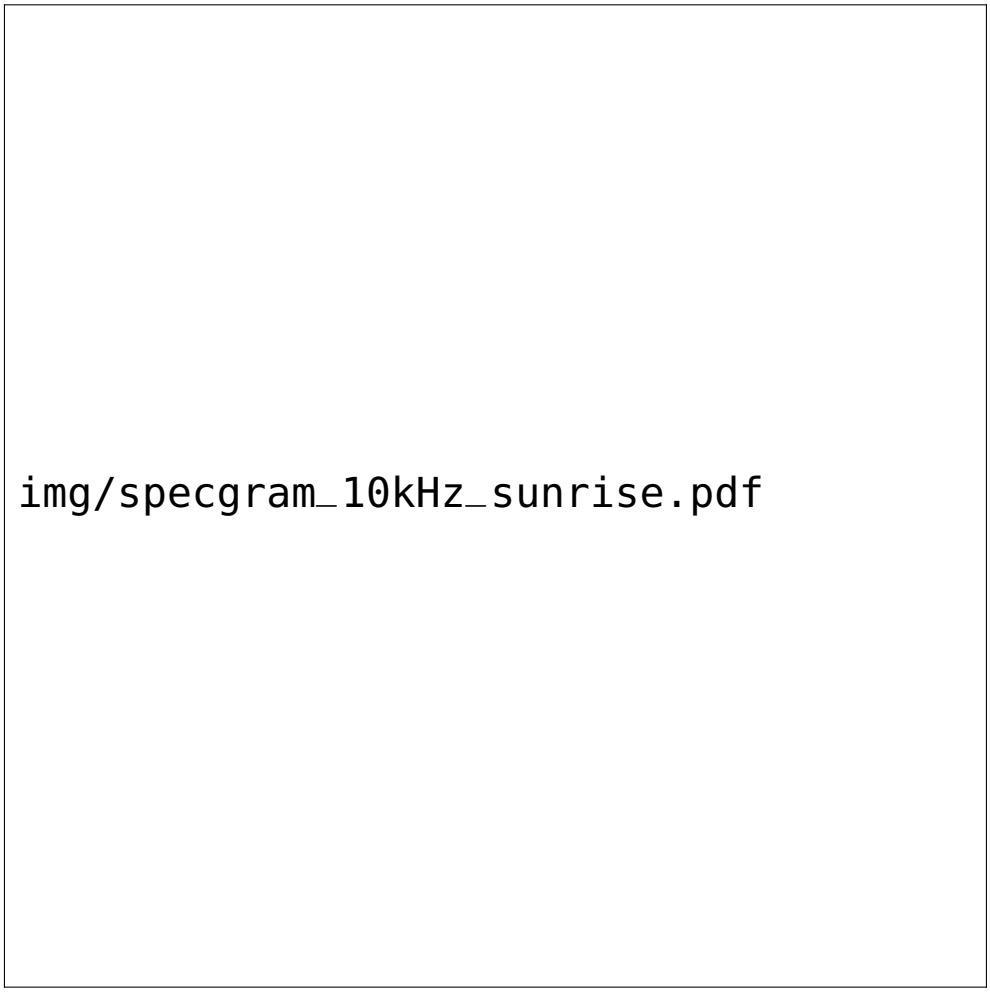
Asynchronous Sample Rate Conversion

ASRC

Prof. Dr. Christian Münker



sinc-Funktion

5. November 2014

Christian.Muenker@hm.edu

Inhaltsverzeichnis

1 Überblick

Mit dem Siegeszug der digitalen Signalverarbeitung kam das Problem auf, Mess-, Audio-, Foto- und Videoformaten zwischen verschiedenen Abtastraten (bzw. Auflösungen bei Fotos und Videos) möglichst verlustarm zu konvertieren. Diese Problematik ist eng verwandt mit Techniken der *Interpolation*, also der Problemstellung zu einer Reihe von Zahlenwerten verlässliche Zwischenwerte zu ermitteln. [Mei02] beschreibt die Geschichte der Interpolation von der Antike bis in das 20. Jahrhundert und ordnet sie so in einen größeren Kontext ein.

Zu einer Anzahl von diskreten Stützstellen wird eine kontinuierliche Funktion bestimmt (*kontinuierlicher Interpolator*). Diese Funktion kann abschnittsweise definiert sein, ist an jedem Punkt auswertbar und stimmt an den Stützstellen mit diesen exakt (Lagrange-Interpolation: ZOH, lineare, quadratische, kubische, ... Interpolation) oder näherungsweise (Spline-Approximation) überein. Die Abtastwerte zu den gewünschten (neuen) Zeitpunkten werden mit Hilfe dieser zeitkontinuierlichen Funktion berechnet, dieser Prozess kann in Software mit hoher Genauigkeit angenähert werden. Zur Veranschaulichung kann dieser Prozess als **analoges System** modelliert werden (Abb. 1).

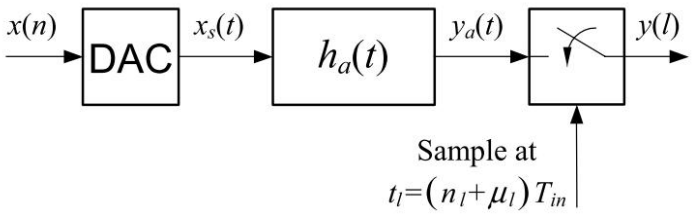


Abb. 1: Analoges Modell der ASRC [DB]

$x_S(t)$ ist dabei ein ideal gewandeltes Analogsignal (zeitkontinuierliche Diracstöße, kein Zero-Order Hold!), $h_a(t)$ ist die zeitkontinuierliche Impulsantwort des kontinuierlichen Interpolators. Sowohl DAC als auch der kontinuierliche Interpolator sind nur Modelle, für die es kein

Hardware-Äquivalent gibt! Im Folgenden wird der DAC als Teil des kontinuierlichen Interpolators betrachtet. Der Sampler modelliert die Ermittlung des Ausgangswerts bei den gewünschten neuen Zeitpunkten (*Dezimation*). Ein großer Vorteil dieses Modells ist, dass es im Zeit- und Frequenzbereich untersucht werden kann. Die praktische Realisierung ist aber nicht trivial, da im Allgemeinen die zeitliche Relation zwischen Ein- und Ausgangssamples bestimmt werden muss - die eben gerade nicht synchron zu einander laufen. Daher benötigt man normalerweise eine *analoge* Zeitmessung (PLL, DLL, TDC, ...) zur Ermittlung der neuen Ausgangswerte.

In der **Mathematik** heißt „Interpolation“ eine glatte Kurve zu konstruieren, die *durch* alle gegebenen Punkte geht. Die Konstruktion einer Kurve, die so *nah* wie möglich an alle gegebenen Punkte herankommt, wird *Approximation* genannt (z.B. Least-Squares Approximation). [Pin] behandelt z.B. die Anfänge der Approximationstheorie nach Weierstraß, [Fom00] gibt eine funktionalanalytische Beschreibung der Interpolation mithilfe von Basisfunktionen (z.B. für die Approximation mittels B-Splines). Generell wird in der Mathematik Interpolation als ein Polynom-Fitting Problem (z.B. Lagrange) in der Zeit- bzw. Ortsebene betrachtet. Für DSP-Anwendungen ist diese Betrachtungsweise nicht besonders hilfreich, da i.A. die Zeiteigenschaften von Signalen nicht a priori bekannt sind. Hier ist eine Betrachtung in der Frequenzebene einfacher, da meist die spektralen Eigenschaften der Signale (Bandbreite etc.) bekannt sind.

In der **digitalen Signalverarbeitung** wird daher Interpolation primär als Filterung in der Frequenzebene betrachtet; die Kombination aus Upsampling (= Erhöhung der Abtastrate durch Nullenstopfen) und Anti-Image Filter wird Interpolator genannt - daraus resultiert aber nicht immer eine Interpolation im mathematischen Sinn! Filter, die auch im mathematischen Sinn interpolieren, werden *Nyquist-Filter* genannt (= frei von Intersymbol Interferenz).

Einen guten Einstieg in diese Betrachtungsweise gibt das Vorlesungsskript „Polynomial-based Interpolation Filters for DSP Applications“ von Djordje Babic (Uni Tampere, Finnland) [DB]. Unter [LBR04] findet sich eine interessante Gegenüberstellung unterschiedlicher Modelle von Interpolationsfiltern. In [SR73] gibt Ronald Schafer schließlich eine gute Übersicht über Interpolation in der Signalverarbeitung, Abtastratenkonvertierung, Lagrange Interpolatoren und über den Entwurf von FIR Interpolationsfiltern.

Die folgenden Authoren bauen alle auf dem analogen Model der Interpolation auf:

Julius O. Smith [Smi14]„Digital Audio Resampling Home Page“ (`https://ccrma.stanford.edu/~jos/resample/resample.html`, auch als pdf). Das Skript zur Vorlesung findet sich in [Smi11].

Gennaro Evangelista hat in seiner Dissertation [Eva00] ein mathematisches Modell für Abtastratenwandler vorgestellt und der Entwurf von Interpolatoren behandelt, immer mit Blick auf die Implementierbarkeit. Die Herangehensweise ist dabei (nur) so mathematisch wie es die Thematik erfordert und nicht unbedingt für den Einstieg geeignet. Eine Kurzfassung bietet das abgeleitete Paper Journal Paper [Eva03a]. Das Thema wird weiter vertieft in [Eva03b] (Quantisierungseffekte).

Ivar Løkken [Lok05] unterscheidet zwischen „arbitrary sample rate conversion“ (ASRC) und „asynchronous arbitrary sample rate conversion“ (AASRC), Evangelista redet von Generalized Arbitrary Sample Rate Conversion.

Roman Kappeler und David Grünert [KG04] schildern in einer Studienarbeit (???) an der ETH Zürich den Entwurf eines ASICs zur ASRC von 192 kHz, 24 Bit Audiosignalen mit 150 dB SNR. Zur Interpolation werden hier B-Splines verwendet.

F. Francesconi schildert in [FLL+93a] und [FLL+93b] den Entwurf von multiplizierenlosen Interpolatoren für feste Upsamplingraten, die Lagrange-Interpolation anstelle von sinc-Interpolation verwenden und damit einen deutlich flacheren Verlauf im Passband und besser Imagedämpfung erzielen.

Wichtige Begriffe

Sample Rate (Abtastrate) die Rate, mit der ein zeitdiskretes Signal vorliegt

Upsampling Erhöhen der Abstastrate um einen ganzzahligen Faktor, die fehlenden Samples werden durch Null gefüllt (zero-stuffing) oder durch Wiederholen des letzten Werts (Zero-Order Hold). Dabei entstehen Kopien des ehemaligen Basisbands bis zur neuen -> Nyquistfrequenz, sog. -> Images, die durch Anti-Image Filter entfernt werden müssen.

Downsampling Verringerung der Abtastrate um einen ganzzahligen Faktor durch Weglassen von Abtastwerten. Ohne vorherige Bandbreitenbegrenzung (Anti-Alias Filterung) besteht die Gefahr von -> Aliasing.

Resampling / Abtastratenwandlung: Systeme bestehend aus Auf- bzw. Abwärtstastern, Filtern und/oder Interpolatoren

Synchron wenn sich das Verhältnis von zwei Abtastraten durch ein einfachen Bruch ausdrücken lässt, z.B. beim Resampling von 44.1kHz nach 48kHz, 44.1/48 = 147 / 160 .

Asynchron wenn die zwei Abtastraten kein einfaches rationales Verhältnis haben

Fractional-Delay-Filter: Ein kontinuierlicher Interpolator, aus dessen Ausgangssignal nur ein Wert entnommen wird

Interpolationsfilter: Filter zum Entfernen der Images bei Aufwärtstastung um einen ganzzahligen Faktor

Effiziente Strukturen: Polyphasen-Filter, Farrow-Filter

1.1 Aufbau

Dieses Paper fokussiert sich auf *Real-Time Abtastratenwandlung von Audiosignalen* in Software und Hardware, also *bandbegrenzten* Signalen. Die effiziente Implementierung von Interpolation sowie Verfahren/Metriken, die Qualität der Ausgangssignale zu bewerten sind wichtige Teilaspekte.

2 Interpolationsverfahren

Alle hier betrachteten Verfahren lassen sich mit dem gleichen Modell Abb. 1 analysieren. Aus praktischen Gründen, die im Folgenden klar werden, wird die Abtastrate $f_{S,1}$ des zeitdiskreten Eingangssignals zunächst durch Nullenstopfen um den Faktor L erhöht (Abb. 2). Aufgrund dieses Upsamplings entstehen Images bei $kf_{S,1}$ mit $k = 1 \ldots L/2$, die vor der Dezimation weitestgehend eliminiert werden sollten.

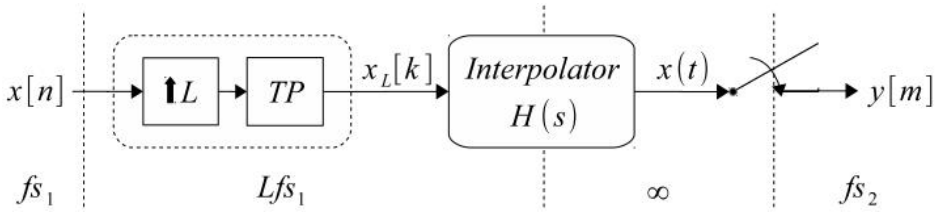
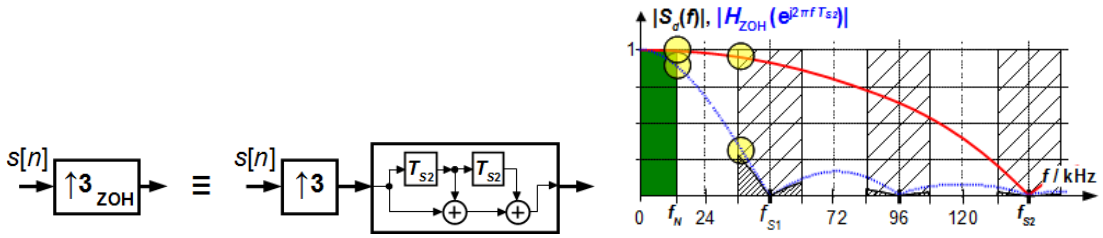


Abb. 2: Analoges Modell der ASRC mit Upsampling

Soll der letzte Wert wiederholt werden (ZOH), entspricht das einem Moving Average Filter mit $M = L$, das hohe Frequenzen dämpft mit $\mathrm{di}\,\pi f T_{S,1}$ (Dämpfung des Nutzbands, Dirichlet-Kernel) und Nullstellen bei $kf_{S,1}$ (Dämpfung der Images) hervorruft.



(a) (b)

Abb. 3: Images bei Upsampling mit Nullenstopfen und ZOH ($L = 3$)

Zum numerischen Vergleich der verschiedenen Interpolationsverfahren wird untersucht, welche Performance ein solches System mit folgenden Randbedingungen erreichen kann:

- Eingangssamplerate: $f_{S,in} = 44.1$ oder 48 kHz
- Ausgangssamplerate: $f_{S,in} = 48$ kHz
- Oversamplingrate: $L = 256$

```
1 interp1d(x, y, kind='linear', axis=-1, copy=True, bounds_error=True, fill_value=np.nan,
      assume_sorted=False)
```

4.1.2 (Interpolated)UnivariateSpline

... berechnet eine eindimensionale Glättungs-Spline der Ordnung `k` zu den Eingangsdaten:

```
UnivariateSpline(x, y, w=None, bbox=[None, None], k=3, s=None)
```

Die **Bounding Box `bbox`** gibt den Bereich der Ursprungsdaten an, der für die Interpolation benutzt werden soll (Default: `bbox = [x[0],x[-1]]`) - wird versucht, außerhalb dieses Bereichs zu interpolieren, gibt es einen Fehler!

Die Daten werden mit dem Array **`w` gewichtet**, das daher die gleiche Länge haben muss wie die Daten.

Der **smoothing factor `s`** setzt einen Zielwert für den RMS - Interpolationsfehler an den ursprünglichen Datenpunkten `y`, also `s = RMS(y - yinterpolated)` . Da `s` mit dem Absolutwert von |y| skaliert, setzt man z.B. für einen RMS-Zielfehler von 1 % von `max|y|`

```
s = 0.01 * np.fabs(y).max())**2
f_ip = UnivariateSpline(x, y, s=s)
ynew = f_ip(xnew)
print('Knoten:'f_ip.get_knots(), '\n RMS:', f_ip.get_residual())
```

`get_residual()` gibt das erreichte `s` zurück und `get_knots()` die Knoten, mit zunehmendem Zielfehler `s` werden weniger Knoten bestimmt (Polynomabschnitte).

Mit der Funktion `InterpolatedUnivariateSpline` oder mit `s=0` *interpoliert* man die Daten: Die Interpolationsfunktion geht also durch die ursprünglichen Datenpunkte, mit `s=0` erhält man ungefähr `N = len(x)` Knoten.

4.2 pandas.timeseries

Schwerpunkt auf Datums / Zeitkonvertierung

Uri Nieto Audio Resampling in Python, http://urinieto.com/2011/05/audio-resampling-in-python/

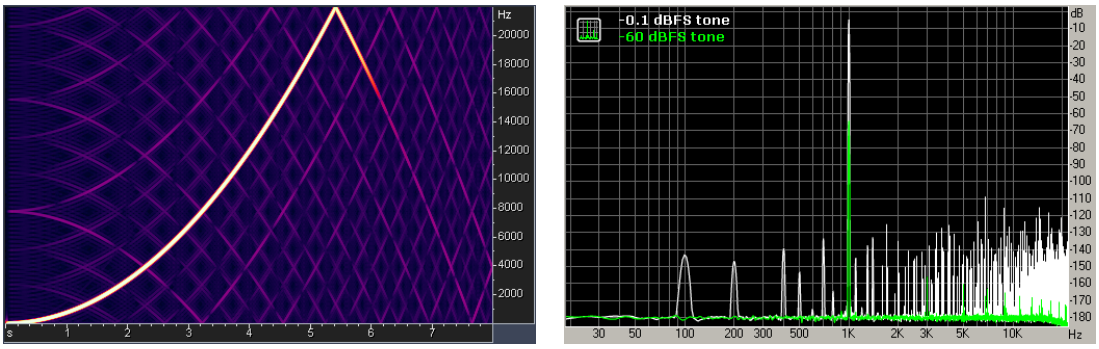
5 Bewertung der Qualität

Übliche Maßzahlen zur Bewertung der Qualität eines Abtastratenwandlers sind:

Signal-to-Noise Ratio (SNR): Eine Maßzahl dafür, wieviel Störungen der Umwandlungsprozess dem Signal hinzufügt (gemessen in dB).

Bandbreite: Tiefpassfilterung ist ein unvermeidlicher Bestandteil jeder Ratenwandlung; abhängig von der Ordnung des verwendeten Filters wird dabei das Signalband mehr oder weniger stark beschnitten.

Geschwindigkeit / Rechenaufwand: Vor allem für Real-Time Anwendungen ist ein minimaler Rechenaufwand essenziell.



(a) (b)

Abb. 4: Performance eines Sample Rate Converter bei der Umwandlung 96 kHz -> 44.1 kHz, (a) sine chirp, (b) 1 kHz sine

Bei den meisten Verfahren zur Abtastratenwandlung hängt die Qualität ab von der Frequenz des Eingangsssignals und dem Verhältnis zwischen Eingangs- und Ausgangsabtastrate.

Das Infinite Wave Masteringstudio hat unter http://src.infinitewave.ca/ einen sehr umfassenden Vergleich verschiedener ASRC Softwareimplementierungen in Betriebssystemen und Audio-Software bereit gestellt. Unter Help finden sich viele nützliche Informationen. Die Spektrogramme wurden erstellt mit den folgenden Einstellungen: FFT Länge: 2048 (ohne zero padding), Time overlap: 8x, Window: Kaiser (beta = 20), Amplitudenbereich: -180 dB bis 0 dB (Danke an Dave Horrocks von Infinite Wave für die Auskunft!).

Verzerrungen und Aliasing: Mit Hilfe eines Sweeps zwischen 0 und 48 kHz sieht man schnell, welches Maß an Verzerrungen und Aliasing bei welchen Eingangsfrequenzen auftritt. Mit einem 1 kHz Testtons bei -0.1 dBFS und -60 dBFS sieht man detailliert, bei welchen Frequenzen Verzerrungen anfallen und wo Quantisierungseffekte auftreten.

Frequenzgang im Pass- und Stoppband: Eine ungenügende Dämpfung im Stoppband führt zu Aliasing, Variationen im Passband beeinflussen das Nutzsignal.

Impulsantwort: Der Impulsantwort kann man entnehmen, wie steilbandig das Filter ist (starkes Ringing), ob das Filter linearphasig ist (konstante Gruppenlaufzeit) oder minimalphasig (geringes Pre-Ringing). Nicht-linearphasige Filter können die Lokalisierbarkeit von Audioquellen beeinträchtigen, minimalphasige Filter haben dafür weniger Einschwingen / Klingeln vor dem eigentlichen Audio-Event (z.B. Snare-Schlag).

Phasengang / Gruppenlaufzeit: Aus dem Phasengang kann man ähnliche Ergebnisse ableiten wie aus der Impulsantwort.

SNR wurde hier nicht verglichen, diese Maßzahl ist für sich auch nicht besonders aussagekräftig.

Die Bestimmung des SNRs der von mir in Python modellierten Wandler machte es notwendig, mich im Detail mit FFT basierter Spektralanalyse, und der Fensterung von Signalen zu beschäftigen. [HRS02] gibt zu beiden Themen eine gute Übersicht. Das Paper von Fredric Harris unter [Har78] ist das Standardwerk zum Einsatz von Fenstern. Unter [JH07] findet sich eine ausführliche Erklärung des Zusammenhangs zwischen Leistungsdichte-Spektrum und

Leistungs-Spektrum im Spectrum-Scope Block von Simulink, und in [Nar98] wird die Fourier-Transformierte einer Funktion durch die DFT approximiert.

6 Literatur

6.1 Interpolatoren

Zum Entwurf von Lagrange Interpolatoren erfährt man etwas in [SR73] und [Ye03]. Eine gute Übersicht über B-Splines gibt Michael Unser in seinen Papers, z.B. zum Einstieg in das Thema in [Uns99], oder in [UAE93a] und [UAE93b]. Der Zusammenhang zwischen kardinalen B-Splines und deren Konvergenz gegen den idealen bandbegrenzten Interpolator wird in [AA92] behandelt.

6.2 Implementierung von Interpolatoren

Zur Implementierung eines Interpolators auf einem FPGA ist eine Abbildung der Theorie auf eine effiziente Hardware-Struktur nötig. In seinem Buch behandelt Fredric Harris einige Aspekte der Implementierung von Polyphasen- und Farrow-Filtern in Hardware [Har04]. In [**?**] entwickelt Steven Smith eine an der Praxis orientierte Methode, ideale bandbegrenzte Interpolation zu approximieren und mit Lookup-Tables zu implementieren. [Far88] ist das berühmte Paper von C. Farrow, in dem er das Farrow-Filter zur effizienten Implementierung von polynomialen Filtern vorstellt. In [NCD09] wird ein Fractional-Delay-Filter auf einem FPGA in einer aufwandsarmen Struktur realisiert. In [UAE93b] behandelt Michael Unser die Implementierung von B-Spline Interpolatoren. In [FLL+93a] und [FLL+93b] wird eine Struktur von Lagrange Interpolatoren vorgestellt, die ohne Multiplizierer auskommt.

6.3 Anwendungen von Interpolatoren

Interpolatoren kommen nicht nur in Abtastratenwandlern zum Einsatz, sondern auch in der Bildverarbeitung [UTY95], und Medizintechnik [LGS01]. Sie werden auch dazu verwendet, um mit einer kontinuierlichen Form eines Signals zu rechnen [DB]. In [Nie01] findet sich ein sehr interessanter Vergleich verschiedener Polynom-Interpolatoren zum Resampling von überabgetasteten (oversampled) Audio Signalen.

6.4 Entwurf von Abtastratenwandlern

Ein Interpolator ist ein Teil-System eines Abtastratenwandlers. Er ist im einfachsten Falle ein Zero-Order-Hold. In [Lok05] gibt Ivar Løkken einen einfachen Einstieg in den Entwurf von mehrstufigen Wandlern mit Zero-Order-Hold, und in Grundzügen auch, mit First-Order-Hold Interpolatoren. In [CR81] geht es um Ähnliches. Unter [Inf14] finden sich interessante Tests verschiedener Wandler.

Literatur

[AA92] M. E. A. Aldroubi, M. Unser, *Cardinal spline filters: Stability and convergence to ideal sinc interpolator*, Signal Processing **28** (1992), 127–138.

[CR81] R. E. Crochiere and L. R. Rabiner, *Interpolation and decimation of digital signals—a tutorial review*, Proceedings of the IEEE **69** (1981), no. 3, 300–331.

[DB] J. V. D. Babic, *Polynomial-based interpolation filters for dsp applications*, Lecture Notes, http://www.cs.tut.fi/kurssit/TLT-5806/Interpol.pdf.

[Eva00] G. Evangelista, *Zum Entwurf digitaler Systeme zur asynchronen Abtastratenumsetzung*, Ph.D. thesis, Ruhr-Universität Bochum, Dez. 2000.

[Eva03a] _____, *Design of Digital Systems for Arbitrary Sampling Rate Conversion*, EURASIP J. Signal Processing **83** (2003), no. 2, 377–387.

[Eva03b] _____, *Roundoff noise analysis in digital systems for arbitrary sampling rate conversion*, Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on **50** (2003), no. 12, 1016–1023.

[Far88] C. W. Farrow, *A continuously variable digital delay element*, Proc. IEEE Int Circuits and Systems Symp, 1988, pp. 2641–2645.

[FLL+93a] F. Francesconi, G. Lazzari, V. Liberali, F. Maloberti, and G. Torelli, *Multiplier-free lagrange interpolators for oversampled d/a converters*, Proc. IEEE Int Circuits and Systems, ISCAS '93 Symp, 1993, pp. 219–222.

[FLL+93b] _____, *A novel interpolator architecture for sigma-delta dacs*, Proc. [4th] European Conf. with the European Event in ASIC Design Design Automation, 1993, pp. 249–253.

[Fom00] S. Fomel, *On the general theory of data interpolation*, Stanford Exploration Project **Report SERGEY** (2000), 101–116, http://sepwww.stanford.edu/oldsep/sergey/sepsergey/genint.pdf.

[Har78] F. J. Harris, *On the use of windows for harmonic analysis with the discrete fourier transform*, Proceedings of the IEEE **66** (1978), no. 1, 51–83.

[Har04] F. J. Harris, *Multirate signal processing for communication systems*, Prentice Hall, 2004.

[HRS02] G. Heinzel, A. Rüdiger, and R. Schilling, *Spectrum and spectral density estimation by the Discrete Fourier transform (DFT), including a comprehensive list of window functions and some new flat-top windows*, Max-Planck-Institut für Gravitationsphysik (2002), http://www.rssd.esa.int/SP/LISAPATHFINDER/docs/Data_Analysis/GH_FFT.pdf.

[Inf14] Infinite Wave Mastering Studio, *Test of Sampling Rate Converters*, http://src.infinitewave.ca/, 2014.

[JH07] F. Q. Josef Hoffmann, *Signalverarbeitung mit matlab und simulink*, Oldenbourg, 2007, Abschnitt 7.4 Spektrale Leistungsdichte und Power Spectrum.

[KG04] R. Kappeler and D. Grünert, *Sample Rate Converter - 192 kHz Stereo Sample Rate Conversion with B-Spline Interpolation*, Mar. 2004.

[LBR04] V. Lehtinen, D. Babic, and M. Renfors, *Comparison of continuous-and discrete-time modelling of polynomial-based interpolation filters*, Proc. 6th Nordic Signal Processing Symp. NORSIG 2004, 2004, pp. 49–52.

[LGS01] T. M. Lehmann, C. Gonner, and K. Spitzer, *Addendum: B-spline interpolation in medical image processing*, IEEE Transactions on Medical Imaging **20** (2001), no. 7, 660–665.

[Lok05] I. Lokken, *The ups and downs of arbitrary sample rate conversion*, April 2005.

[Mei02] E. Meijering, *A chronology of interpolation: from ancient astronomy to modern signal and image processing*, Proceedings of the IEEE **90** (2002), no. 3, 319–342.

[Nar98] F. J. Narcowich, *The dft approximation to the fourier transform*, Department of Mathematics (1998), http://www.math.tamu.edu/~fnarc/psfiles/fft_ft.ps.

[NCD09] U. Nithirochananont, S. Chivapreecha, and K. Dejhan, *An fpga-based implementation of variable fractional delay filter*, Proc. 5th Int. Colloquium Signal Processing & Its Applications CSPA 2009, 2009, pp. 104–107.

[Nie01] O. Niemitalo, *Polynomial interpolators for high-quality resampling of oversampled audio*, October 2001.

[Pin] A. Pinkus, *Weierstrass and approximation theory*, http://www.math.umanitoba.ca/HAT/fpapers/wap.pdf.

[Smi11] J. O. Smith, *Interpolated delay lines, ideal bandlimited interpolation, and fractional delay filter design*, Lecture Notes, February 2011, https://ccrma.stanford.edu/~jos/Interpolation/Interpolation.pdf.

[Smi14] _____, *Bandlimited interpolation - introduction and algorithm (*, 2014, https://ccrma.stanford.edu/~jos/resample/resample.html.

[SR73] R. W. Schafer and L. R. Rabiner, *A digital signal processing approach to interpolation*, Proceedings of the IEEE **61** (1973), no. 6, 692–702.

[UAE93a] M. Unser, A. Aldroubi, and M. Eden, *B-spline signal processing. i. theory*, IEEE Transactions on Signal Processing **41** (1993), no. 2, 821–833.

[UAE93b] _____, *B-spline signal processing. ii. efficiency design and applications*, IEEE Transactions on Signal Processing **41** (1993), no. 2, 834–848.

[Uns99] M. Unser, *Splines: a perfect fit for signal and image processing*, IEEE Signal Processing Magazine **16** (1999), no. 6, 22–38.

[UTY95] M. Unser, P. Thevenaz, and L. Yaroslavsky, *Convolution-based interpolation for fast, high-quality rotation of images*, IEEE Transactions on Image Processing **4** (1995), no. 10, 1371–1381.

[Ye03] Z. Ye, *Linear phase lagrange interpolation filter using odd number of basepoints*, Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP '03), vol. 6, 2003.