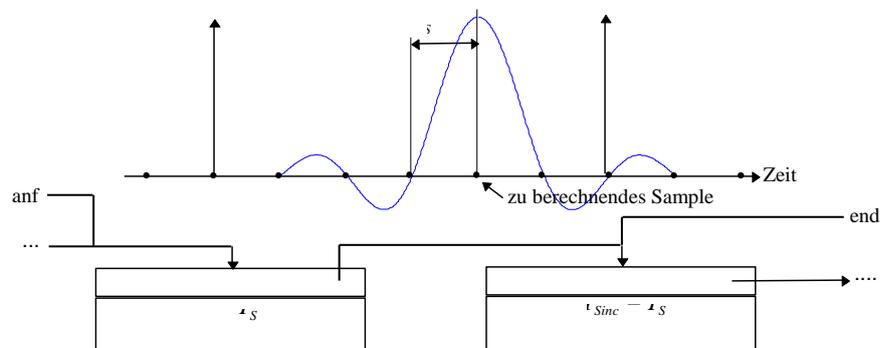


Bild 80: Der rekursive Teil der Berechnung beginnt mit dem Weiterschieben des Sinc-Impulses um einen Abtastwert, bzw. mit der Subtraktion der Abtastzeitdauer von den Zeitwerten der Elemente.

Vor der Berechnung des aktuellen Abtastwertes wird die aktuelle Frequenz f_1 berücksichtigt. Dabei wird ein neues Element am Ende der Struktur eingefügt, welches nun um den Zeitwert $T_1 = 1/f_1$ vom vorherigen Impuls "entfernt" ist. Schließlich wird auch der Pointer "end" auf dieses Element gesetzt.

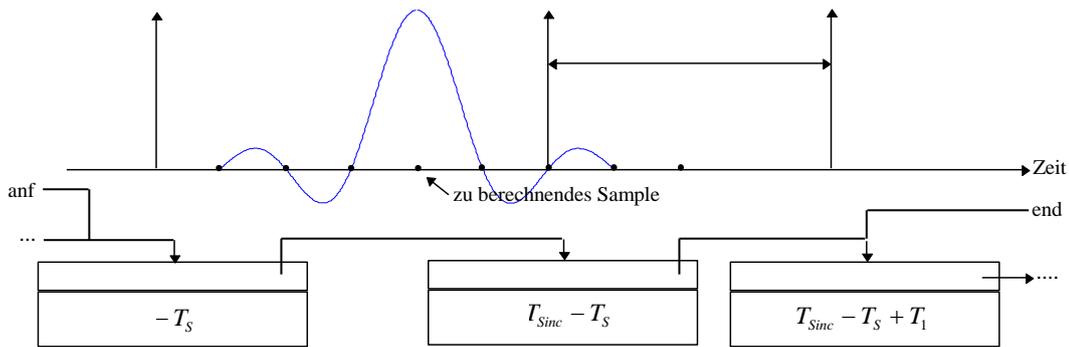


Bild 81: Ein neues Element wird hinzugefügt, wenn der Impuls des bisher letzten Elementes innerhalb des Sinc-Impulses fällt.

Um die Rekursion des Algorithmus zu gewährleisten, wird nur dann ein neues Element eingefügt, wenn sich außerhalb der Sinc-Funktion noch kein Element befindet - d. h., wenn der Zeitwert des letzten Elementes kleiner als T_{sinc} ist. Andernfalls wird lediglich der Zeitwert des Elementes entsprechend der Eingangsfrequenz verändert.

Weiters werden neue Impulse, welche sich durch eine plötzlich hohe Eingangsfrequenz innerhalb der Sinc-Funktion befinden würden, auf das Ende der Sinc-Funktion gesetzt. Somit befindet sich der letzte Impuls immer außerhalb oder am Ende der Sinc-Funktion und braucht bei der Berechnung der Faltung nicht berücksichtigt zu werden. (Dieser "Fehler" besteht nur für die Dauer eines Abtastwertes und bewirkt quasi eine Interpolation der Eingangsfrequenz, wenn sich diese innerhalb eines Samples auf sehr hohe Werte ($> 1/T_{sinc}$) verändert.)

Da Impulse, die schon außerhalb der Sinc-Funktion liegen - d. h. ihr Zeitwert ist kleiner Null - für die Faltung keinen Beitrag mehr liefern, können diese verworfen werden. Dabei ist zu beachten, daß ein Element mit einem Zeitwert kleiner Null berücksichtigt werden muß, um bei tiefen Eingangsfrequenzen - wenn der nächste Impuls außerhalb der Sinc-Funktion liegt - einen Bezugswert für die Berechnung des Zeitpunktes des nächsten Impulses zu haben. Wenn also die ersten zwei Elemente in der Struktur Zeitwerte kleiner Null haben, so kann das erste Element verworfen werden. Dies geschieht durch Weitersetzen des Pointers "anf" auf das nächste Element im Ringspeicher.

Schließlich erfolgt die eigentliche Berechnung des Samples durch Faltung der Sinc-Funktion mit den Impulsen. Diese erfolgt durch Summieren der Funktionswerte des Sinc-Impulses an den jeweiligen Stellen der Impulse. Das erste und letzte Element der Struktur braucht also nicht berücksichtigt zu werden. Die Sinc-Funktion wird dabei durch eine Tabelle realisiert, welche linear interpoliert wird.

4.1.2 Bipolarer Impulsgenerator

Durch eine Erweiterung der Struktur eines Elementes des Ringspeichers auf die Form

Pointer zu nächstem Element
Zeitwert des Impulses bezogen auf den Sinc-Beginn
Vorzeichen des Impulses

läßt sich durch Auswertung des Vorzeichens eine bipolare Impulsfunktion realisieren.

Wird der Struktur ein neuer Impuls hinzugefügt, so muß diesem lediglich das zum letzten Impuls gegenteilige Vorzeichen übergeben werden. Bei der Faltung werden dann die Funktionswerte der Sinc-Funktion mit dem entsprechenden Vorzeichen berücksichtigt.

4.1.3 Bipolarer Impulsgenerator mit Steuerung des Tastverhältnisses

Das Tastverhältnis des bipolaren Impulsgenerators läßt sich einfach durch dessen Berücksichtigung bei der Berechnung der Zeitwerte des in die Struktur neu hinzugefügten Impulses berücksichtigen.

Wenn der letzte Impuls positiv war, so wird das neue Element mit negativen Vorzeichen und dem resultierenden Zeitwert $T_1' = t_v / (100f_1)$ berücksichtigt, wobei der neue Eingangsparameter t_v in Prozent interpretiert wird.

Analog wird bei positiven Vorzeichen des neuen Elementes der Zeitwert zu $T_1' = (100 - t_v) / (100f_1)$ berechnet.

4.1.4 Rechteckgenerator

Der Rechteckgenerator wurde direkt als Patch realisiert, wobei dem oben erwähnten bipolaren Impulsgenerator ein nichtidealer Integrator in Form eines "2p2z"-Patches (Filter mit zwei Pol- und Nullstellen) nachgeschaltet wurde. Um Übersteuerungen des resultierenden Signals zu vermeiden, wird noch ein "gain"-Patch mit konstantem Verstärkungsfaktor nachgeschaltet.

4.1.5 Dreieckgenerator

Ein weiterer Integrator führt zu einem Dreiecksignal. Da die Flächen des Rechtecksignals bei unterschiedlichen Frequenzen zu verschiedenen Amplituden des Dreiecksignals führen, muß dieses mit einem frequenzabhängigen Faktor noch verstärkt werden. Weiters muß aufgrund der nichtidealen Integratoren eine Korrektur der Amplitude für tiefe Frequenzen vorgenommen werden.

4.1.6 Auswertung des VCO

Bei der Auswertung des VCO wurden lediglich verschiedenen 8192 Punkte-FFTs der unipolaren Impulsfunktion bei einer Grundfrequenz von 800 Hertz untersucht, da durch die Erweiterung derselben auf bipolare Impulsfunktion, Rechteck- und Dreieckfunktion keine neuen Frequenzen entstehen.

Die durchgezogene Linie in den Diagrammen zeigt die Stärke der Aliasing-Komponenten.

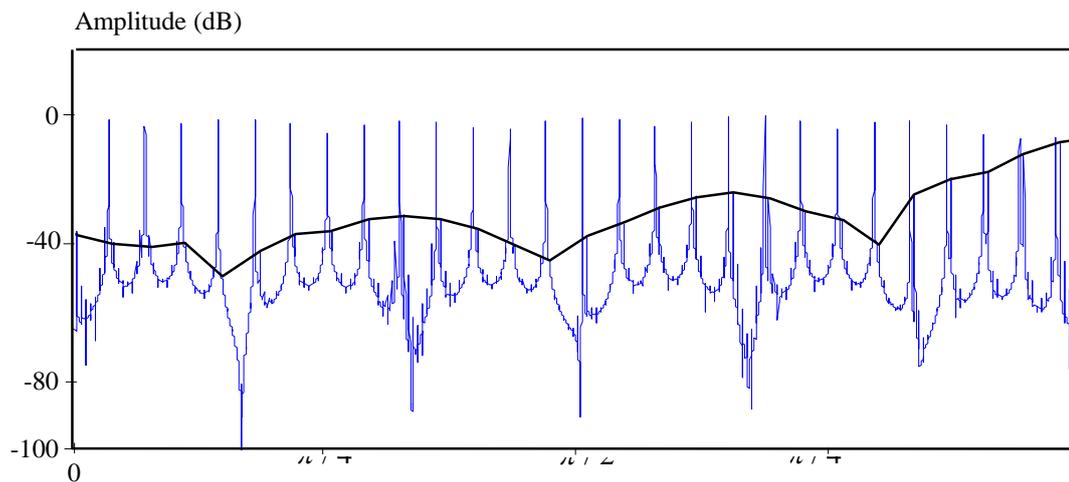


Bild 82: Spektrum einer Impulsfunktion, realisiert durch eine mit einem Rechteck gefensterten Sinc-Impuls mit einer Länge von $6 T_s$ und einer Hauptkeulenbreite von $2 T_s$. Der Sinc-Impuls wurde in eine 512-Punkte-Tabelle geschrieben.

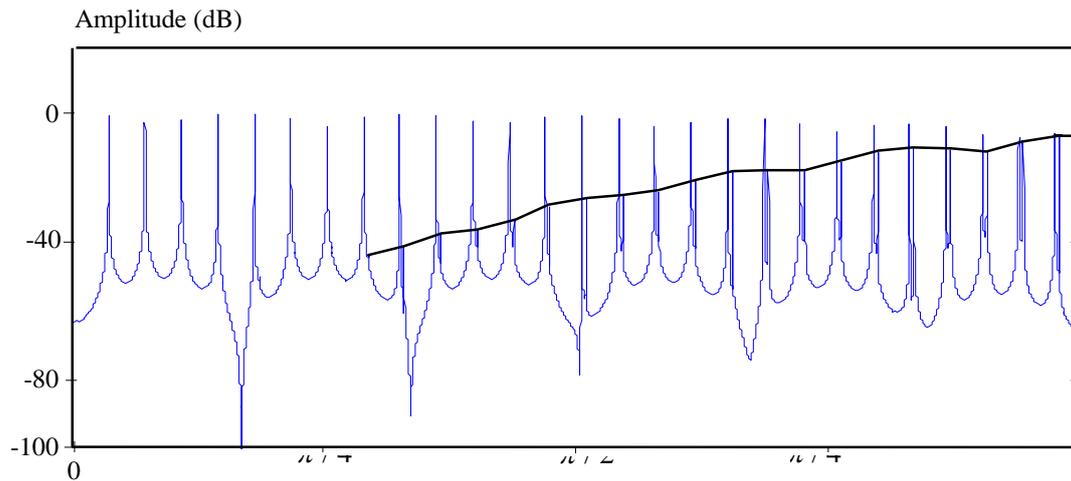


Bild 83: Spektrum einer Impulsfunktion, realisiert durch eine mit einem Blackman-Fenster gefensterten Sinc-Impuls mit einer Länge von $6 T_s$ und einer Hauptkeulenbreite von $2 T_s$. Der Sinc-Impuls wurde in eine 512-Punkte-Tabelle geschrieben.

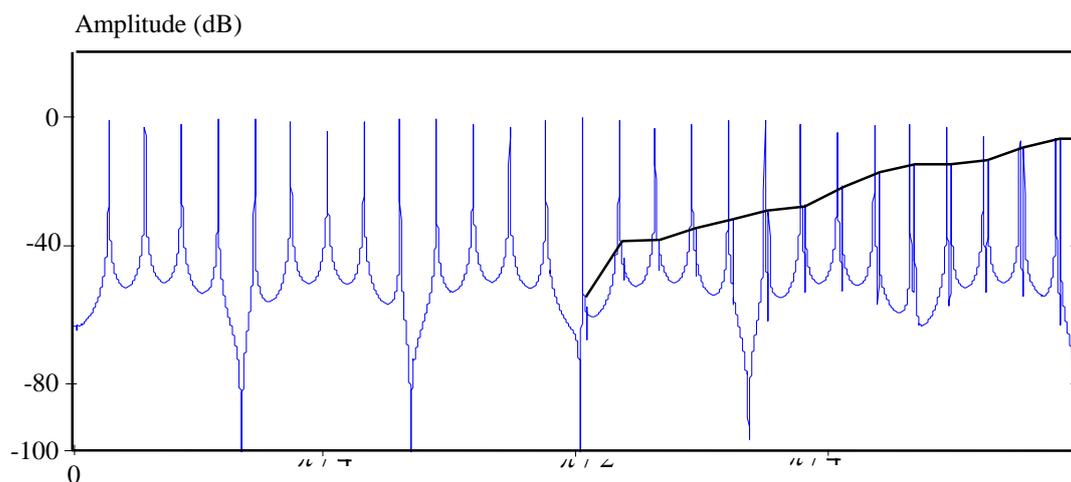


Bild 84: Spektrum einer Impulsfunktion, realisiert durch eine mit einem Blackman-Fenster gefensterten Sinc-Impuls mit einer Länge von $10 T_s$ und einer Hauptkeulenbreite von $2 T_s$. Der Sinc-Impuls wurde in eine 512-Punkte-Tabelle geschrieben.

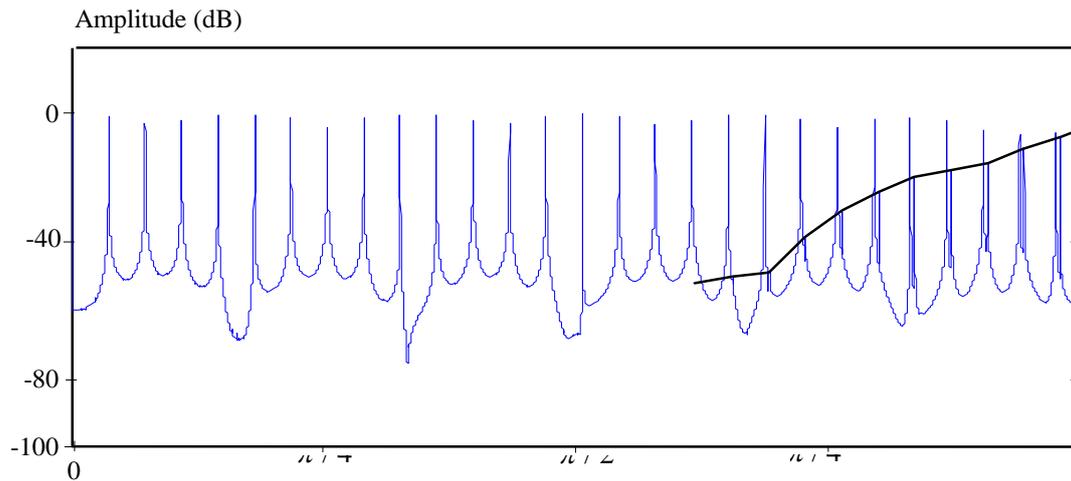


Bild 85: Spektrum einer Impulsfunktion, realisiert durch eine mit einem Blackman-Fenster gefensterten Sinc-Impuls mit einer Länge von $14 T_s$ und einer Hauptkeulenbreite von $2 T_s$. Der Sinc-Impuls wurde in eine 512-Punkte-Tabelle geschrieben.

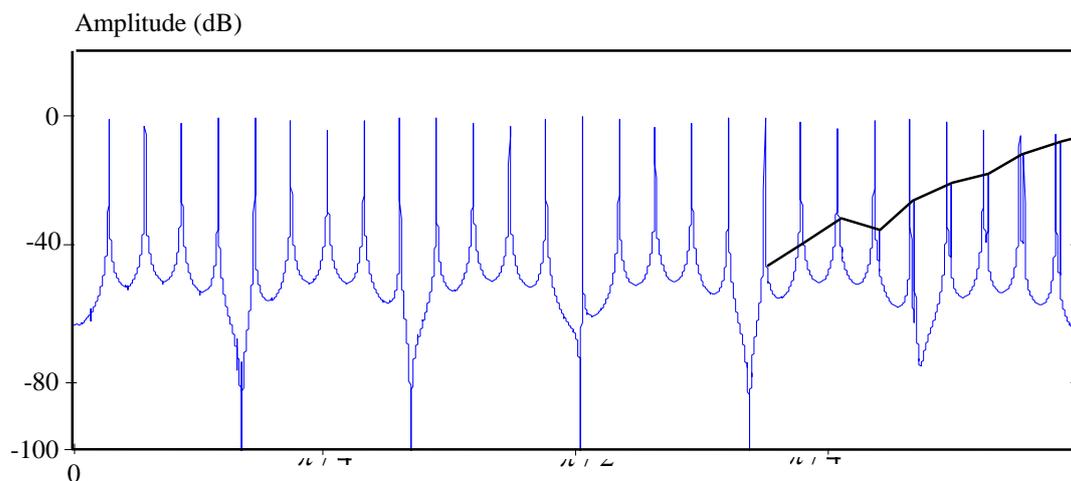


Bild 86: Spektrum einer Impulsfunktion, realisiert durch eine mit einem Blackman-Fenster gefensterten Sinc-Impuls mit einer Länge von $18 T_s$ und einer Hauptkeulenbreite von $2 T_s$. Der Sinc-Impuls wurde in eine 512-Punkte-Tabelle geschrieben.

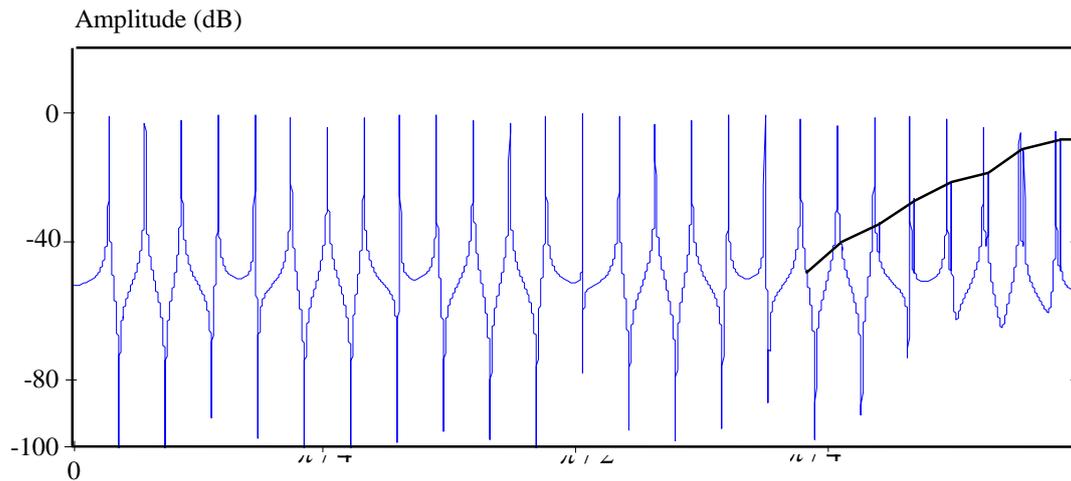


Bild 87: Spektrum einer Impulsfunktion, realisiert durch eine mit einem Blackman-Fenster gefensterten Sinc-Impuls mit einer Länge von $18 T_s$ und einer Hauptkeulenbreite von $2 T_s$. Der Sinc-Impuls wurde in eine 1024-Punkte-Tabelle geschrieben.

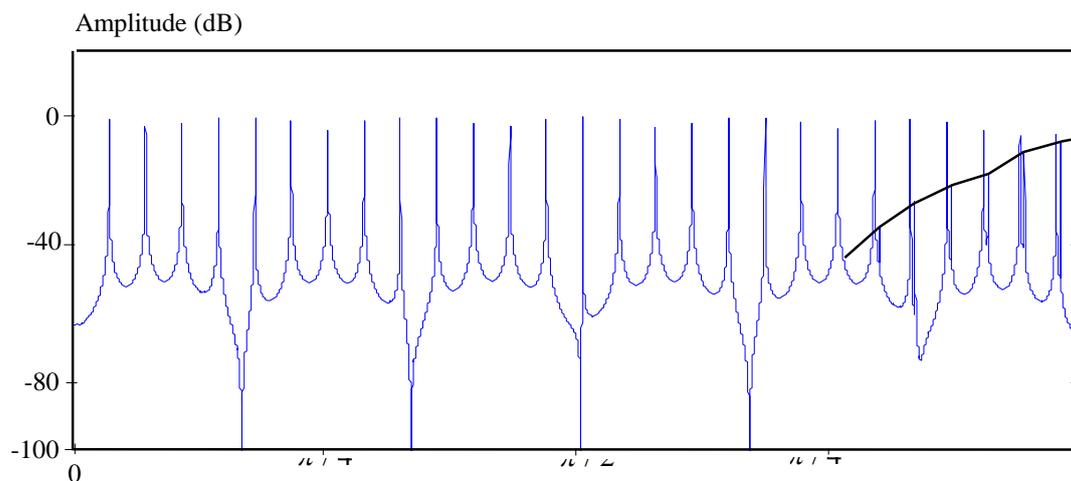


Bild 88: Spektrum einer Impulsfunktion, realisiert durch eine mit einem Blackman-Fenster gefensterten Sinc-Impuls mit einer Länge von $18 T_s$ und einer Hauptkeulenbreite von $2 T_s$. Der Sinc-Impuls wurde in eine 2048-Punkte-Tabelle geschrieben.

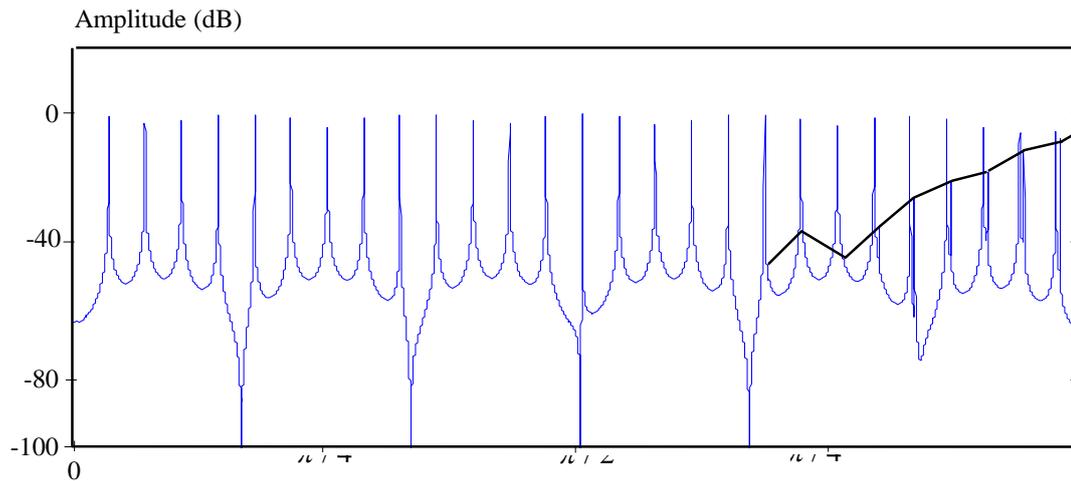


Bild 89: Spektrum einer Impulsfunktion, realisiert durch eine mit einem Blackman-Fenster gefensterten Sinc-Impuls mit einer Länge von $18 T_s$ und einer Hauptkeulenbreite von $2 T_s$. Der Sinc-Impuls wurde in eine 4096-Punkte-Tabelle geschrieben.

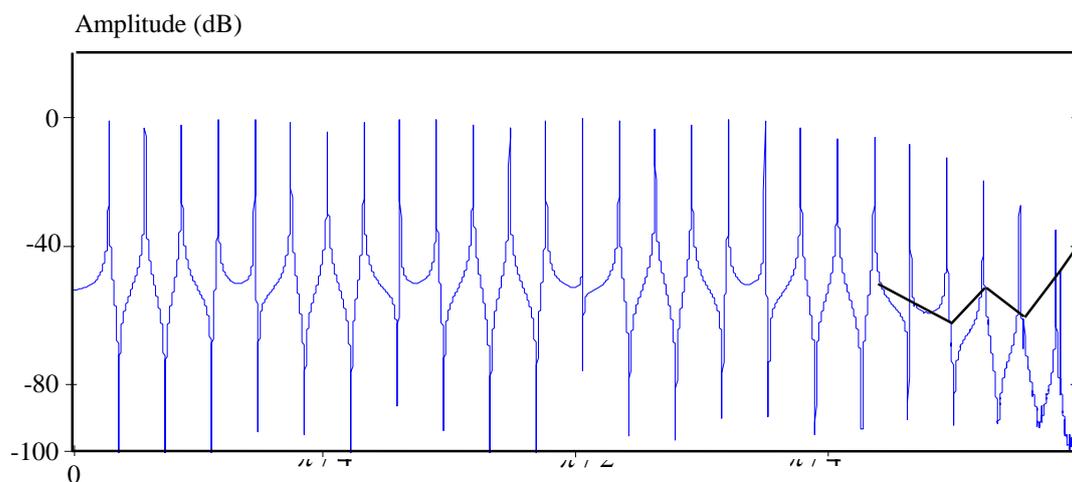


Bild 90: Spektrum einer Impulsfunktion, realisiert durch eine mit einem Blackman-Fenster gefensterten Sinc-Impuls mit einer Länge von $18 T_s$ und einer Hauptkeulenbreite von $2.44 T_s$. Der Sinc-Impuls wurde in eine 4096-Punkte-Tabelle geschrieben.

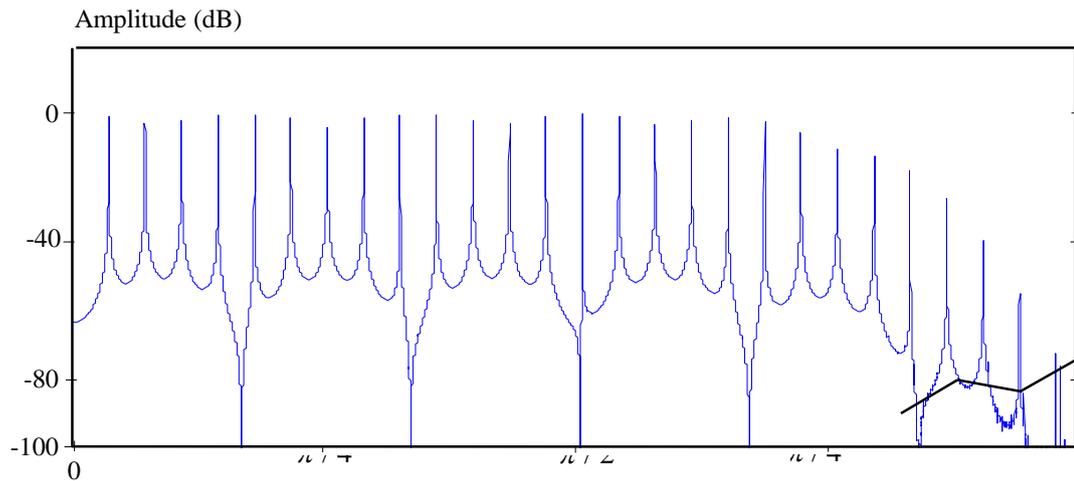


Bild 91: Spektrum einer Impulsfunktion, realisiert durch eine mit einem Blackman-Fenster gefensterten Sinc-Impuls mit einer Länge von $18 T_s$ und einer Hauptkeulenbreite von $2.66 T_s$. Der Sinc-Impuls wurde in eine 4096-Punkte-Tabelle geschrieben.

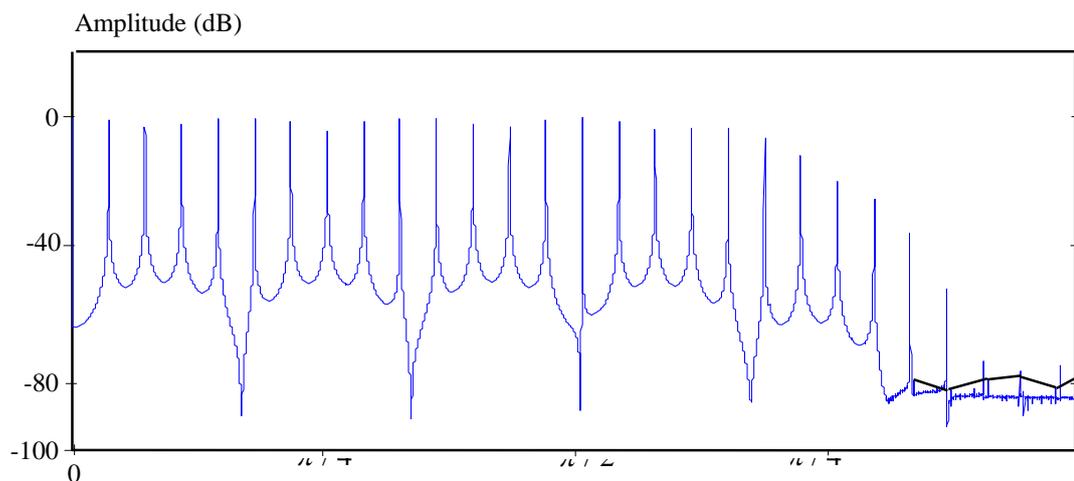


Bild 92: Spektrum einer Impulsfunktion, realisiert durch eine mit einem Blackman-Fenster gefensterten Sinc-Impuls mit einer Länge von $18 T_s$ und einer Hauptkeulenbreite von $2.88 T_s$. Der Sinc-Impuls wurde in eine 4096-Punkte-Tabelle geschrieben.

4.2 VCF-Realisierungen

4.2.1 Moog-VCF

Die Realisierung entspricht dem im Kapitel 3.6.2.2.1 - "Approximation des Tiefpaßfilters" vorgestellten Algorithmus. Dabei wurde die Filterkaskade und die Rückkopplungsschleife als MAX-external realisiert. Die Korrekturfunktion für die Frequenz wurde als "function-table" realisiert. Um Instabilitäten des Filters vorzubeugen, wurde zusätzlich eine Korrektur des Rückkopplungsfaktors vorgenommen. Dabei wurde die in Bild 62 dargestellte Kurve für $Q = \infty$ durch eine Funktion zweiter Ordnung angenähert.

4.2.2 Auswertung des Moog-VCF

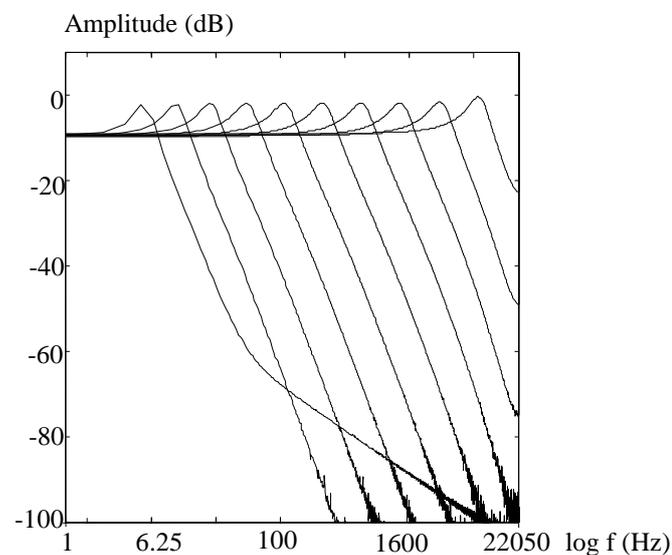


Bild 93: Übertragungsfunktionen des Moog-VCF bei den Grenzfrequenzen von 6.25, 12.5, 25, 50, 100, 200, 400, 800 und 1600 Hertz, sowie einem Rückkopplungsfaktor von 2.

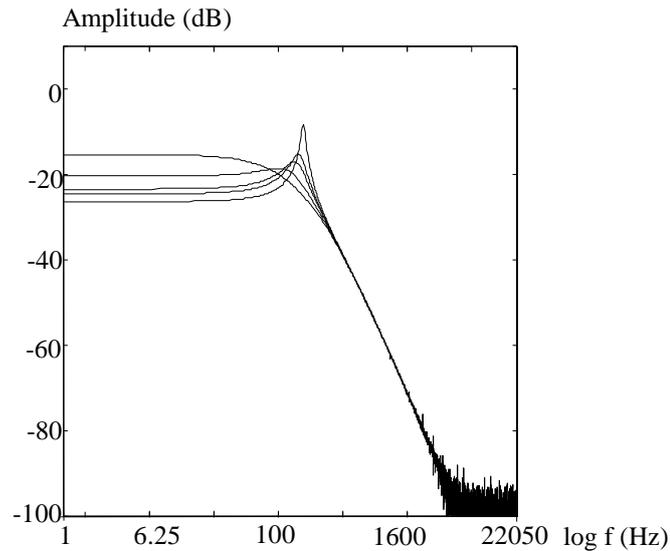


Bild 94: Übertragungsfunktionen des Moog-VCF bei den Rückkopplungsfaktoren von 0, 1, 2, 3, 4, und einer Grenzfrequenz von 100 Hertz.

4.2.3 Zustandsvariablen-Filter

Die Realisierung entspricht dem im Kapitel 3.6.4 - "Zustandsvariablen-Filter" behandelten Algorithmus. Wieder wurde die Differenzgleichung direkt in C als MAX-external implementiert.

4.2.4 Auswertung des Zustandsvariablen-Filters

Tiefpaßfilter

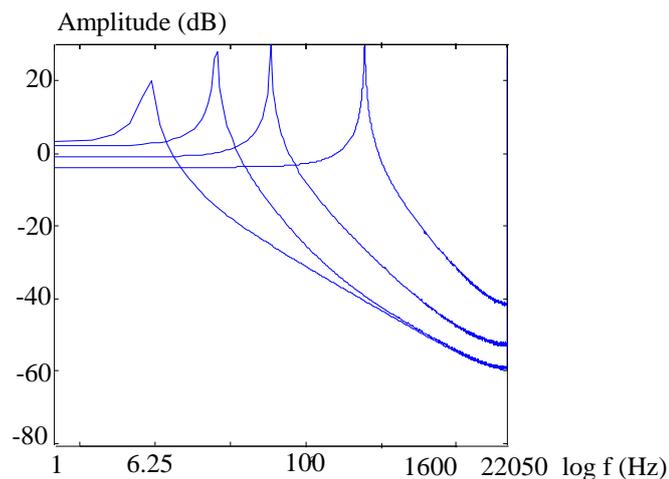


Bild 95: Übertragungsfunktion des Zustandsvariablen-Filters (Tiefpaßausgang) bei den Grenzfrequenzen 6.25, 25, 100 und 400 Hertz und einem Dämpfungsfaktor von 0.01.

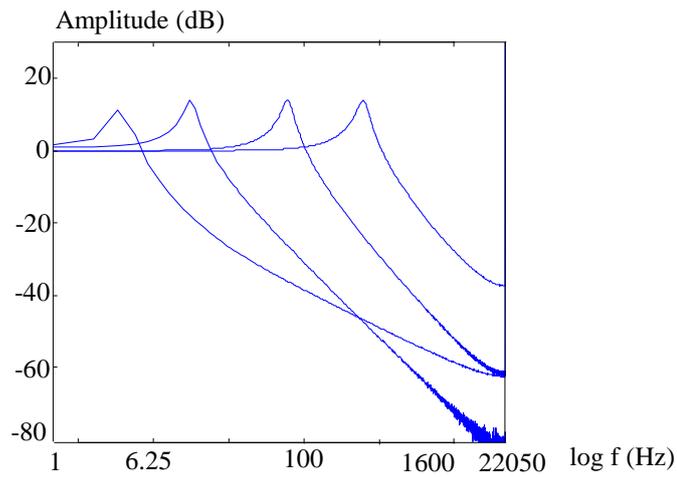


Bild 96: Übertragungsfunktion des Zustandsvariablen-Filters (Tiefpaßausgang) bei den Grenzfrequenzen 6.25, 25, 100 und 400 Hertz und einem Dämpfungsfaktor von 0.1.

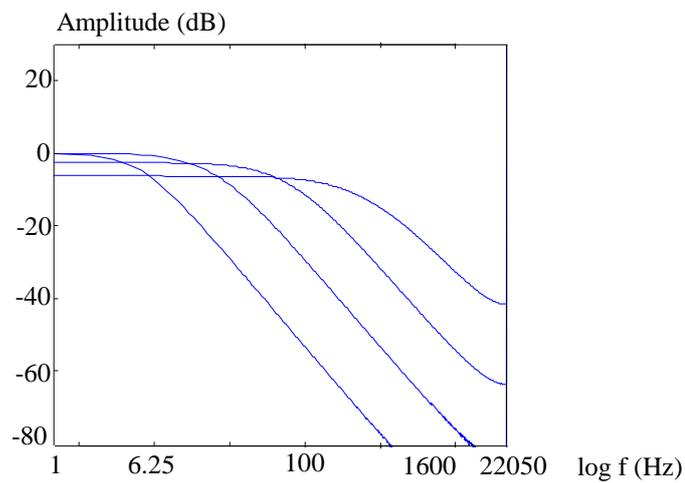


Bild 97: Übertragungsfunktion des Zustandsvariablen-Filters (Tiefpaßausgang) bei den Grenzfrequenzen 6.23, 25, 100 und 400 Hertz und einem Dämpfungsfaktor von 1.

Bandpaßfilter

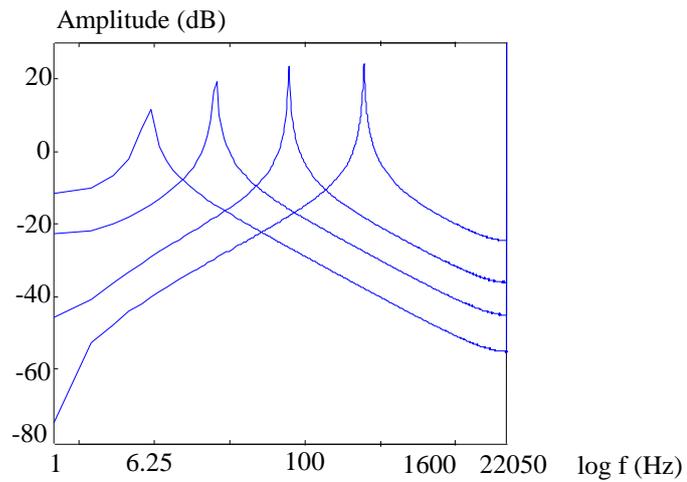


Bild 98: Übertragungsfunktion des Zustandsvariablen-Filters (Bandpaßausgang) bei den Grenzfrequenzen 6.23, 25, 100 und 400 Hertz und einem Dämpfungsfaktor von 0.01.

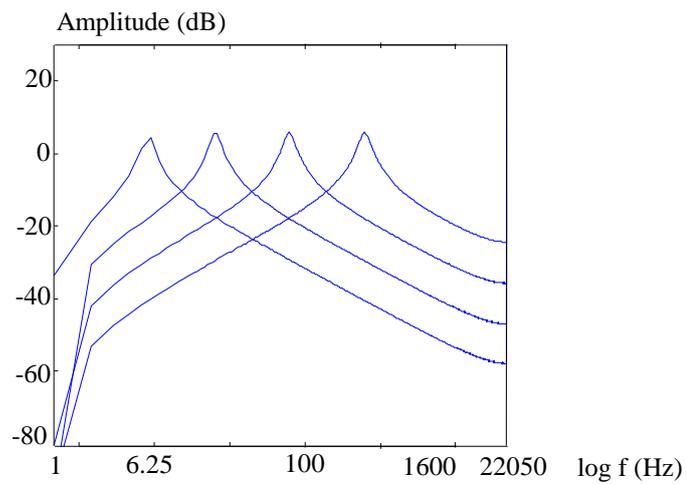


Bild 99: Übertragungsfunktion des Zustandsvariablen-Filters (Bandpaßausgang) bei den Grenzfrequenzen 6.23, 25, 100 und 400 Hertz und einem Dämpfungsfaktor von 0.1.

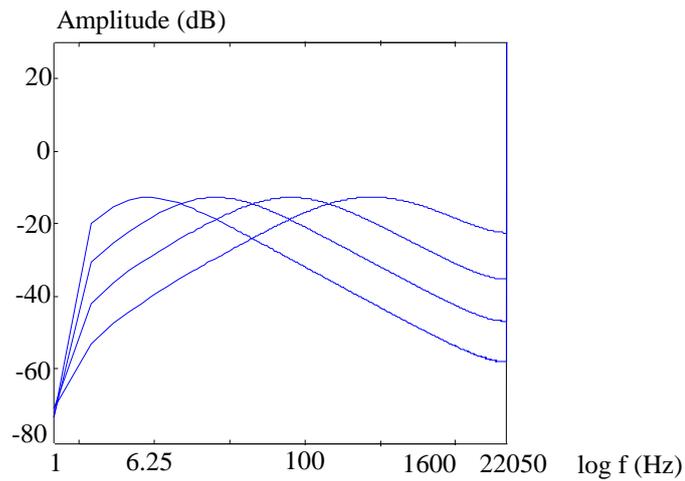


Bild 100: Übertragungsfunktion des Zustandsvariablen-Filters (Bandpaßausgang) bei den Grenzfrequenzen 6.23, 25, 100 und 400 Hertz und einem Dämpfungsfaktor von 1.

Hochpaßfilter

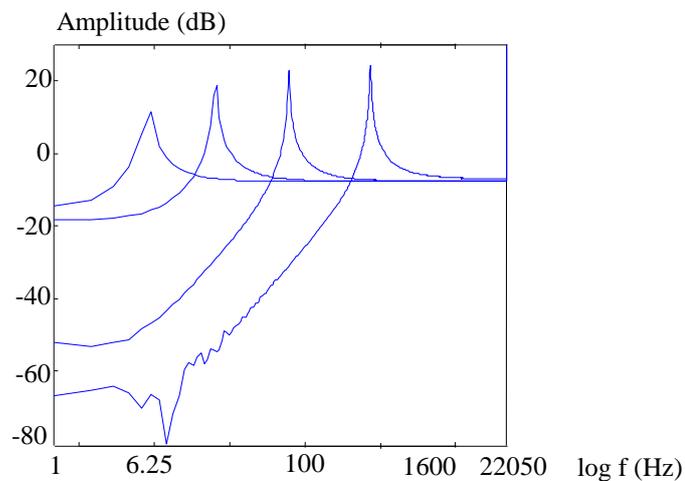


Bild 101: Übertragungsfunktion des Zustandsvariablen-Filters (Hochpaßausgang) bei den Grenzfrequenzen 6.23, 25, 100 und 400 Hertz und einem Dämpfungsfaktor von 0.01.

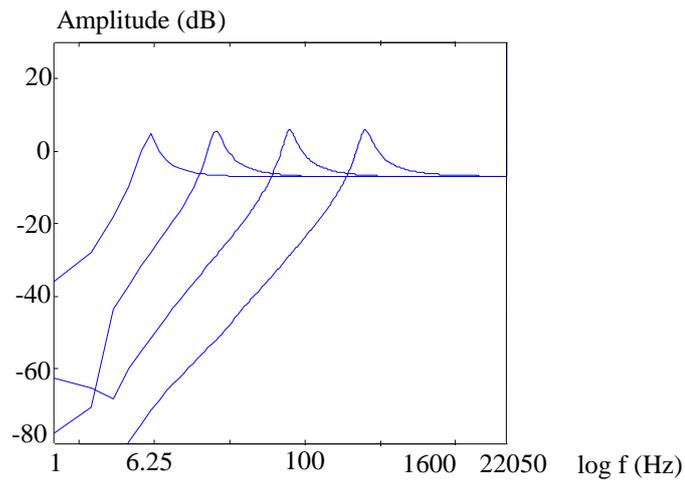


Bild 102: Übertragungsfunktion des Zustandsvariablen-Filters (Hochpaßausgang) bei den Grenzfrequenzen 6.23, 25, 100 und 400 Hertz und einem Dämpfungsfaktor von 0.1.

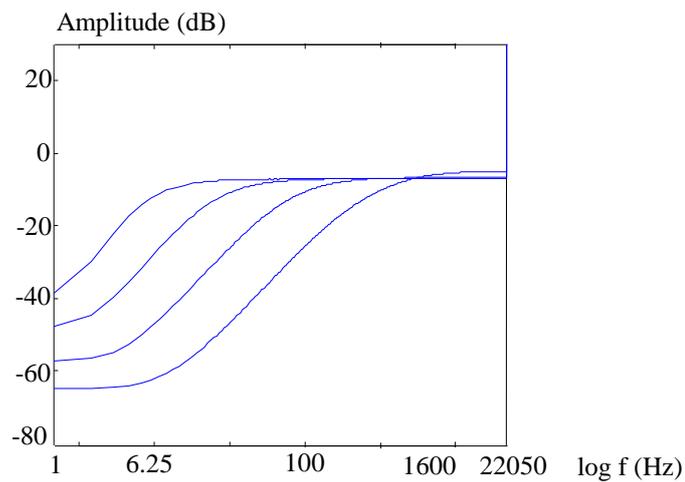


Bild 103: Übertragungsfunktion des Zustandsvariablen-Filters (Hochpaßausgang) bei den Grenzfrequenzen 6.23, 25, 100 und 400 Hertz und einem Dämpfungsfaktor von 1.

Hörbeispiele

Nr.	Beschreibung
1	Analoges Impulssignal (27.5-14080Hz logarithmisch)
2	Digitales Impulssignal (27.5-14080Hz logarithmisch) zeitlich gerundet (siehe Kap. 3.1.2.1.3)
3	Digitales Impulssignal (27.5-14080Hz logarithmisch) linear interpoliert (siehe Kap. 3.1.2.1.3)
4	Digitales Impulssignal (27.5-14080Hz logarithmisch) Sinc-Daten: Table 512, Breite: $2 T_s$, Rechteckfenster, Hauptkeule $2 T_s$
5	Digitales Impulssignal (27.5-14080Hz logarithmisch) Sinc-Daten: Table 512, Breite: $2 T_s$, Blackman-Fenster, Hauptkeule $2 T_s$
6	Digitales Impulssignal (27.5-14080Hz logarithmisch) Sinc-Daten: Table 512, Breite: $4 T_s$, Blackman-Fenster, Hauptkeule $2 T_s$
7	Digitales Impulssignal (27.5-14080Hz logarithmisch) Sinc-Daten: Table 512, Breite: $8 T_s$, Blackman-Fenster, Hauptkeule $2 T_s$
8	Digitales Impulssignal (27.5-14080Hz logarithmisch) Sinc-Daten: Table 512, Breite: $6 T_s$, Rechteckfenster, Hauptkeule $2 T_s$ (siehe Bild 82)
9	Digitales Impulssignal (27.5-14080Hz logarithmisch) Sinc-Daten: Table 512, Breite: $6 T_s$, Blackman-Fenster, Hauptkeule $2 T_s$ (siehe Bild 83)
10	Digitales Impulssignal (27.5-14080Hz logarithmisch) Sinc-Daten: Table 512, Breite: $10 T_s$, Blackman-Fenster, Hauptkeule $2 T_s$ (siehe Bild 84)
11	Digitales Impulssignal (27.5-14080Hz logarithmisch) Sinc-Daten: Table 512, Breite: $14 T_s$, Blackman-Fenster, Hauptkeule $2 T_s$ (siehe Bild 85)
12	Digitales Impulssignal (27.5-14080Hz logarithmisch) Sinc-Daten: Table 512, Breite: $18 T_s$, Blackman-Fenster, Hauptkeule $2 T_s$ (siehe Bild 86)
13	Digitales Impulssignal (27.5-14080Hz logarithmisch) Sinc-Daten: Table 1024, Breite: $18 T_s$, Blackman-Fenster, Hauptkeule $2 T_s$ (siehe Bild 87)
14	Digitales Impulssignal (27.5-14080Hz logarithmisch) Sinc-Daten: Table 2048, Breite: $18 T_s$, Blackman-Fenster, Hauptkeule $2 T_s$ (siehe Bild 88)
15	Digitales Impulssignal (27.5-14080Hz logarithmisch) Sinc-Daten: Table 4096, Breite: $18 T_s$, Blackman-Fenster, Hauptkeule $2 T_s$ (siehe Bild 89)
16	Digitales Impulssignal (27.5-14080Hz logarithmisch) Sinc-Daten: Table 4096, Breite: $18 T_s$, Blackman-Fenster, Hauptkeule $2.44 T_s$ (siehe Bild 90)

Nr.	Beschreibung
17	Digitales Impulssignal (27.5-14080Hz logarithmisch) Sinc-Daten: Table 4096, Breite: $18 T_s$, Blackman-Fenster, Hauptkeule $2.66 T_s$ (siehe Bild 91)
18	Digitales Impulssignal (27.5-14080Hz logarithmisch) Sinc-Daten: Table 4096, Breite: $18 T_s$, Blackman-Fenster, Hauptkeule $2.88 T_s$ (siehe Bild 92)
19	Analoges Rechtecksignal (27.5-14080Hz logarithmisch)
20	Digitales Rechtecksignal mit Aliasing (27.5-14080Hz logarithmisch)
21	Analoges Rechtecksignal mit Tastverhältnissteuerung
22	Analoges Dreiecksignal (27.5-14080Hz logarithmisch)
23	Digitales Dreiecksignal mit Aliasing (27.5-14080Hz logarithmisch)
24	Analoges Dreiecksignal mit Tastverhältnissteuerung
25	Musikbeispiel (Original)
26	Musikbeispiel mit Flanger
27	Musikbeispiel mit Phaser
28	Musikbeispiel mit VCF (Modulation der Grenzfrequenz und des Rückkopplungsfaktors)
29	Ringmodulator: 1. Eingang: Musiksignal (perkussiv), 2. Eingang: Sinusschwingung mit langsam steigender Frequenz
30	Ringmodulator: 1. Eingang: Musiksignal (Gesang) 2. Eingang: Sinusschwingung mit konstanter Frequenz

Literatur

- [1] J. W. Gordon & J. O. Smith: "A sine generation algorithm for VLSI application"; Proceedings of the 1985 International Computer Music Conference; pp. 165-168; 1985
- [2] J. O. Smith: "The second-order digital waveguide oscillator"; Proceedings of the 1992 International Computer Music Conference, San Jose; pp. 150-153; 1992
- [3] F. R. Moore: "Table lookup noise for sinusoidal digital oscillators"; Computer Music Journal; vol. 1(1); pp. 26-29; 1977
- [4] I. N. Bronstein & K. A. Semendjajew: "Taschenbuch der Mathematik"; Verlag Harri Deutsch; 1985; ISBN3-87144-492-8; Kap. 7.1.2.6
- [5] F. de Bernadinis, R. Roncella, R. Saletti, P. Terreni & G. Bertini: "A new VLSI implementation of additive synthesis"; Computer Music Journal; vol. 22(3); pp. 49-61; 1998
- [6] T. Stilson & J. O. Smith: "Alias-free digital synthesis of classic analog waveforms"; Proceedings of the 1996 International Computer Music Conference, Hong Kong; pp. 332-335; 1996
- [7] J. A. Moorer: "The synthesis of complex audio spectra by means of discrete summation formulas"; Journal of the AES, vol. 24(9); pp. 717-727; 1976
- [8] W. Godfrey & K. Steiglitz: "Input generators for digital sound synthesis"; Journal of the acoustic society of America; vol. 47(2); pp. 665-666; 1970
- [9] C. Dodge & T. A. Jerse: "Computer music - synthesis, composition, and performance"; Schirmer Books; 1977; ISBN 0-02-873100-X; Kap. 4.3
- [10] J. O. Smith & P. Gossett: "A flexible sampling-rate conversion method"; Proceedings of the Intern. Conf. on acoustics, speech and signal processing, San Diego; vol. 2; pp. 19.4.1-19.4.2; 1984
- [11] M. Goodwin & A. Kogon: "Overlap-add synthesis of nonstationary sinusoids"; Proceedings of the 1995 International Computer Music Conference; pp. 355-356; 1995
- [12] X. Rodet & Ph. Depalle: "A new additive synthesis method using inverse Fourier transform and spectral envelopes"; Proceedings of the 1992 International Computer Music Conference; pp 410-411; 1992

-
- [13] M. Goodwin & X. Rodet: "Efficient Fourier synthesis of nonstationary sinusoids"; Proceedings of the 1994 International Computer Music Conference; pp. 333-334; 1994
- [14] J. Lane, D. Hoory, E. Martinez & P. Wang: "Modeling analog synthesis with DSPs"; Computer Music Journal, vol. 21(4); pp. 23-41; 1997
- [15] F.R. Moore: "Elements of computer music"; Englewood Cliffs: Prentice-Hall; 1990; ISBN 0-13-252552-b; pp. 408-413 & pp. 439-453;
- [16] J. Dattorro: "The implementation of recursive digital filters for high-fidelity audio"; AES Journal; vol. 36(11); pp. 851-878; 1988
- [17] R. Wilson: "Filter Topologies"; AES Journal; vol. 41(9); pp. 667-677; 1993
- [18] U. Zölzer: "Roundoff error analysis of digital filters"; AES Journal; vol. 42(4); pp. 232-244; 1994
- [19] V. Välimäki, T. I. Laakso & J. Mackenzie: "Elimination of transients in time-varying allpass fractional delay filters with application to digital waveguide modeling"; Proceedings of the 1995 International Computer Music Conference; pp. 327-334; 1995
- [20] T. Stilson & J. O. Smith: "Analyzing the Moog VCF with considerations for digital implementation"; Proceedings of the 1996 International Computer Music Conference; pp. 398-401; 1996
- [21] U. Zölzer: "Digitale Audiosignalverarbeitung"; B. G. Teubner Stuttgart; Kap. 5.1.2; ISBN 3-519-06180-5; 1996
- [22] P. Dutilleux: "Spinning the sounds in real-time"; Proceedings of the 1989 International Computer Music Conference; pp. 94-97; 1989
- [23] D. C. Massie: "An engineering study of the four-multiply normalized ladder filter"; AES Journal; vol. 41(7/8); pp. 564-582; 1993
- [24] S. J. Orfanidis: "Introduction to signal processing"; Prentice Hall New Jersey; ISBN 0-13-209172-0; pp. 360-367; 1996
- [25] T. I. Laakso, V. Välimäki, M. Karjalainen & U. K. Laine: "Splitting the unit delay"; IEEE Signal Processing Magazine 1996/1; pp. 30-60; 1996
- [26] A. V. Oppenheim & R. W. Schaffer: "Zeitdiskrete Signalverarbeitung"; R. Oldenbourg Verlag; 1995; ISBN 3-486-22948-6; Kap. 6.7