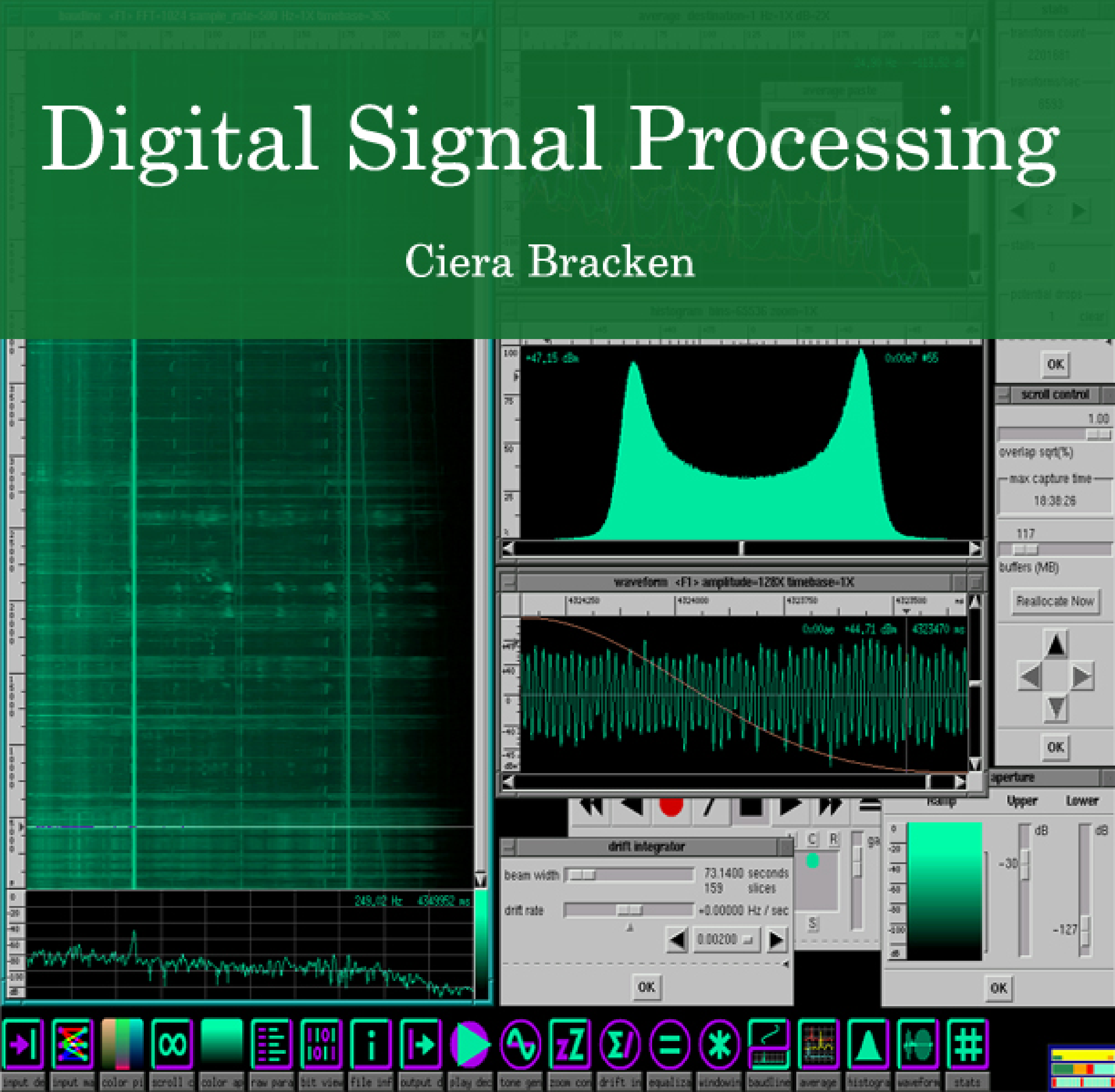
Digital Signal Processing

Ciera Bracken



First Edition, 2012

ISBN 978-81-323-3040-0

Published by:
Research World
4735/22 Prakashdeep Bldg,
Ansari Road, Darya Ganj,
Delhi - 110002
Email: info@wtbooks.com

Table of Contents

Introduction

Digital signal processing (**DSP**) is concerned with the representation of signals by a sequence of numbers or symbols and the processing of these signals. Digital signal processing and analog signal processing are subfields of signal processing. DSP includes subfields like: audio and speech signal processing, sonar and radar signal processing, sensor array processing, spectral estimation, statistical signal processing, digital image processing, signal processing for communications, control of systems, biomedical signal processing, seismic data processing, etc.

The goal of DSP is usually to measure, filter and/or compress continuous real-world analog signals. The first step is usually to convert the signal from an analog to a digital form, by *sampling* it using an analog-to-digital converter (ADC), which turns the analog signal into a stream of numbers. However, often, the required output signal is another analog output signal, which requires a digital-to-analog converter (DAC). Even if this process is more complex than analog processing and has a discrete value range, the application of computational power to digital signal processing allows for many advantages over analog processing in many applications, such as error detection and correction in transmission as well as data compression.

DSP algorithms have long been run on standard computers, on specialized processors called digital signal processors (DSPs), or on purpose-built hardware such as application-specific integrated circuit (ASICs). Today there are additional technologies used for digital signal processing including more powerful general purpose microprocessors, field-programmable gate arrays (FPGAs), digital signal controllers (mostly for industrial apps such as motor control), and stream processors, among others.

Signal sampling

With the increasing use of computers the usage of and need for digital signal processing has increased. In order to use an analog signal on a computer it must be digitized with an analog-to-digital converter. Sampling is usually carried out in two stages, discretization and quantization. In the discretization stage, the space of signals is partitioned into equivalence classes and quantization is carried out by replacing the signal with representative signal of the corresponding equivalence class. In the quantization stage the representative signal values are approximated by values from a finite set.

The Nyquist–Shannon sampling theorem states that a signal can be exactly reconstructed from its samples if the sampling frequency is greater than twice the highest frequency of the signal; but requires an infinite number of samples . In practice, the sampling frequency is often significantly more than twice that required by the signal's limited bandwidth.

A digital-to-analog converter is used to convert the digital signal back to analog. The use of a digital computer is a key ingredient in digital control systems.

DSP domains

In DSP, engineers usually study digital signals in one of the following domains: time domain (one-dimensional signals), spatial domain (multidimensional signals), frequency domain, autocorrelation domain, and wavelet domains. They choose the domain in which to process a signal by making an informed guess (or by trying different possibilities) as to which domain best represents the essential characteristics of the signal. A sequence of samples from a measuring device produces a time or spatial domain representation, whereas a discrete Fourier transform produces the frequency domain information, that is the frequency spectrum. Autocorrelation is defined as the cross-correlation of the signal with itself over varying intervals of time or space.

Time and space domains

The most common processing approach in the time or space domain is enhancement of the input signal through a method called filtering. Digital filtering generally consists of some linear transformation of a number of surrounding samples around the current sample of the input or output signal. There are various ways to characterize filters; for example:

- A "linear" filter is a linear transformation of input samples; other filters are "non-linear". Linear filters satisfy the superposition condition, i.e. if an input is a weighted linear combination of different signals, the output is an equally weighted linear combination of the corresponding output signals.

- A "causal" filter uses only previous samples of the input or output signals; while a "non-causal" filter uses future input samples. A non-causal filter can usually be changed into a causal filter by adding a delay to it.

- A "time-invariant" filter has constant properties over time; other filters such as adaptive filters change in time.

- Some filters are "stable", others are "unstable". A stable filter produces an output that converges to a constant value with time, or remains bounded within a finite interval. An unstable filter can produce an output that grows without bounds, with bounded or even zero input.

- A "finite impulse response" (FIR) filter uses only the input signals, while an "infinite impulse response" filter (IIR) uses both the input signal and previous samples of the output signal. FIR filters are always stable, while IIR filters may be unstable.

Filters can be represented by block diagrams which can then be used to derive a sample processing algorithm to implement the filter using hardware instructions. A filter may also be described as a difference equation, a collection of zeroes and poles or, if it is an FIR filter, an impulse response or step response.

The output of a digital filter to any given input may be calculated by convolving the input signal with the impulse response.

Frequency domain

Signals are converted from time or space domain to the frequency domain usually through the Fourier transform. The Fourier transform converts the signal information to a magnitude and phase component of each frequency. Often the Fourier transform is converted to the power spectrum, which is the magnitude of each frequency component squared.

The most common purpose for analysis of signals in the frequency domain is analysis of signal properties. The engineer can study the spectrum to determine which frequencies are present in the input signal and which are missing.

In addition to frequency information, phase information is often needed. This can be obtained from the Fourier transform. With some applications, how the phase varies with frequency can be a significant consideration.

Filtering, particularly in non-realtime work can also be achieved by converting to the frequency domain, applying the filter and then converting back to the time domain. This is a fast, O(n log n) operation, and can give essentially any filter shape including excellent approximations to brickwall filters.

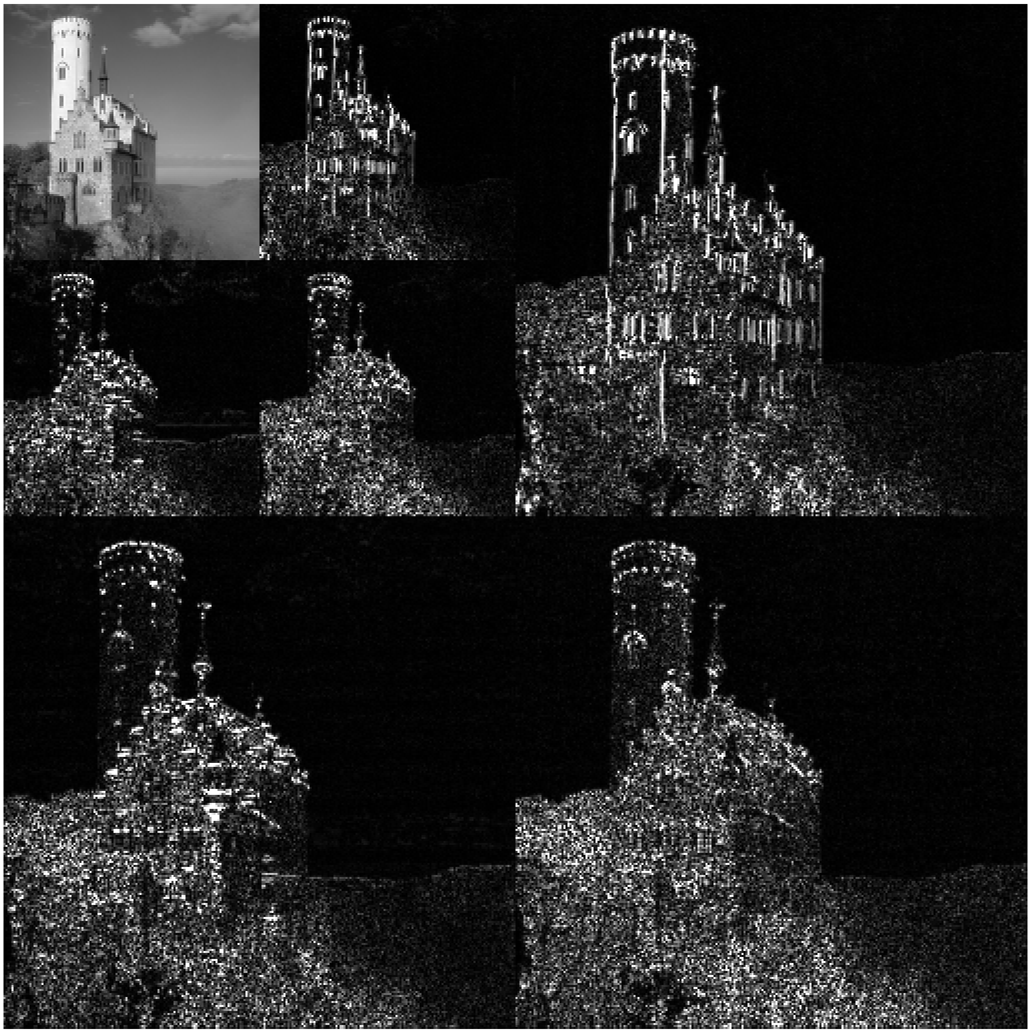
There are some commonly used frequency domain transformations. For example, the cepstrum converts a signal to the frequency domain through Fourier transform, takes the logarithm, then applies another Fourier transform. This emphasizes the frequency components with smaller magnitude while retaining the order of magnitudes of frequency components.

Frequency domain analysis is also called *spectrum-* or *spectral analysis*.

Z-plane analysis

Whereas analog filters are usually analysed in terms of transfer functions in the s plane using Laplace transforms, digital filters are analysed in the z plane in terms of Z-transforms. A digital filter may be described in the z plane by its characteristic collection of zeroes and poles.

Wavelet



An example of the 2D discrete wavelet transform that is used in JPEG2000. The original image is high-pass filtered, yielding the three large images, each describing local changes in brightness (details) in the original image. It is then low-pass filtered and downscaled, yielding an approximation image; this image is high-pass filtered to produce the three smaller detail images, and low-pass filtered to produce the final approximation image in the upper-left.

In numerical analysis and functional analysis, a **discrete wavelet transform** (DWT) is any wavelet transform for which the wavelets are discretely sampled. As with other wavelet transforms, a key advantage it has over Fourier transforms is temporal resolution: it captures both frequency *and* location information (location in time).

Applications

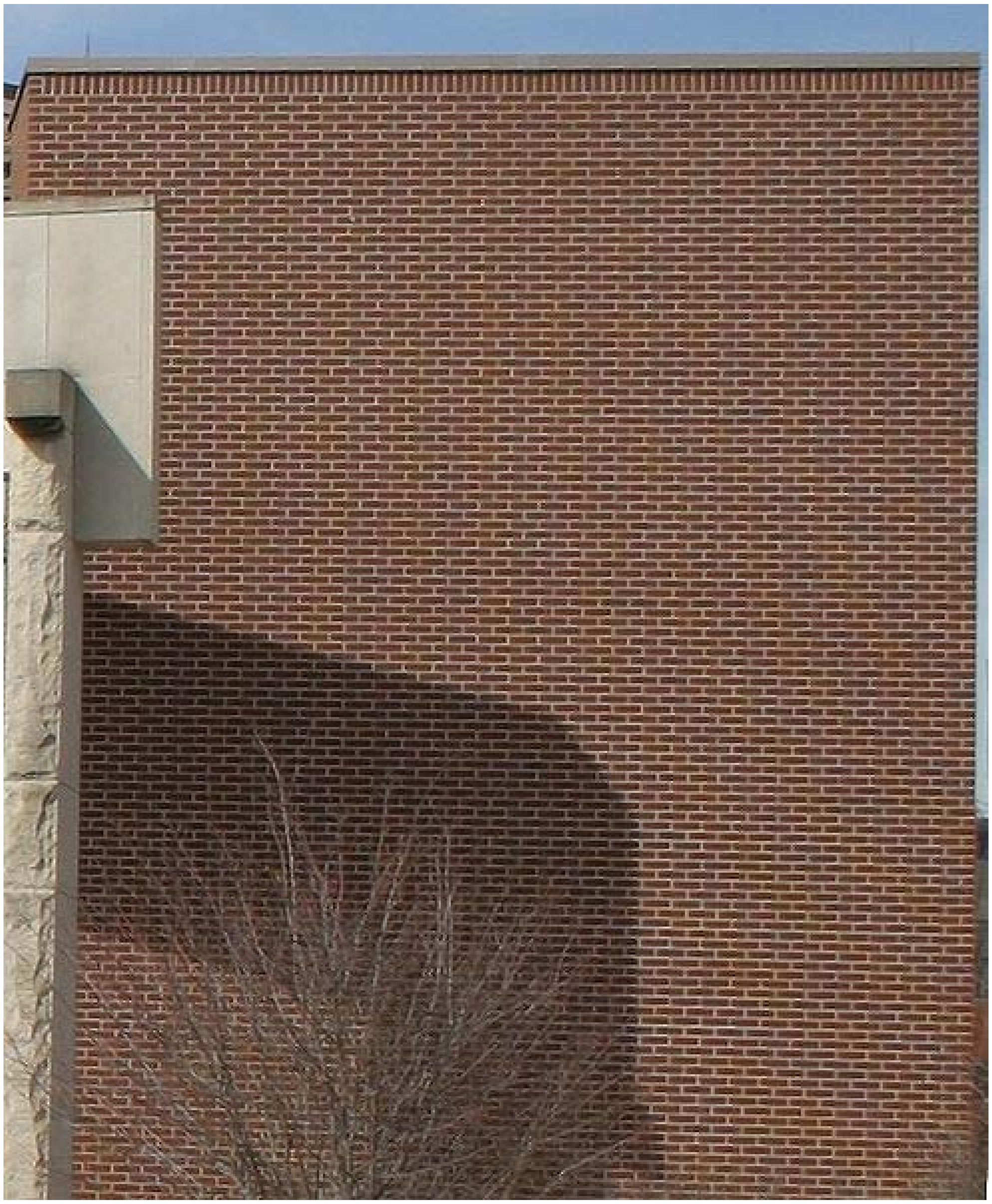
The main applications of DSP are audio signal processing, audio compression, digital image processing, video compression, speech processing, speech recognition, digital communications, RADAR, SONAR, seismology and biomedicine. Specific examples are speech compression and transmission in digital mobile phones, room correction of sound in hi-fi and sound reinforcement applications, weather forecasting, economic forecasting, seismic data processing, analysis and control of industrial processes, medical imaging such as CAT scans and MRI, MP3 compression, computer graphics, image manipulation, hi-fi loudspeaker crossovers and equalization, and audio effects for use with electric guitar amplifiers.

Implementation

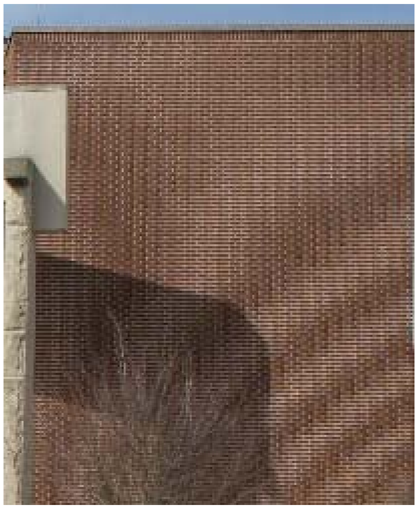
Digital signal processing is often implemented using specialised microprocessors such as the DSP56000, the TMS320, or the SHARC. These often process data using fixed-point arithmetic, although some versions are available which use floating point arithmetic and are more powerful. For faster applications FPGAs might be used. Beginning in 2007, multicore implementations of DSPs have started to emerge from companies including Freescale and Stream Processors, Inc. For faster applications with vast usage, ASICs might be designed specifically. For slow applications, a traditional slower processor such as a microcontroller may be adequate. Also a growing number of DSP applications are now being implemented on Embedded Systems using powerful PCs with a Multi-core processor.

Chapter 1

Aliasing



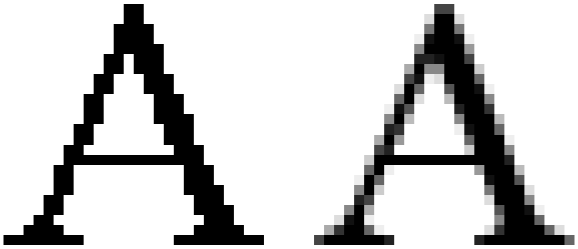
Properly sampled image of brick wall.



Spatial aliasing in the form of a Moiré pattern.

In signal processing and related disciplines, **aliasing** refers to an effect that causes different signals to become indistinguishable (or *aliases* of one another) when sampled. It also refers to the distortion or artifact that results when the signal reconstructed from samples is different from the original continuous signal.

Description



Aliasing example of the A letter in Times New Roman. Left: aliased image, right: *antialiased* image.

When a digital image is viewed, a reconstruction—also known as an interpolation—is performed by a display or printer device, and by the eyes and the brain. If the resolution is too low, the reconstructed image will differ from the original image, and an alias is seen. An example of **spatial aliasing** is the Moiré pattern one can observe in a poorly pixelized image of a brick wall. Techniques that avoid such poor pixelizations are called anti-aliasing. Aliasing can be caused either by the sampling stage or the reconstruction stage; these may be distinguished by calling sampling aliasing *prealiasing* and reconstruction aliasing *postaliasing.*

Temporal aliasing is a major concern in the sampling of video and audio signals. Music, for instance, may contain high-frequency components that are inaudible to humans. If a piece of music is sampled at 32000 samples per second (sps), any frequency components above 16000 Hz (the Nyquist frequency) will cause aliasing when the music is reproduced by a digital to analog converter (DAC). To prevent that, it is customary to remove components above the Nyquist frequency (with an anti-aliasing filter) prior to sampling. But any realistic filter or DAC will also affect (attenuate) the components just below the Nyquist frequency. Therefore, it is also customary to choose a higher Nyquist frequency by sampling faster (typically 44100 sps (CD), 48000 (professional audio), or 96000 (high definition audio)).

In video or cinematography, temporal aliasing results from the limited frame rate, and causes the wagon-wheel effect, whereby a spoked wheel appears to rotate too slowly or even backwards. Aliasing has changed its apparent frequency of rotation. A reversal of direction can be described as a negative frequency. Temporal aliasing frequencies in video and cinematography are determined by the frame rate of the camera, but the relative intensity of the aliased frequencies is determined by the shutter timing (exposure time) or the use of a temporal aliasing reduction filter during filming.

Like the video camera, most sampling schemes are periodic; that is they have a characteristic sampling frequency in time or in space. Digital cameras provide a certain number of samples (pixels) per degree or per radian, or samples per mm in the focal plane of the camera. Audio signals are sampled (digitized) with an analog-to-digital converter, which produces a constant number of samples per second. Some of the most dramatic and subtle examples of aliasing occur when the signal being sampled also has periodic content.

Bandlimited functions

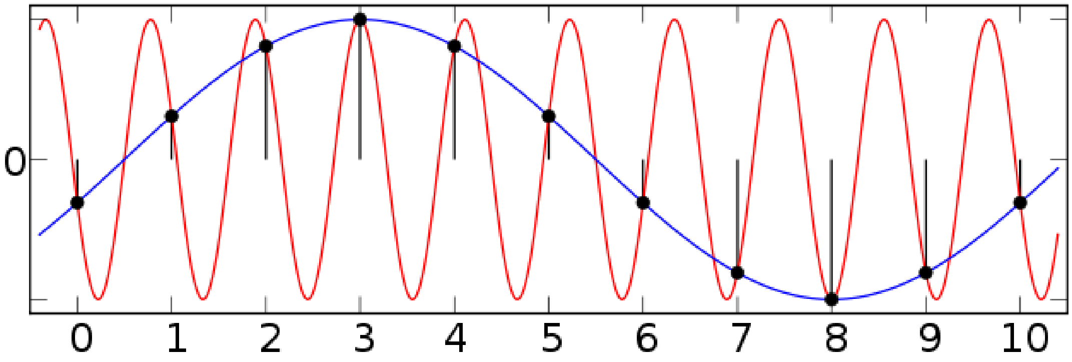
Actual signals have finite duration and their frequency content, as defined by the Fourier transform, has no upper bound. Some amount of aliasing always occurs when such functions are sampled. Functions whose frequency content is bounded (*bandlimited*) have infinite duration. If sampled at a high enough rate, determined by the *bandwidth*, the original function can in theory be perfectly reconstructed from the infinite set of samples.

Bandpass signals

Sometimes aliasing is used intentionally on signals with no low-frequency content, called *bandpass* signals. Undersampling, which creates low-frequency aliases, can produce the same result, with less effort, as frequency-shifting the signal to lower frequencies before sampling at the lower rate. Some digital channelizers exploit aliasing in this way for computational efficiency.

Sampling sinusoidal functions

Sinusoids are an important type of periodic function, because realistic signals are often modeled as the summation of many sinusoids of different frequencies and different amplitudes (with a Fourier series or transform). Understanding what aliasing does to the individual sinusoids is useful in understanding what happens to their sum.



Two different sinusoids that fit the same set of samples.

Here a plot depicts a set of samples whose sample-interval is 1.0, and two (of many) different sinusoids that could have produced the samples. The sample-rate in this case is $f_s = 1.0$. For instance, if the interval is 1 second, the rate is 1 sample per second. Nine cycles of the red sinusoid and 1 cycle of the blue sinusoid span an interval of 10. The respective sinusoid frequencies are f_{red}= 0.9 and f_{blue}= 0.1.

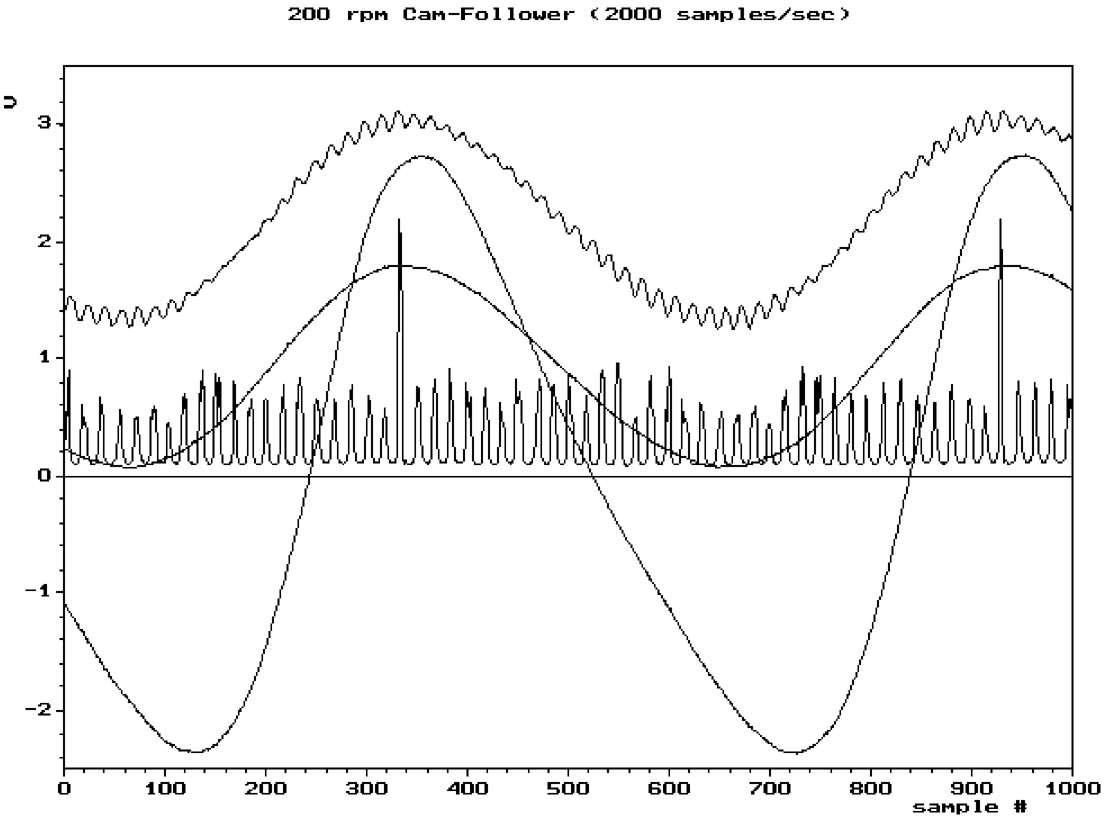
In general, when a sinusoid of frequency f is sampled with frequency f_s, the resulting samples are indistinguishable from those of another sinusoid of frequency $f_{\text{image}}(N) = |f - N f_s|$ for any integer N (with $f_{\text{image}}(0) = f$ being the actual signal frequency). Most reconstruction techniques produce the minimum of these frequencies, so it is often important that $f_{\text{image}}(0)$ be the unique minimum. A sufficient condition for that is $f_s/2 > f$, where $f_s/2$ is commonly called the Nyquist frequency of a system that samples at rate f_s.

In our graphic example, the Nyquist condition is satisfied if the original signal is the blue sinusoid ($f = f_{\text{blue}}$). But if $f = f_{\text{red}}$, the lowest image frequency is:

$$f_{\text{image}}(1) = |0.9 - 1.0| = 0.1 = f_{\text{blue}}.$$

- A reconstruction technique that constructs the lowest possible frequency from the samples will reproduce the blue sinusoid instead of the red one.
- We note that -0.1 is also an image frequency, but since there is no way to distinguish a sinusoid of frequency $-f$ from one of frequency f, all aliases can be described in terms of just positive frequencies.

Sample frequency



Aliasing animated gif - a graph of signals sampled at different rates, showing how the character of some signals changes dramatically when the rate is too low.

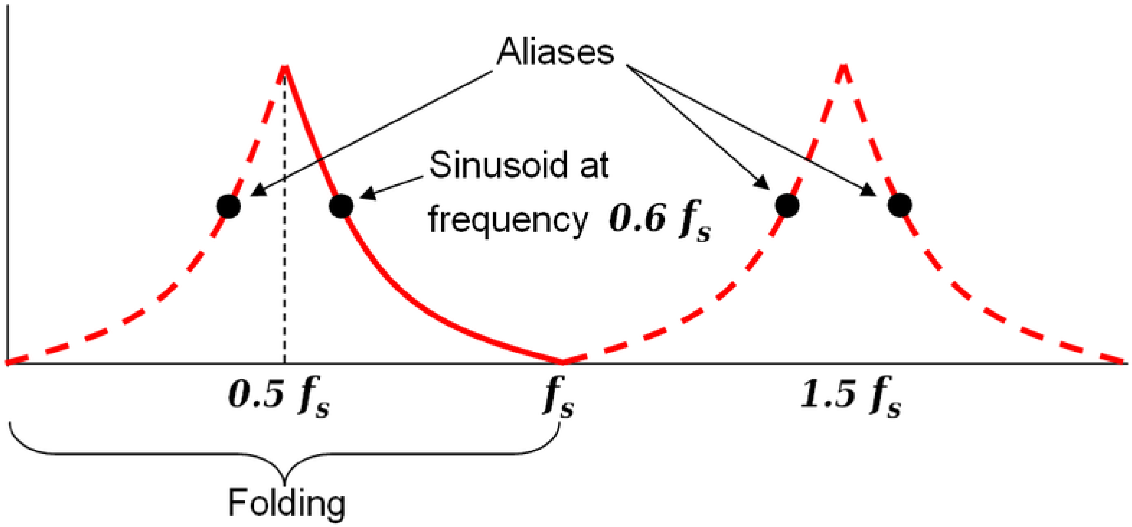
When the condition $f_s/2 > f$ is met for the highest frequency component of the original signal, then it is met for all the frequency components, a condition known as the Nyquist criterion. That is typically approximated by filtering the original signal to attenuate high frequency components before it is sampled. They still generate low-frequency aliases, but at very low amplitude levels, so as not to cause a problem. A filter chosen in anticipation of a certain sample frequency is called an anti-aliasing filter. The filtered signal can subsequently be reconstructed without significant additional distortion, for example by the Whittaker–Shannon interpolation formula.

The Nyquist criterion presumes that the frequency content of the signal being sampled has an upper bound. Implicit in that assumption is that the signal's duration has *no* upper bound. Similarly, the Whittaker–Shannon interpolation formula represents an interpolation filter with an unrealizable frequency response. These assumptions make up a mathematical model that is an idealized approximation, at best, to any realistic situation. The conclusion, that perfect reconstruction is possible, is mathematically correct for the model, but only an approximation for real samples of a real signal.

Complex signal representation

Complex signals are signals whose samples are complex numbers, and the concept of negative frequency is necessary for such signals. In that case, the frequencies of the aliases are given by just**:** $f_{\text{image}}(N) = f - Nf_s$.Therefore, as $f$increases from $f_s/2$to f_s,the image closest to 0 moves from $-f_s/2$up to 0.

Folding



The black dots are aliases of each other. The solid red line is an example of adjusting amplitude vs frequency. The dashed red lines are the corresponding paths of the aliases.

Real-valued sinusoids have the same negative-frequency aliases as complex ones. The absolute value operator, $f_{\text{image}}(N) = |f - Nf_s|$is introduced because there is always an equivalent sinusoid with a positive frequency. Therefore, as f increases from 0 to $f_s/2$, the image at frequency $f_{\text{image}}(1)$ moves from f_s to $f_s/2$. Similarly, as f increases from $f_s/2$ to f_s, $f_{\text{image}}(1)$ continues decreasing from $f_s/2$ to 0.

A graph of amplitude vs frequency for a single sinusoid at frequency $0.6f_s$ and some of its aliases at $0.4f_s$, $1.4f_s$, and $1.6f_s$ would look like the 4 black dots in the adjacent figure. The red lines depict the paths (loci) of the 4 dots if we were to adjust the frequency and amplitude of the sinusoid along the solid red segment (between $f_s/2$ and f_s). No matter what function we choose to change the amplitude vs frequency, the graph will exhibit symmetry between 0 and f_s. This symmetry is commonly referred to as **folding**, and another name for $f_s/2$ (the Nyquist frequency) is **folding frequency**. Folding is most often observed in practice when viewing the frequency spectrum of real-valued, sampled signals.

Historical usage

Historically the term *aliasing* evolved from radio engineering because of the action of superheterodyne receivers. When the receiver shifts multiple signals down to lower frequencies, from RF to IF by heterodyning, an unwanted signal, from an RF frequency equally far from the local oscillator (LO) frequency as the desired signal, but on the wrong side of the LO, can end up at the same IF frequency as the wanted one. If it is strong enough it can interfere with reception of the desired signal. This unwanted signal is known as an *image* or *alias* of the desired signal.

Angular aliasing

Aliasing occurs whenever the use of discrete elements to capture or produce a continuous signal causes frequency ambiguity.

Spatial aliasing, particular of angular frequency, can occur when reproducing a light field or sound field with discrete elements, as in 3D displays or wave field synthesis of sound.

This aliasing is visible in images such as posters with lenticular printing: if they have low angular resolution, then as one moves past them, say from left-to-right, the 2D image does not initially change (so it appears to move left), then as one moves to the next angular image, the image suddenly changes (so it jumps right) – and the frequency and amplitude of this side-to-side movement corresponds to the angular resolution of the image (and, for frequency, the speed of the viewer's lateral movement), which is the angular aliasing of the 4D light field.

The lack of parallax on viewer movement in 2D images and in 3-D film produced by stereoscopic glasses (in 3D films the effect is called "yawing", as the image appears to rotate on its axis) can similarly be seen as loss of angular resolution, all angular frequencies being aliased to 0 (constant).

More examples

Online "live" example

The qualitative effects of aliasing can be heard in the following audio demonstration. Six sawtooth waves are played in succession, with the first two sawtooths having a fundamental frequency of 440 Hz (A4), the second two having fundamental frequency of 880 Hz (A5), and the final two at 1760 Hz (A6). The sawtooths alternate between bandlimited (non-aliased) sawtooths and aliased sawtooths and the sampling rate is 22.05 kHz. The bandlimited sawtooths are synthesized from the sawtooth waveform's Fourier series such that no harmonics above the Nyquist frequency are present.

The aliasing distortion in the lower frequencies is increasingly obvious with higher fundamental frequencies, and while the bandlimited sawtooth is still clear at 1760 Hz, the aliased sawtooth is degraded and harsh with a buzzing audible at frequencies lower than the fundamental. Note that the audio file has been coded using Vorbis codec, and as such the audio is somewhat degraded.

Direction finding

A form of spatial aliasing can also occur in antenna arrays or microphone arrays used to estimate the direction of arrival of a wave signal, as in geophysical exploration by seismic waves. Waves must be sampled at more than two points per wavelength, or the wave arrival direction becomes ambiguous.

Chapter 2

All-Pass Filter & Anti-Aliasing Filter

All-Pass Filter

An **all-pass filter** is a signal processing filter that passes all frequencies equally, but changes the phase relationship between various frequencies. It does this by varying its propagation delay with frequency. Generally, the filter is described by the frequency at which the phase shift crosses 90° (i.e., when the input and output signals go into quadrature — when there is a quarter wavelength of delay between them).

They are generally used to compensate for other undesired phase shifts that arise in the system, or for mixing with an unshifted version of the original to implement a notch comb filter.

They may also be used to convert a mixed phase filter into a minimum phase filter with an equivalent magnitude response or an unstable filter into a stable filter with an equivalent magnitude response.

Active analog implementation

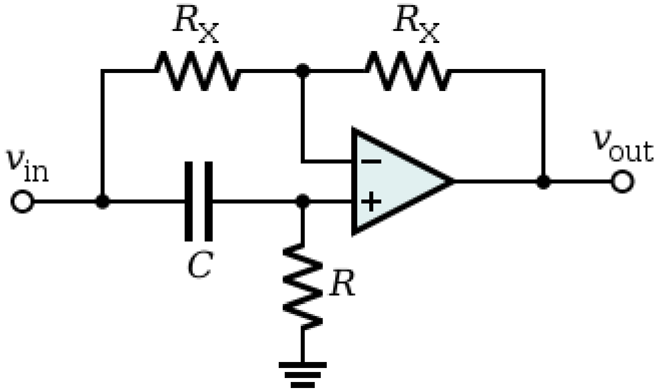


Figure 1: Schematic of an op amp all-pass filter

The operational amplifier circuit shown in Figure 1 implements an active all-pass filter with the transfer function

$$H(s) \triangleq \frac{sRC-1}{sRC+1},$$

which has one pole at -1/RC and one zero at 1/RC (i.e., they are *reflections* of each other across the imaginary axis of the complex plane). The magnitude and phase of H(iω) for some angular frequency ω are

$$|H(i\omega)| = 1 \quad \text{and} \quad \angle H(i\omega) = 180^\circ - 2\arctan(\omega RC).$$

As expected, the filter has unity-gain magnitude for all ω. The filter introduces a different delay at each frequency and reaches input-to-output *quadrature* at ω=1/RC (i.e., phase shift is 90 degrees).

This implementation uses a high-pass filter at the non-inverting input to generate the phase shift and negative feedback to compensate for the filter's attenuation.

- At high frequencies, the capacitor is a short circuit, thereby creating a unity-gain voltage buffer (i.e., no phase shift).
- At low frequencies and DC, the capacitor is an open circuit and the circuit is an inverting amplifier (i.e., 180 degree phase shift) with unity gain.
- At the corner frequency ω=1/RC of the high-pass filter (i.e., when input frequency is 1/(2πRC)), the circuit introduces a 90 degree shift (i.e., output is in quadrature with input; it is delayed by a quarter wavelength).

In fact, the phase shift of the all-pass filter is double the phase shift of the high-pass filter at its non-inverting input.

Implementation using low-pass filter

A similar all-pass filter can be implemented by interchanging the position of the resistor and capacitor, which turns the high-pass filter into a low-pass filter. The result is a phase shifter with the same quadrature frequency but a 180 degree shift at high frequencies and no shift at low frequencies. In other words, the transfer function is negated, and so it has the same pole at -1/RC and reflected zero at 1/RC. Again, the phase shift of the all-pass filter is double the phase shift of the first-order filter at its non-inverting input.

Voltage controlled implementation

The resistor can be replaced with a FET in its *ohmic mode* to implement a voltage-controlled phase shifter; the voltage on the gate adjusts the phase shift. In electronic music, a phaser typically consists of four or six of these phase-shifting sections connected in tandem and summed with the original. A low-frequency oscillator (LFO) ramps the control voltage to produce the characteristic swooshing sound.

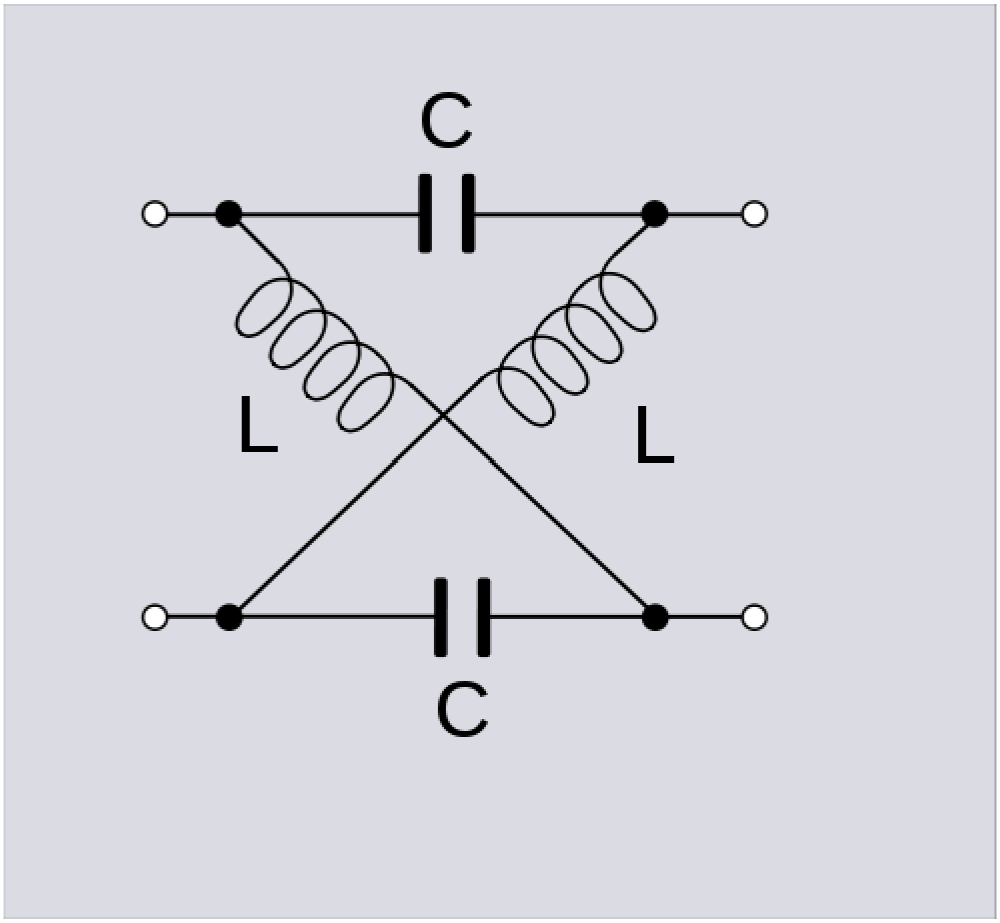
General usage

These circuits are used as phase shifters and in systems of phase shaping and time delay. Filters such as the above can be cascaded with unstable or mixed-phase filters to create a stable or minimum-phase filter without changing the magnitude response of the system. For example, by proper choice of pole (and therefore zero), a pole of an unstable system that is in the right-hand plane can be canceled and reflected on the left-hand plane.

Passive analog implementation

The benefit to implementing all-pass filters with active components like operational amplifiers is that they do not require inductors, which are bulky and costly in integrated circuit designs. In other applications where inductors are readily available, all-pass filters can be implemented entirely without active components. There are a number of circuit topologies that can be used for this. The following are the most commonly used circuits.

Lattice filter



An all-pass filter using lattice topology

The **lattice phase equaliser**, or **filter**, is a filter composed of lattice, or X-sections. With single element branches it can produce a phase shift up to 180°, and with resonant branches it can produce phase shifts up to 360°. The filter is an example of a constant-resistance network (i.e., its image impedance is constant over all frequencies).

T-section filter

The phase equaliser based on T topology is the unbalanced equivalent of the lattice filter and has the same phase response. While the circuit diagram may look like a low pass filter it is different in that the two inductor branches are mutually coupled. This results in transformer action between the two inductors and an all-pass response even at high frequency.

Bridged T-section filter

The bridged T topology is used for delay equalisation, particularly the differential delay between two landlines being used for stereophonic sound broadcasts. This application requires that the filter has a linear phase response with frequency (i.e., constant group delay) over a wide bandwidth and is the reason for choosing this topology.

Digital Implementation

A Z-transform implementation of an all-pass filter with a complex pole at z_0 is

$$H(z) = \frac{z^{-1} - z_0^*}{1 - z_0 z^{-1}}$$

which has a zero at $1/z_0^*$, where * denotes the complex conjugate. The pole and zero sit at the same angle but have reciprocal magnitudes (i.e., they are *reflections* of each other across the boundary of the complex unit circle). The placement of this pole-zero pair for a given z_0 can be rotated in the complex plane by any angle and retain its all-pass magnitude characteristic. Complex pole-zero pairs in all-pass filters help control the frequency where phase shifts occur.

To create an all-pass implementation with real coefficients, the complex all-pass filter can be cascaded with an all-pass that substitutes $z_0^*$for z_0, leading to the Z-transform implementation

$$H(z) = \frac{z^{-1} - z_0^*}{1 - z_0 z^{-1}} \times \frac{z^{-1} - z_0}{1 - z_0^* z^{-1}} = \frac{z^{-2} - 2\Re(z_0) z^{-1} + |z_0|^2}{1 - 2\Re(z_0) z^{-1} + |z_0|^2 z^{-2}},$$

which is equivalent to the difference equation

$$y[k] - 2\Re(z_0) y[k-1] + |z_0|^2 y[k-2] = x[k-2] - 2\Re(z_0) x[k-1] + |z_0|^2 x[k],$$

where *y*[*k*] is the output and *x*[*k*] is the input at discrete time step *k*.

Filters such as the above can be cascaded with unstable or mixed-phase filters to create a stable or minimum-phase filter without changing the magnitude response of the system. For example, by proper choice of z_0, a pole of an unstable system that is outside of the unit circle can be canceled and reflected inside the unit circle.

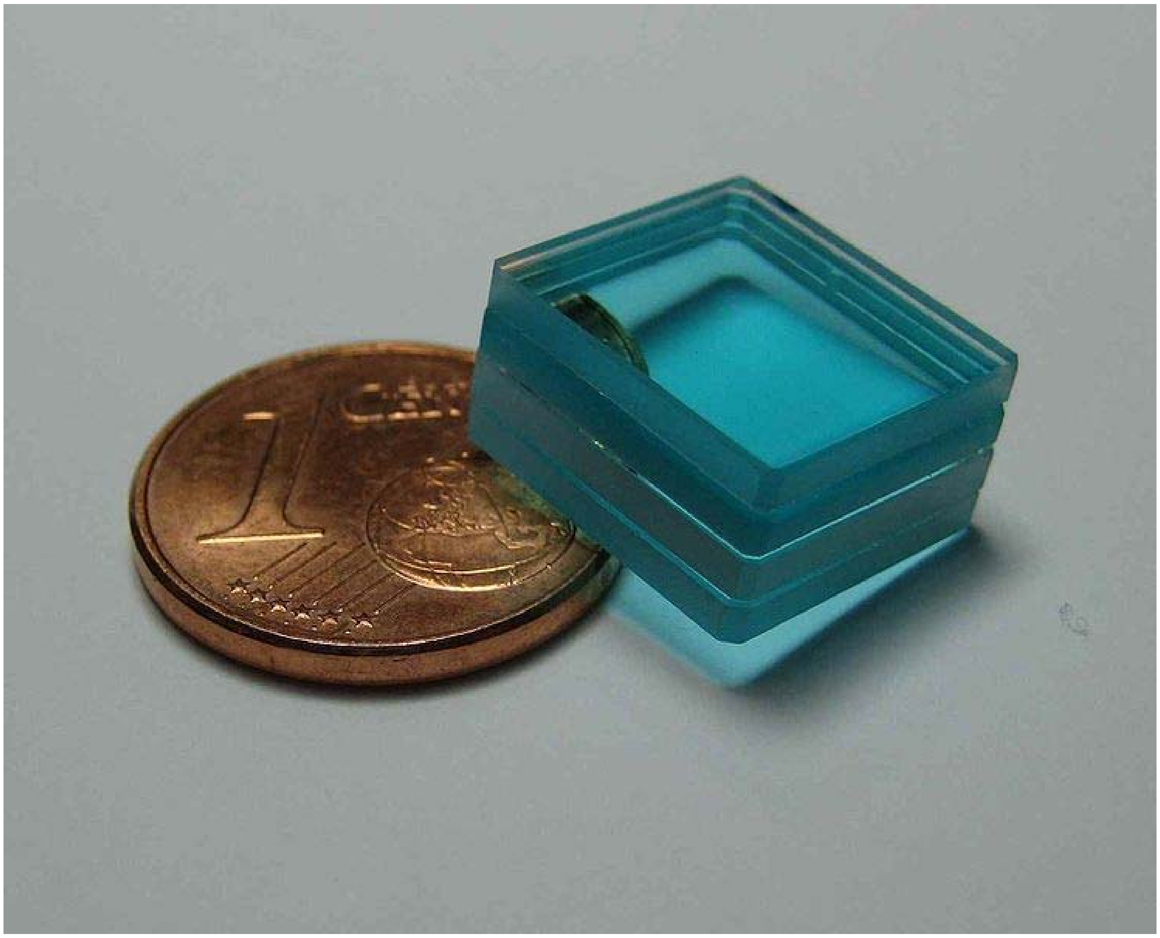
Anti-Aliasing Filter

An **anti-aliasing filter** is a filter used before a signal sampler, to restrict the bandwidth of a signal to approximately satisfy the sampling theorem. Since the theorem states that unambiguous interpretation of the signal from its samples is possible when the power of frequencies above the Nyquist frequency is zero, a real anti-aliasing filter can generally not completely satisfy the theorem. A realizable anti-aliasing filter will typically permit some aliasing to occur; the amount of aliasing that does occur depends on how good the filter is and what the frequency content of the input signal is.

Anti-aliasing filters are commonly used at the input of digital signal processing systems, for example in sound digitization systems; similar filters are used as reconstruction filters at the output of such systems, for example in music players. In the later case, the filter is to prevent aliasing in the conversion of samples back to a continuous signal, where again perfect stop-band rejection would be required to guarantee zero aliasing.

The theoretical impossibility of realizing perfect filters is not much an impediment in practice, though practical considerations do lead to system design choices such as oversampling to make it easier to realize "good enough" anti-aliasing filters.

Optical anti-aliasing filter



Anti-aliasing and IR-reject filter from an older digital video camera

In the case of optical image sampling, as by image sensors in digital cameras, the anti-aliasing filter is also known as an *optical lowpass filter* or *blur filter* or *AA filter*. The mathematics of sampling in two spatial dimensions is similar to the mathematics of time-domain sampling, but the filter implementation technologies are different. The typical implementation in digital cameras is two layers of birefringent material such as lithium niobate, which spreads each optical point into a cluster of four points.

The choice of spot separation for such a filter involves a tradeoff among sharpness, aliasing, and fill factor. In a monochrome or three-CCD or Foveon X3 camera, the fill factor alone, if near 100% effective with microlenses, can provide a significant anti-aliasing effect, while in color filter array (CFA, e.g. Bayer filter) cameras, an additional filter is generally needed to reduce aliasing to an acceptable level.

Applicability of oversampling

A technique known as oversampling is commonly used in audio conversion, especially audio output. The idea is to use a higher intermediate digital sample rate, so that a nearly-ideal digital filter can sharply cut off aliasing near the original low Nyquist frequency, while a much simpler analog filter can stop frequencies above the new higher Nyquist frequency.

The purpose of oversampling is to relax the requirements on the anti-aliasing filter, or to further reduce the aliasing. Since the initial anti-aliasing filter is analog, oversampling allows for the filter to be cheaper because the requirements are not as stringent, and also allows the anti-aliasing filter to have a smoother frequency response, and thus a less complex phase response.

On input, an initial analog anti-aliasing filter is relaxed, the signal is sampled at a high rate, and then downsampled using a nearly ideal digital anti-aliasing filter.

Bandpass signals

Often, an anti-aliasing filter is a low-pass filter; however, this is not a requirement. Generalizations of the Nyquist–Shannon sampling theorem allow sampling of other band-limited passband signals instead of baseband signals.

For signals that are bandwidth limited, but not centered at zero, a band-pass filter can be used as an anti-aliasing filter. For example, this could be done with a single-sideband modulated or frequency modulated signal. If one desired to sample an FM radio broadcast centered at 87.9 MHz and bandlimited to a 200 kHz band, then an appropriate anti-alias filter would be centered on 87.9 MHz with 200 kHz bandwidth (or pass-band of 87.8 MHz to 88.0 MHz), and the sampling rate would be no less than 400 kHz, but should also satisfy other constraints to prevent aliasing.

Signal overload

It is very important to avoid input signal overload when using an anti-aliasing filter. If the signal is strong enough, it can cause clipping at the analog-to-digital converter, even after filtering. When distortion due to clipping occurs after the anti-aliasing filter, it can create components outside the passband of the anti-aliasing filter; these components can then alias, causing the reproduction of other non-harmonically-related frequencies. In digital audio, the resulting aliased distorted signal of "digital clipping" has a characteristic sound that can be easily recognized.

Chapter 3

Audio Timescale-Pitch Modification

Time stretching is the process of changing the speed or duration of an audio signal without affecting its pitch. **Pitch scaling** or **pitch shifting** is the opposite: the process of changing the pitch without affecting the speed. There are also more advanced methods used to change speed, pitch, or both at once, as a function of time.

These processes are used, for instance, to match the pitches and tempos of two pre-recorded clips for mixing when the clips cannot be reperformed or resampled. (A drum track containing no pitched instruments could be moderately resampled for tempo without adverse effects, but a pitched track could not). They are also used to create effects such as increasing the range of an instrument (like pitch shifting a guitar down an octave).

Resampling

The simplest way to change the duration or pitch of a digital audio clip is to resample it. This is a mathematical operation that effectively rebuilds a continuous waveform from its samples and then samples that waveform again at a different rate. When the new samples are played at the original sampling frequency, the audio clip sounds faster or slower. Unfortunately, the frequencies in the sample are always scaled at the same rate as the speed, transposing its perceived pitch up or down in the process. In other words, slowing down the recording lowers the pitch, speeding it up raises the pitch, and the two effects cannot be separated. This is analogous to speeding up or slowing down an analog recording, like a phonograph record or tape, creating the chipmunk effect.

Phase vocoder

One way of stretching the length of a signal without affecting the pitch is to build a phase vocoder after Flanagan, Golden, and Portnoff.

Basic steps:

1. compute the instantaneous frequency/amplitude relationship of the signal using the STFT, which is the discrete Fourier transform of a short, overlapping and smoothly windowed block of samples;
2. apply some processing to the Fourier transform magnitudes and phases (like resampling the FFT blocks); and
3. perform an inverse STFT by taking the inverse Fourier transform on each chunk and adding the resulting waveform chunks.

The phase vocoder handles sinusoid components well, but early implementations introduced considerable smearing on transient ("beat") waveforms at all non-integer compression/expansion rates, which renders the results phasey and diffuse. Recent improvements allow better quality results at all compression/expansion ratios but a residual smearing effect still remains.

The phase vocoder technique can also be used to perform pitch shifting, chorusing, timbre manipulation, harmonizing, and other unusual modifications, all of which can be changed as a function of time.

Time domain

SOLA

Rabiner and Schafer in 1978 put forth an alternate solution that works in the time domain: attempt to find the period (or equivalently the fundamental frequency) of a given section of the wave using some pitch detection algorithm (commonly the peak of the signal's autocorrelation, or sometimes cepstral processing), and crossfade one period into another.

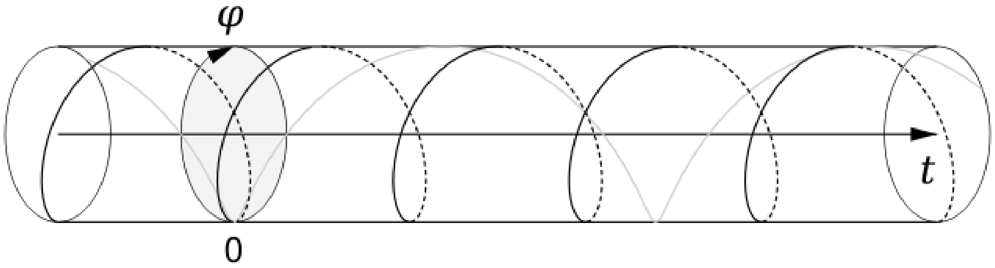
This is called time domain harmonic scaling or the synchronized overlap-add method (SOLA) and performs somewhat faster than the phase vocoder on slower machines but fails when the autocorrelation mis-estimates the period of a signal with complicated harmonics (such as orchestral pieces).

Adobe Audition (formerly Cool Edit Pro) seems to solve this by looking for the period closest to a center period that the user specifies, which should be an integer multiple of the tempo, and between 30 Hz and the lowest bass frequency.

This is much more limited in scope than the phase vocoder based processing, but can be made much less processor intensive, for real-time applications. It provides the most coherent results for single-pitched sounds like voice or musically monophonic instrument recordings.

High-end commercial audio processing packages either combine the two techniques (for example by separating the signal into sinusoid and transient waveforms), or use other techniques based on the wavelet transform, or artificial neural network processing, producing the highest-quality time stretching.

Untangling phase and time



Modelling a monophonic sound as observation along a helix of a function with a cylinder domain

Another way to shift pitch and stretch time is to separate phase and time in a monophonic sound such as the ones of melody instruments. By altering only the time control, you can stretch, shrink or reverse time, or generate loops as needed in sampling synthesizers. Time shrinkage can also be used for compression purposes. By altering only the phase control, you can shift the pitch or apply FM synthesis distortions to an existing sound. This can be used to play instruments alternatively to wavetable synthesis.

For controlling phase and time independently we would need to know the displacement of the sound for every pair of phase and time position. This corresponds to a cylinder as shown in the figure. However, a sound signal is a one-dimensional signal. You can consider this sound signal as observation of the full function on the cylinder. This is drawn as black line in the figure. The full function on the cylinder can be approximated by interpolating between points on the helix with (approximately) the same phase. From this function a different sound signal can be derived. E.g. in the figure the grey line shows the path of a sound that has the same time progression but a frequency lower than the original one, or a sound that has the same frequency and a faster time progression, or something between. In the end the whole process can be implemented for discrete sound signals as interpolation between values with similar phase and similar time.

Sinusoidal/Spectral Modeling

Another alternative method for time stretching relies on a spectral model of the signal. In this method, peaks are identified in frames using the STFT of the signal, and sinusoidal "tracks" are created by connecting peaks in adjacent frames. The tracks are then re-synthesized at a new time scale. This method can yield good results on both polyphonic and percussive material, especially when the signal is separated into sub-bands. However, this method is more computationally demanding than other methods.

Speed reading

Time stretching can be used with audio books and recorded lectures. Slowing down may improve comprehension of foreign languages .

While one might expect speeding up to reduce comprehension, Herb Friedman says that "Experiments have shown that the brain works most efficiently if the information rate through the ears--via speech--is the "average" reading rate, which is about 200-300 wpm (words per minute), yet the average rate of speech is in the neighborhood of 100-150 wpm."

Speeding up audio is seen as the equivalent of "speed reading".

Other

Time stretching is often used to adjust Radio commercials and the audio of Television advertisements to fit exactly into the 30 or 60 seconds available.

Pitch scaling

These techniques can also be used to transpose an audio sample while holding speed or duration constant. This may be accomplished by time stretching and then resampling back to the original length. Alternatively, the frequency of the sinusoids in a sinusoidal model may be altered directly, and the signal reconstructed at the appropriate time scale.

Transposing can be called **frequency scaling** or **pitch shifting**, depending on perspective.

For example, one could move the pitch of every note up by a perfect fifth, keeping the tempo the same. One can view this transposition as "pitch shifting", "shifting" each note up 7 keys on a piano keyboard, or adding a fixed amount on the Mel scale, or adding a fixed amount in linear pitch space. One can view the same transposition as "frequency scaling", "scaling" (multiplying) the frequency of every note by 3/2.

Musical transposition preserves the ratios of the harmonic frequencies that determine the sound's timbre, unlike the *frequency shift* performed by amplitude modulation, which adds a fixed frequency offset to the frequency of every note. (In theory one could perform a literal *pitch scaling* in which the musical pitch space location is scaled [a higher note would be shifted at a greater interval in linear pitch space than a lower note], but that is highly unusual, and not musical).

Time domain processing works much better here, as smearing is less noticeable, but scaling vocal samples distorts the formants into a sort of Alvin and the Chipmunks-like effect, which may be desirable or undesirable. A process that preserves the formants and character of a voice involves analyzing the signal with a channel vocoder or LPC vocoder plus any of several pitch detection algorithms and then resynthesizing it at a different fundamental frequency.

A detailed description of older analog recording techniques for pitch shifting can be found within the Alvin and the Chipmunks entry.

Chapter 4

Bilinear Transform

The **bilinear transform** (also known as **Tustin's method**) is used in digital signal processing and discrete-time control theory to transform continuous-time system representations to discrete-time and vice versa.

The bilinear transform is a special case of a conformal mapping (namely, the Möbius transformation), often used to convert a transfer function $H_a(s)$of a linear, time-invariant (LTI) filter in the continuous-time domain (often called an analog filter) to a transfer function $H_d(z)$of a linear, shift-invariant filter in the discrete-time domain (often called a digital filter although there are analog filters constructed with switched capacitors that are discrete-time filters). It maps positions on the $j\omega$axis, $Re[s] = 0$, in the s-plane to the unit circle, $|z| = 1$, in the z-plane. Other bilinear transforms can be used to warp the frequency response of any discrete-time linear system (for example to approximate the non-linear frequency resolution of the human auditory system) and are implementable in the discrete domain by replacing a system's unit delays $\left(z^{-1}\right)$with first order all-pass filters.

The transform preserves stability and maps every point of the frequency response of the continuous-time filter, $H_a(j\omega_a)$to a corresponding point in the frequency response of the discrete-time filter, $H_d(e^{j\omega_d T})$although to a somewhat different frequency, as shown in the Frequency warping section below. This means that for every feature that one sees in the frequency response of the analog filter, there is a corresponding feature, with identical gain and phase shift, in the frequency response of the digital filter but, perhaps, at a somewhat different frequency. This is barely noticeable at low frequencies but is quite evident at frequencies close to the Nyquist frequency.

Discrete-time approximation

The bilinear transform is a first-order approximation of the natural logarithm function that is an exact mapping of the z-plane to the s-plane. When the Laplace transform is performed on a discrete-time signal (with each element of the discrete-time sequence attached to a correspondingly delayed unit impulse), the result is precisely the Z transform of the discrete-time sequence with the substitution of

$$\begin{aligned} z &= e^{sT} \\ &= \frac{e^{sT/2}}{e^{-sT/2}} \\ &\approx \frac{1 + sT/2}{1 - sT/2} \end{aligned}$$

where T is the sample time (the reciprocal of the sampling frequency) of the discrete-time filter. The above bilinear approximation can be solved for s or a similar approximation for $s = (1/T)\ln(z)$ can be performed.

The inverse of this mapping (and its first-order bilinear approximation) is

$$\begin{aligned} s &= \frac{1}{T}\ln(z) \\ &= \frac{2}{T}\left[\frac{z-1}{z+1} + \frac{1}{3}\left(\frac{z-1}{z+1}\right)^3 + \frac{1}{5}\left(\frac{z-1}{z+1}\right)^5 + \frac{1}{7}\left(\frac{z-1}{z+1}\right)^7 + \cdots\right] \\ &\approx \frac{2}{T}\frac{z-1}{z+1} \\ &= \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} \end{aligned}$$

The bilinear transform essentially uses this first order approximation and substitutes into the continuous-time transfer function, $H_a(s)$

$$s \leftarrow \frac{2}{T}\frac{z-1}{z+1}.$$

That is

$$H_d(z) = H_a(s)\Big|_{s=\frac{2}{T}\frac{z-1}{z+1}} = H_a\left(\frac{2}{T}\frac{z-1}{z+1}\right).$$

Stability and minimum-phase property preserved

A continuous-time causal filter is stable if the poles of its transfer function fall in the left half of the complex s-plane. A discrete-time causal filter is stable if the poles of its transfer function fall inside the unit circle in the complex z-plane. The bilinear transform maps the left half of the complex s-plane to the interior of the unit circle in the z-plane. Thus filters designed in the continuous-time domain that are stable are converted to filters in the discrete-time domain that preserve that stability.

Likewise, a continuous-time filter is minimum-phase if the zeros of its transfer function fall in the left half of the complex s-plane. A discrete-time filter is minimum-phase if the zeros of its transfer function fall inside the unit circle in the complex z-plane. Then the same mapping property assures that continuous-time filters that are minimum-phase are converted to discrete-time filters that preserve that property of being minimum-phase.

Example

As an example take a simple low-pass RC filter. This continuous-time filter has a transfer function

$$\begin{aligned} H_a(s) &= \frac{1/sC}{R+1/sC} \\ &= \frac{1}{1+RCs}. \end{aligned}$$

If we wish to implement this filter as a digital filter, we can apply the bilinear transform by substituting for *s* the formula above; after some reworking, we get the following filter representation:

$$\begin{aligned} H_d(z) &= H_a\left(\frac{2}{T}\frac{z-1}{z+1}\right) \\ &= \frac{1}{1+RC\left(\frac{2}{T}\frac{z-1}{z+1}\right)} \\ &= \frac{1+z}{(1-2RC/T)+(1+2RC/T)z}. \\ &= \frac{1+z^{-1}}{(1+2RC/T)+(1-2RC/T)z^{-1}}. \end{aligned}$$

The coefficients of the denominator are the 'feed-backward' coefficients and the coefficients of the numerator are the 'feed-forward' coefficients used to implement a real-time digital filter.

Frequency warping

To determine the frequency response of a continuous-time filter, the transfer function $H_a(s)$is evaluated at $s = j\omega$which is on the $j\omega$axis. Likewise, to determine the frequency response of a discrete-time filter, the transfer function $H_d(z)$is evaluated at $z = e^{j\omega T}$which is on the unit circle, $|z| = 1$. When the actual frequency of $\omega$is input to the discrete-time filter designed by use of the bilinear transform, it is desired to know at what frequency, ω_a, for the continuous-time filter that this $\omega$is mapped to.

$$
\begin{aligned}
H_d(z) &= H_a\left(\frac{2}{T}\frac{z-1}{z+1}\right) \\
H_d(e^{j\omega T}) &= H_a\left(\frac{2}{T}\frac{e^{j\omega T}-1}{e^{j\omega T}+1}\right) \\
&= H_a\left(\frac{2}{T}\cdot\frac{e^{j\omega T/2}\left(e^{j\omega T/2}-e^{-j\omega T/2}\right)}{e^{j\omega T/2}\left(e^{j\omega T/2}+e^{-j\omega T/2}\right)}\right) \\
&= H_a\left(\frac{2}{T}\cdot\frac{\left(e^{j\omega T/2}-e^{-j\omega T/2}\right)}{\left(e^{j\omega T/2}+e^{-j\omega T/2}\right)}\right) \\
&= H_a\left(j\frac{2}{T}\cdot\frac{\left(e^{j\omega T/2}-e^{-j\omega T/2}\right)/(2j)}{\left(e^{j\omega T/2}+e^{-j\omega T/2}\right)/2}\right) \\
&= H_a\left(j\frac{2}{T}\cdot\frac{\sin(\omega T/2)}{\cos(\omega T/2)}\right) \\
&= H_a\left(j\frac{2}{T}\cdot\tan\left(\omega\frac{T}{2}\right)\right) \\
&= H_a\left(j\omega_a\right).
\end{aligned}
$$

This shows that every point on the unit circle in the discrete-time filter z-plane, $z = e^{j\omega T}$is mapped to a point on the $j\omega$axis on the continuous-time filter s-plane, $s = j\omega_a$. That is, the discrete-time to continuous-time frequency mapping of the bilinear transform is

$$\omega_a = \frac{2}{T}\tan\left(\omega\frac{T}{2}\right)$$

and the inverse mapping is

$$\omega = \frac{2}{T} \arctan \left(\omega_a \frac{T}{2} \right) .$$

The discrete-time filter behaves at frequency $\omega$the same way that the continuous-time filter behaves at frequency $(2/T)\tan(\omega T/2)$. Specifically, the gain and phase shift that the discrete-time filter has at frequency $\omega$is the same gain and phase shift that the continuous-time filter has at frequency $(2/T)\tan(\omega T/2)$. This means that every feature, every "bump" that is visible in the frequency response of the continuous-time filter is also visible in the discrete-time filter, but at a different frequency. For low frequencies (that is, when $\omega \ll 2/T$or $\omega_a \ll 2/T$), $\omega \approx \omega_a$.

One can see that the entire continuous frequency range

$$-\infty < \omega_a < +\infty$$

is mapped onto the fundamental frequency interval

$$-\frac{\pi}{T} < \omega < +\frac{\pi}{T}.$$

The continuous-time filter frequency $\omega_a = 0$corresponds to the discrete-time filter frequency $\omega = 0$and the continuous-time filter frequency $\omega_a = \pm\infty$correspond to the discrete-time filter frequency $\omega = \pm\pi/T.$

One can also see that there is a nonlinear relationship between $\omega_a$and ω.This effect of the bilinear transform is called ***frequency warping***. The continuous-time filter can be designed to compensate for this frequency warping by setting $\omega_a = \frac{2}{T} \tan \left(\omega \frac{T}{2} \right)$for every frequency specification that the designer has control over (such as corner frequency or center frequency). This is called ***pre-warping*** the filter design.

The main advantage of the warping phenomenon is the absence of aliasing distortion of the frequency response characteristic, such as observed with Impulse invariance. It is necessary, however, to compensate for the frequency warping by pre-warping the given frequency specifications of the continuous-time system. These pre-warped specifications may then be used in the bilinear transform to obtain the desired discrete-time system.

Chapter 5

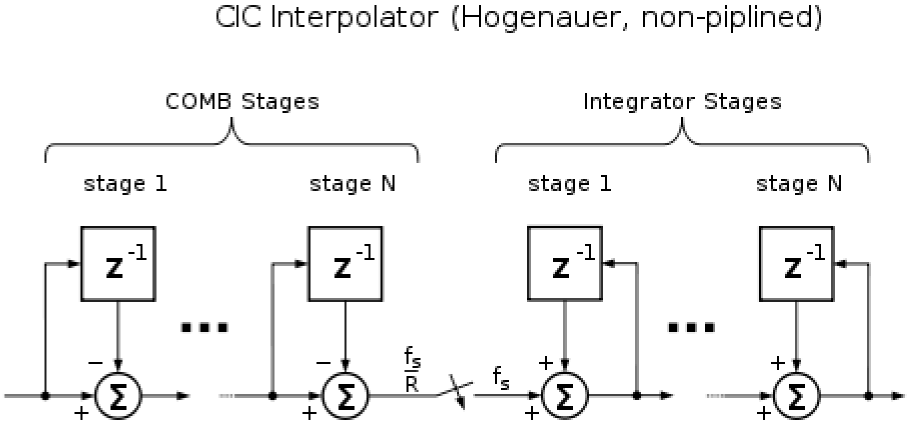
Cascaded Integrator-Comb Filter & Digital Down Converter

Cascaded Integrator-Comb Filter

In digital signal processing, a **cascaded integrator-comb (CIC)** is an optimized class of finite impulse response filter combined with an interpolator or decimator.

A CIC filter consists of one or more integrator and comb filter pairs. In the case of a decimating CIC, the input signal is fed through one or more cascaded integrators, then a down-sampler, followed by one or more comb sections (equal in number to the number of integrators). An interpolating CIC is simply the reverse of this architecture, with the down-sampler replaced with a zero-stuffer (up-sampler).

The CIC filter



CIC Interpolator by factor R, Hogenauer Non-Pipelined

CIC filters were invented by Eugene B. Hogenauer, and are a class of FIR filters used in multi-rate processing. The CIC filter finds applications in interpolation and decimation. Unlike most FIR filters, it has a decimator or interpolator built into the architecture. The figure at the right shows the Hogenauer architecture for a CIC Interpolator.

The system function for the composite CIC filter referenced to the high sampling rate, f_s is:

$$\begin{aligned} H(z) &= \left[\sum_{k=0}^{RM-1} z^{-k}\right]^N \\ &= \left(\frac{1-z^{-RM}}{1-z^{-1}}\right)^N \end{aligned}$$

Where:

R = decimation or interpolation ratio
M = number of samples per stage (usually 1 but sometimes 2)
N = number of stages in filter

Characteristics of CIC Filters

1. Linear phase response;
2. Utilize only delay and addition and subtraction; that is, it requires no multiplication operations;

CIC as a moving average filter

A CIC filter is an efficient implementation of a moving average filter. To see this, consider how a moving average filter can be implemented recursively by adding the newest sample $x[n]$ to the previous result $y[n-1]$ and subtracting the oldest sample. Omitting the division by RM, we have:

$$\begin{aligned} y[n] &= \sum_{k=0}^{RM-1} x[n-k] \\ &= y[n-1] + x[n] - x[n-RM]. \end{aligned}$$

The second equality corresponds to a comb ($c[n] = x[n] - x[n-RM]$) followed by an integrator ($y[n] = y[n-1] + c[n]$). The conventional CIC structure is obtained by cascading N identical moving average filters, then rearranging the sections to place all integrators first (decimator) or combs first (interpolator). Such rearrangement is possible because both combs and integrators are LTI. For an interpolator, the upsampler which normally precedes the interpolation filter can be passed through the comb sections using a Noble identity, reducing the number of delay elements needed by a factor of *R*. Similarly, for a decimator, the downsampler which normally follows the decimation filter can be moved before the comb sections.

The equivalence of a CIC to moving average filter allows us to trivially calculate its bit growth as $N\log_2(RM)$.

Comparison with other filters

CIC filters are used in multi-rate processing. An FIR filter is used in a wide array of applications, and can be used in multi-rate processing in conjunction with an interpolator or decimator. CIC filters have low pass frequency characteristics, while FIR filters can have low-pass, high-pass, or band-pass frequency characteristics. CIC filters use only addition and subtraction. FIR filters use addition, subtraction, but most FIR filters also require multiplication. CIC filters have a specific frequency roll-off, while low pass FIR filters can have an arbitrarily sharp frequency roll-off.

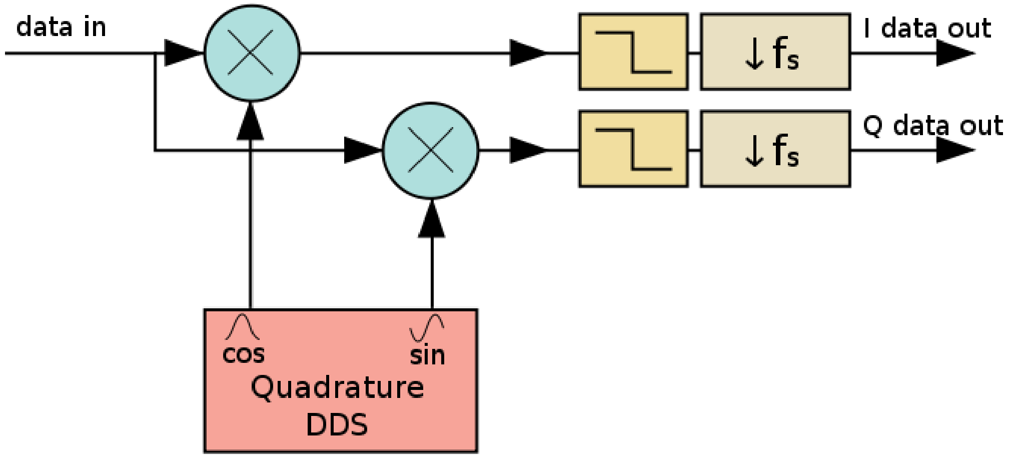
CIC filters are in general much more economical than general FIR filters, but tradeoffs are involved. In cases where only a small amount of interpolation or decimation are needed, FIR filters generally have the advantage. However, when rates change by a factor of 10 or more, achieving a useful FIR filter anti-aliasing stop band requires exponentially increasing numbers of FIR taps.

For large rate changes, a CIC has a significant advantage over a FIR filter with respect to architectural and computational efficiency. Additionally, CIC filters can typically be reconfigured for different rates by changing nothing more than the decimation/interpolation section assuming the bit width of the integrators and comb sections meets certain mathematical criteria based on the maximum possible rate change.

Whereas a FIR filter can use fixed or floating point math, a CIC filter uses only fixed point math. This is necessary because, as a recursively implemented FIR filter, a CIC filter relies on exact cancellation of poles from the integrator sections by zeros from the comb sections. While the reasons are less than intuitive, an inherent characteristic of the CIC architecture is that if fixed bit length overflows occur in the integrators, they are corrected in the comb sections.

The range of filter shapes and responses available from a CIC filter is somewhat limited. Larger amounts of stopband rejection can be achieved by increasing the number of poles. However, doing so requires an increase in bit width in the integrator and comb sections which increases filter complexity. The shape of the filter response provides even fewer degrees of design freedom. For this reason, many real-world filtering requirements cannot be met by a CIC filter alone. However, a CIC filter followed by a short to moderate length FIR or IIR proves highly applicable. Additionally, the FIR filter shape is normalized relative to the CIC's sampling rate at the FIR/CIC interface so one set of FIR coefficients can be used over a range of CIC interpolation and decimation rates.

Digital Down Converter



In digital signal processing, a **digital down-converter** (**DDC**) converts a digitized real signal centered at an intermediate frequency (IF) to a basebanded complex signal centered at zero frequency. In addition to downconversion, DDC's typically decimate to a lower sampling rate, allowing follow-on signal processing by lower speed processors.

Architecture

A DDC consists of three subcomponents: a direct digital synthesizer (DDS), a low-pass filter (LPF), and a downsampler (which may be integrated into the low-pass filter).

The DDS generates a complex sinusoid at the intermediate to downconverting by creating a difference signal at the IF minus the DDS frequency, they also upconvert, generating an unwanted signal at the sum of the two frequencies.

Any suitable low-pass filter can be used including FIR, IIR and CIC filters. The most common choice is a FIR filter for low amounts of decimation (less than ten) or a CIC filter followed by a FIR filter for larger downsampling ratios.

Variations on the DDC

Several variations on the DDC are useful, including many that input a feedback signal into the DDS. These include:

- Decision directed carrier recovery phase locked loops in which the I and Q are compared to the nearest ideal constellation point of a PSK signal, and the resulting error signal is filtered and fed back into the DDS

- A Costas loop in which the I and Q are multiplied and low pass filtered as part of a BPSK/QPSK carrier recovery loop

Implementation

DDC’s are most commonly implemented in logic in field-programmable gate arrays or application-specific integrated circuits. While software implementations are also possible, operations in the DDS, multipliers and input stages of the lowpass filters all run at the sampling rate of the input data. This data is commonly taken directly from analog to digital converters (ADC’s) sampling at tens or hundreds of MHz, which is beyond the real time computational capabilities of software processors.

Chapter 6

Delta-Sigma Modulation

Delta-sigma (**ΔΣ**; or **sigma-delta**, **ΣΔ**) modulation is a method for encoding high resolution signals into lower resolution signals using pulse-density modulation. This technique has found increasing use in modern electronic components such as analog-to-digital converters (ADCs) and digital-to-analog converters (DACs), frequency synthesizers, switched-mode power supplies and motor controlers.

One of the earliest and most widespread uses of delta-sigma modulation is in data conversion. An ADC or DAC circuit which implements this technique can achieve very high resolutions using low-cost complementary metal–oxide–semiconductor manufacturing processes used to produce digital integrated circuits. For this reason it did not come into widespread use until improvements in silicon technology in the 1980s.

Given a particular fabrication process, a sigma-delta ADC can give more bits of resolution than any other ADC structure except the integrating ADC.

Analog to digital conversion

Description

For the ADC, the method can be thought of as a voltage-controlled oscillator, where the controlling voltage is the voltage to be measured and where linearity and proportionality are determined by a negative feedback loop.

The oscillator's output is a pulse stream, each pulse of which has a known, constant amplitude V and duration dt, and thus has a known integral V dt but variable separating interval. The interval between pulses is determined by the feedback loop. A low input voltage produces a long interval between pulses and a high input voltage produces a short interval between pulses. In fact, neglecting switching errors, the interval between pulses is proportional to the inverse of the mean of the input voltage during that interval and thus over that interval, t_s, is a sample of the mean of the input voltage proportional to V/t_s. The feedback loop is arranged so that the integral of the input voltage is matched within one count by the integral of the pulse stream x. The output count finally produced is the digitization of the input voltage determined by counting the pulses produced in the way described above in a fixed summing interval $N\ \mathrm{d}t$ producing a count, Σ. The integral of the pulse stream is $\Sigma V\ \mathrm{d}t$ which is produced over an interval $N\ \mathrm{d}t$ and thus the average of the input voltage over the summing period is $V\Sigma/N$ and is the mean of means and so subject to little variance. The accuracy achieved depends on the accuracy with which V is known and the precision or resolution is within one count in N.

Variations in scaling can be produced by either varying the fixed summing interval, Ndt or by counting down the pulses by a fixed ratio or both methods can be used.

The pulses described above may be treated as the Dirac δ (delta) function in a formal analysis and the count as Σ (sigma). It is these pulses which are transmitted for delta-sigma modulation but are counted to form sigma in the case of analogue to digital conversion.

NB. In electronic usage pulse is the standard term. In mathematics the Dirac delta function is an impulse which, when integrated, produces a step. The analysis of this circuit depends on that fact.

Analysis

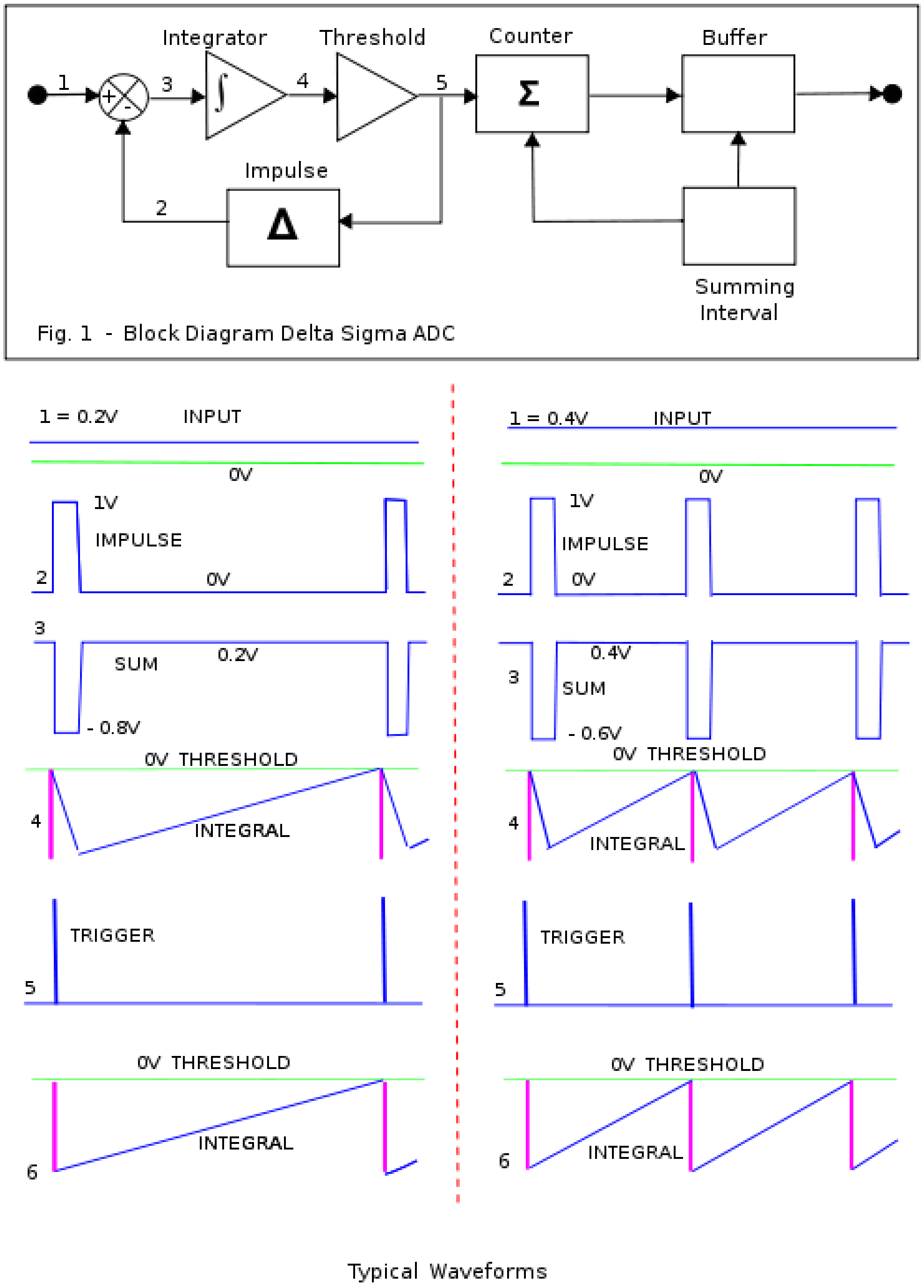


Fig. 1: Block diagram and waveforms for a delta sigma ADC.

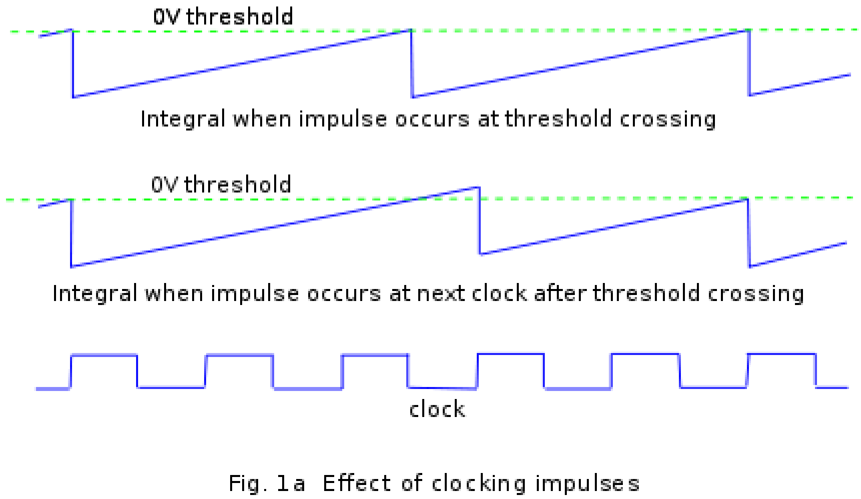

0V threshold
Integral when impulse occurs at threshold crossing
0V threshold
Integral when impulse occurs at next clock after threshold crossing
clock
Fig. 1a Effect of clocking impulses

Fig. 1a: Effect of clocking impulses

Shown below the block diagram illustrated in Fig. 1 are waveforms at points designated by numbers 1 to 5 for an input of 0.2 volts on the left and 0.4 volts on the right.

In most practical applications the summing interval is large compared with the impulse duration and for signals which are a significant fraction of full scale the variable separating interval is also small compared with the summing interval. The Nyquist–Shannon sampling theorem requires two samples to render a varying input signal. The samples appropriate to this criterion are two successive Σ counts taken in two successive summing intervals. The summing interval, which must accommodate a large count in order to achieve adequate precision, is inevitably long so that the converter can only render relatively low frequencies. Hence it is convenient and fair to represent the input voltage (1) as constant over a few impulses.

Consider first the waveforms on the left.

1 is the input and for this short interval is constant at 0.2 V. The stream of delta impulses is shown at 2 and the difference between 1 and 2 is shown at 3. This difference is integrated to produce the waveform 4. The threshold detector generates a pulse 5 which starts as the waveform 4 crosses the threshold and is sustained until the waveform 4 falls below the threshold. Within the loop 5 triggers the impulse generator and external to the loop increments the counter. The summing interval is a prefixed time and at its expiry the count is strobed into the buffer and the counter reset.

It is necessary that the ratio between the impulse interval and the summing interval is equal to the maximum (full scale) count. It is then possible for the impulse duration and the summing interval to be defined by the same clock with a suitable arrangement of logic and counters. This has the advantage that neither interval has to be defined with absolute precision as only the ratio is important. Then to achieve overall accuracy it is only necessary that the amplitude of the impulse be accurately defined.

On the right the input is now 0.4 V and the sum during the impulse is −0.6 V as opposed to −0.8 V on the left. Thus the negative slope during the impulse is lower on the right than on the left.

Also the sum is 0.4 V on the right during the interval as opposed to 0.2 V on the left. Thus the positive slope outside the impulse is higher on the right than on the left.

The resultant effect is that the integral (4) crosses the threshold more quickly on the right than on the left. A full analysis would show that in fact the interval between threshold crossings on the right is half that on the left. Thus the frequency of impulses is doubled. Hence the count increments at twice the speed on the right to that on the left which is consistent with the input voltage being doubled.

Construction of the waveforms illustrated at (4) is aided by concepts associated with the Dirac delta function in that all impulses of the same strength produce the same step when integrated, by definition. Then (4) is constructed using an intermediate step (6) in which each integrated impulse is represented by a step of the assigned strength which decays to zero at the rate determined by the input voltage. The effect of the finite duration of the impulse is constructed in (4) by drawing a line from the base of the impulse step at zero volts to intersect the decay line from (6) at the full duration of the impulse.

As stated, Fig. 1 is a simplified block diagram of the delta-sigma ADC in which the various functional elements have been separated out for individual treatment and which tries to be independent of any particular implementation. Many particular implementations seek to define the impulse duration and the summing interval from the same clock as discussed above but in such a way that the start of the impulse is delayed until the next occurrence of the appropriate clock pulse boundary. The effect of this delay is illustrated in Fig. 1a for a sequence of impulses which occur at a nominal 2.5 clock intervals, firstly for impulses generated immediately the threshold is crossed as previously discussed and secondly for impulses delayed by the clock. The effect of the delay is firstly that the ramp continues until the onset of the impulse, secondly that the impulse produces a fixed amplitude step so that the integral retains the excess it acquired during the impulse delay and so the ramp restarts from a higher point and is now on the same locus as the unclocked integral. The effect is that, for this example, the undelayed impulses will occur at clock points 0, 2.5, 5, 7.5, 10, etc. and the clocked impulses will occur at 0, 3, 5, 8, 10, etc. The maximum error that can occur due to clocking is marginally less than one count. Although the Sigma-Delta converter is generally implemented using a common clock to define the impulse duration and the summing interval it is not absolutely necessary and an implementation in which the durations are independently defined avoids one source of noise, the noise generated by waiting for the next common clock boundary. Where noise is a primary consideration that overrides the need for absolute amplitude accuracy; e.g., in bandwidth limited signal transmission, separately defined intervals may be implemented.

Practical Implementation

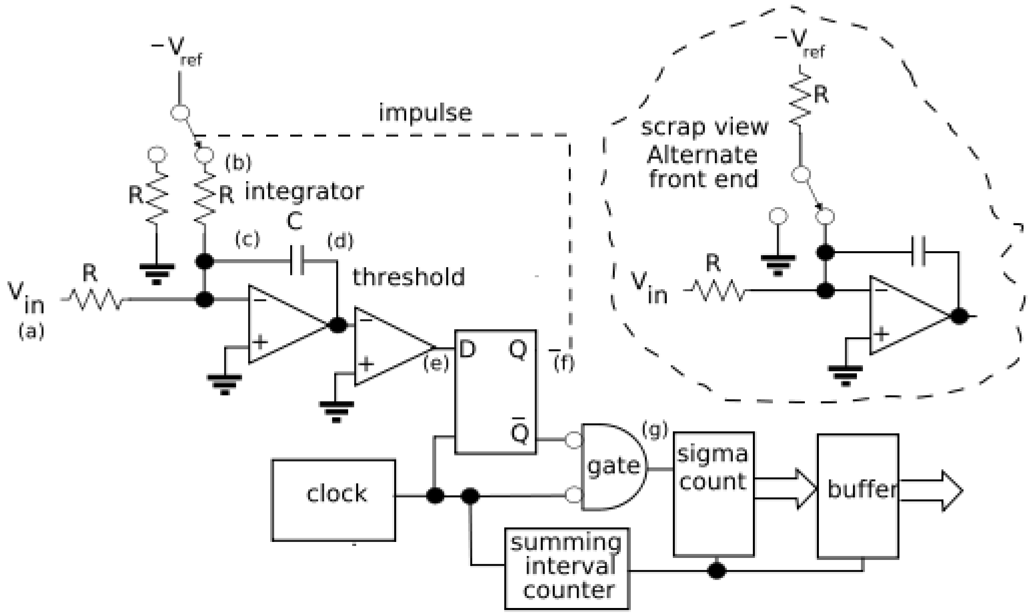


Fig. 1b - Circuit Diagram

Fig. 1b: circuit diagram

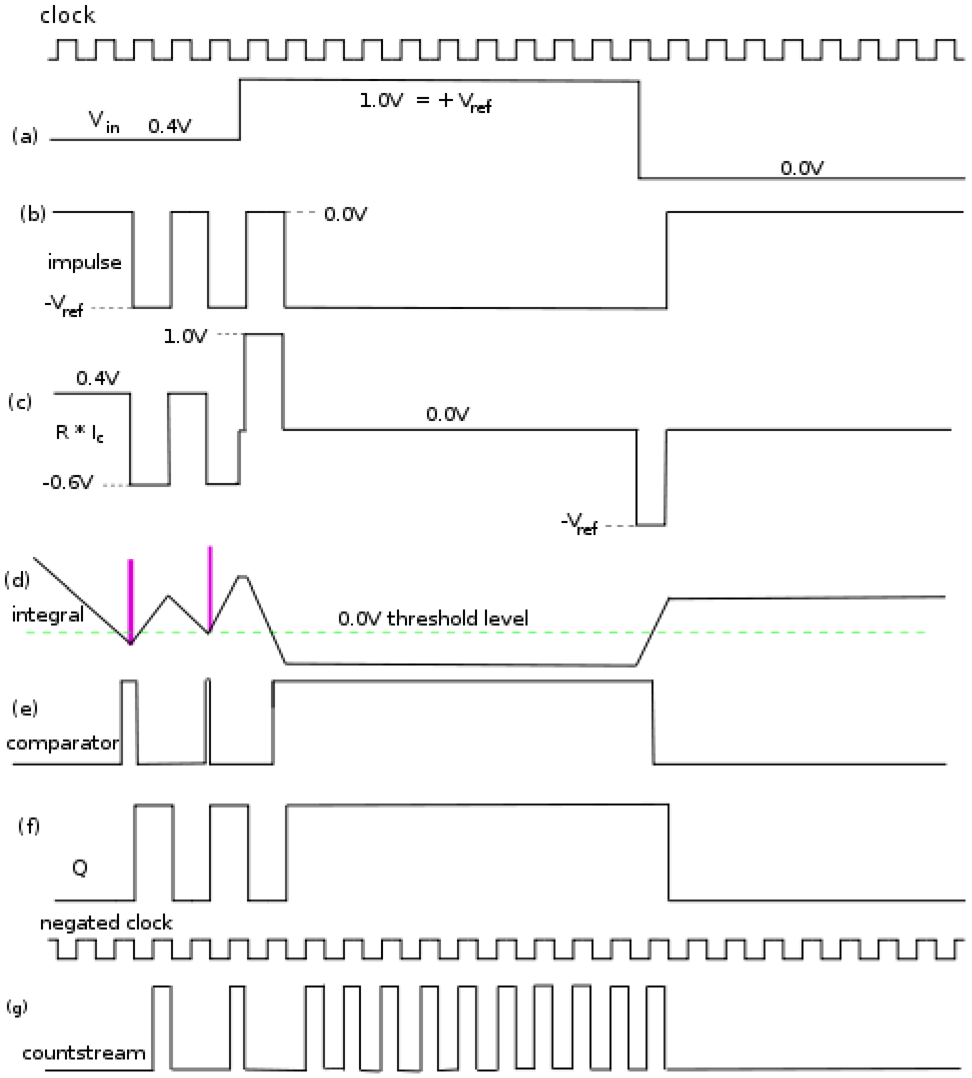


Fig. 1c - ADC waveforms

Fig. 1c: ADC waveforms

A circuit diagram for a practical implementation is illustrated, Fig 1b and the associated waveforms Fig. 1c. A scrap view of an alternative front end is shown in Fig. 1b which has the advantage that the voltage at the switch terminals are relatively constant and close to 0.0 V. Also the current generated through R by $-V_{ref}$ is constant at $-V_{ref}/R$ so that much less noise is radiated to adjacent parts of the circuit. Then this would be the preferred front end in practice but, in order to show the impulse as a voltage pulse so as to be consistent with previous discussion, the front end given here, which is an electrical equivalent, is used.

From the top of Fig 1c the waveforms, labelled as they are on the circuit diagram, are:-

The clock.

(a) V_{in}. This is shown as varying from 0.4 V initially to 1.0 V and then to zero volts to show the effect on the feedback loop.

(b) The impulse waveform. It will be discovered how this acquires its form as we traverse the feedback loop.

(c) The current into the capacitor, I_c, is the linear sum of the impulse voltage upon R and V_{in} upon R. To show this sum as a voltage the product $R \times I_c$ is plotted. The input impedance of the amplifier is regarded as so high that the current drawn by the input is neglected.

(d) The negated integral of I_c. This negation is standard for the op. amp. implementation of an integrator and comes about because the current into the capacitor at the amplifier input is the current out of the capacitor at the amplifier output and the voltage is the integral of the current divided by the capacitance of C.

(e) The comparator output. The comparator is a very high gain amplifier with its plus input terminal connected for reference to 0.0 V. Whenever the negative input terminal is taken negative with respect the positive terminal of the amplifier the output saturates positive and conversely negative saturation for positive input. Thus the output saturates positive whenever the integral (d) goes below the 0 V reference level and remains there until (d) goes positive with respect to the reference level.

(f) The impulse timer is a D type positive edge triggered flip flop. Input information applied at D is transferred to Q on the occurrence of the positive edge of the clock pulse. thus when the comparator output (e) is positive Q goes positive or remains positive at the next positive clock edge. Similarly, when (e) is negative Q goes negative at the next positive clock edge. Q controls the electronic switch to generate the current impulse into the integrator. Examination of the waveform (e) during the initial period illustrated, when V_{in} is 0.4 V, shows (e) crossing the threshold well before the trigger edge (positive edge of the clock pulse) so that there is an appreciable delay before the impulse starts. After the start of the impulse there is further delay while (e) climbs back past the threshold. During this time the comparator output remains high but goes low before the next trigger edge. At that next trigger edge the impulse timer goes low to follow the comparator. Thus the clock determines the duration of the impulse. For the next impulse the threshold is crossed immediately before the trigger edge and so the comparator is only briefly positive. V_{in} (a) goes to full scale, $+V_{ref}$, shortly before the end of the next impulse. For the remainder of that impulse the capacitor current (c) goes to zero and hence the integrator slope briefly goes to zero. Following this impulse the full scale positive current is flowing (c) and the integrator sinks at its maximum rate and so crosses the threshold well before the next trigger edge. At that edge the impulse starts and the Vin current is now matched by the reference current so that the net capacitor current (c) is zero. Then the integration now has zero slope and remains at the negative value it had at the start of the impulse. This has the effect that the impulse current remains switched on because Q is stuck positive because the comparator is stuck positive at every trigger edge. This is consistent with contiguous, butting impulses which is required at full scale input.

Eventually Vin (a) goes to zero which means that the current sum (c) goes fully negative and the integral ramps up. It shortly thereafter crosses the threshold and this in turn is followed by Q, thus switching the impulse current off. The capacitor current (c) is now zero and so the integral slope is zero, remaining constant at the value it had acquired at the end of the impulse.

(g) The countstream is generated by gating the negated clock with Q to produce this waveform. Thereafter the summing interval, sigma count and buffered count are produced using appropriate counters and registers. The V_{in} waveform is approximated by passing the countstream (g) into a low pass filter, however it suffers from the defect discussed in the context of Fig. 1a. One possibility for reducing this error is to halve the feedback pulse length to half a clock period and double its amplitude by halving the impulse defining resistor thus producing an impulse of the same strength but one which never butts onto its adjacent impulses. Then there will be a threshold crossing for every impulse. In this arrangement a monostable flip flop triggered by the comparator at the threshold crossing will closely follow the threshold crossings and thus eliminate one source of error, both in the ADC and the sigma delta modulator.

Remarks

We have mainly dealt with the analogue to digital converter as a stand alone function which achieves astonishing accuracy with what is now a very simple and cheap architecture. Initially the Delta-Sigma configuration was devised by INOSE et al. to solve problems in the accurate transmission of analog signals. In that application it was the pulse stream that was transmitted and the original analog signal recovered with a low pass filter after the received pulses had been reformed. This low pass filter performed the summation function associated with Σ. The highly mathematical treatment of transmission errors was introduced by them and is appropriate when applied to the pulse stream but these errors are lost in the accumulation process associated with Σ to be replaced with the errors associated with the mean of means when discussing the ADC. For those uncomfortable with this assertion consider this.

It is well known that by Fourier analysis techniques the incoming waveform can be represented over the summing interval by the sum of a constant plus a fundamental and harmonics each of which has an exact integer number of cycles over the sampling period. It is also well known that the integral of a sine wave or cosine wave over one or more full cycles is zero. Then the integral of the incoming waveform over the summing interval reduces to the integral of the constant and when that integral is divided by the summing interval it becomes the mean over that interval. The interval between pulses is proportional to the inverse of the mean of the input voltage during that interval and thus over that interval, ts, is a sample of the mean of the input voltage proportional to V/ts. Thus the average of the input voltage over the summing period is VΣ/N and is the mean of means and so subject to little variance.

Unfortunately the analysis for the transmitted pulse stream has, in many cases, been carried over, uncritically, to the ADC.

A very accurate transmission system with constant sampling rate may be formed using the full arrangement shown here by transmitting the samples from the buffer protected with redundancy error correction. In this case there will be a trade off between bandwidth and N, the size of the buffer. The signal recovery system will require redundancy error checking, digital to analog conversion,and sample and hold circuitry. A possible further enhancement is to include some form of slope regeneration.

The above description shows why the impulse is called delta. The integral of an impulse is a step. A one bit DAC may be expected to produce a step and so must be a conflation of an impulse and an integration. The analysis which treats the impulse as the output of a 1-bit DAC hides the structure behind the name (sigma delta) and cause confusion and difficulty interpreting the name as an indication of function. This analysis is very widespread but is deprecated.

A modern alternative method for generating voltage to frequency conversion is discussed in synchronous voltage to frequency converter (SVFC) which may be followed by a counter to produce a digital representation in a similar manner to that described above.

Digital to Analog Conversion

The digital to analog converter (DAC) arrangement can be thought of as open loop with a counter, sigma, which is preloaded with the number to be converted. The counter is counted down to zero by a series of impulses, delta. As above these impulses are of fixed amplitude and duration. At the start an integrator is set to zero and then integrates the impulses to form the analog voltage equivalent of the starting number.

Relationship to Δ-modulation

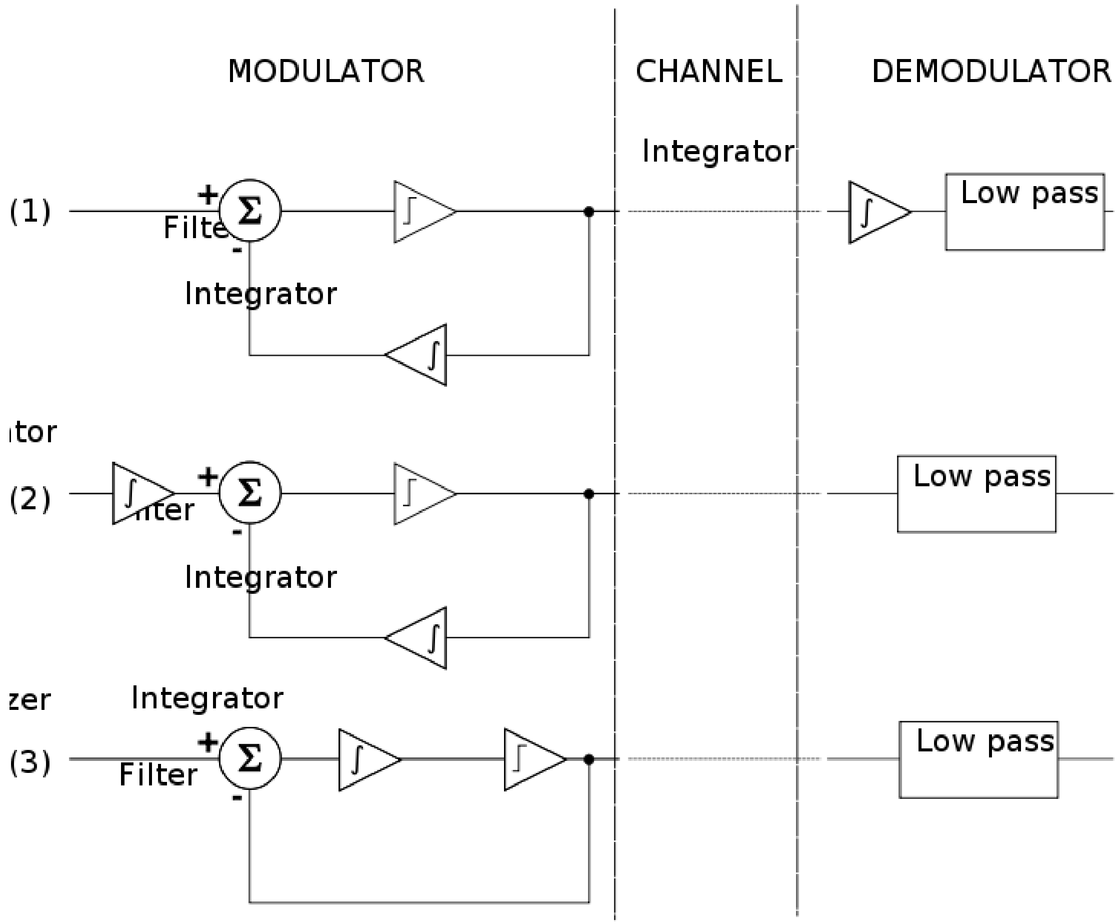


Fig. 2: Derivation of ΔΣ- from Δ-modulation

ΔΣ modulation (SDM) is inspired by Δ modulation (DM), as shown in Fig. 2. If quantization was homogeneous (e.g., if it was linear), the following would be a sufficient derivation of the equivalence of DM and SDM:

1. Start with a block diagram of a Δ-modulator/demodulator.
2. The linearity property of integration ($\int a + \int b = \int (a + b)$) makes it possible to move the integrator, which reconstructs the analog signal in the demodulator section, in front of the Δ-modulator.
3. Again, the linearity property of the integration allows the two integrators to be combined and a ΔΣ-modulator/demodulator block diagram is obtained.

However, the quantizer is **not** homogeneous, and so this explanation is flawed. It's true that ΔΣ is *inspired* by Δ-modulation, but the two are distinct in operation. From the first block diagram in Fig. 2, the integrator in the feedback path can be removed if the feedback is taken directly from the input of the low-pass filter. Hence, for delta modulation of input signal u, the low-pass filter sees the signal

$$y_{\mathrm{DM}} = \int \mathrm{Quantize}\left(u - y_{\mathrm{DM}}\right).$$

However, sigma-delta modulation of the same input signal places at the low-pass filter

$$y_{\mathrm{SDM}} = \mathrm{Quantize}\left(\int \left(u - y_{\mathrm{SDM}}\right)\right).$$

In other words, SDM and DM swap the position of the integrator and quantizer. The net effect is a simpler implementation that has the added benefit of shaping the quantization noise away from signals of interest (i.e., signals of interest are low-pass filtered while quantization noise is high-pass filtered). This effect becomes more dramatic with increased oversampling, which allows for quantization noise to be somewhat programmable. On the other hand, Δ-modulation shapes both noise and signal equally.

Additionally, the quantizer (e.g., comparator) used in DM has a small output representing a small step up and down the quantized approximation of the input while the quantizer used in SDM must take values *outside* of the range of the input signal, as shown in Fig. 3.

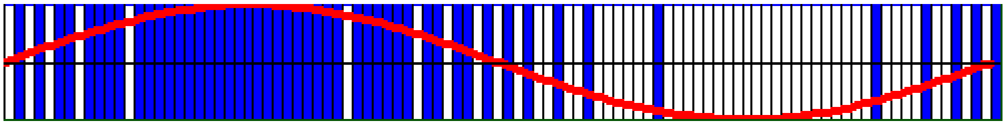


Fig. 3: An example of SDM of 100 samples of one period a sine wave. 1-bit samples (e.g., comparator output) overlaid with sine wave where logic high (e.g., $+V_{CC}$) represented by blue and logic low (e.g., $-V_{CC}$) represented by white.

In general, ΔΣ has some advantages versus Δ modulation:

- The whole structure is simpler:
    - Only one integrator is needed
    - The demodulator can be a simple linear filter (e.g., RC or LC filter) to reconstruct the signal
    - The quantizer (e.g., comparator) can have full-scale outputs
- The quantized value is the integral of the difference signal, which makes it less sensitive to the rate of change of the signal.

Principle

The principle of the ΔΣ architecture is to make rough evaluations of the signal, to measure the error, integrate it and then compensate for that error. The mean output value is then equal to the mean input value if the integral of the error is finite. A demonstration applet is available online to simulate the whole architecture.

Variations

There are many kinds of ADC that use this delta-sigma structure. The above analysis focuses on the simplest 1st-order, 2-level, uniform-decimation sigma-delta ADC. Many ADCs use a second-order 5-level sinc3 sigma-delta structure.

2nd order and higher order modulator

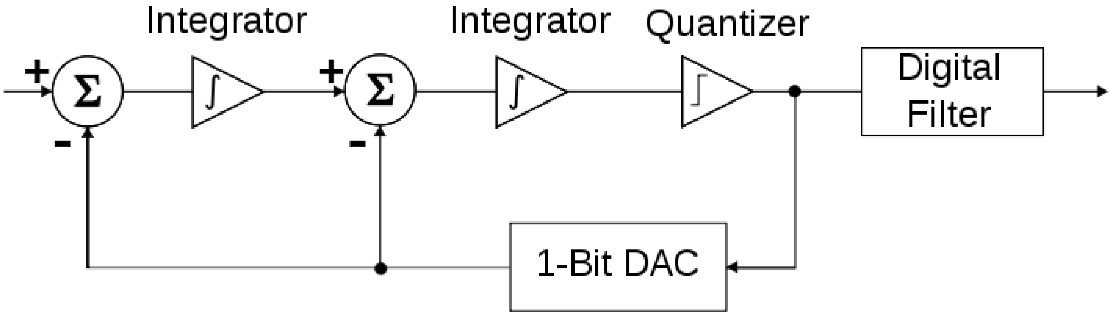


Fig. 4: Block diagram of a 2^{nd} order ΔΣ modulator

The number of integrators, and consequently, the numbers of feedback loops, indicates the *order* of a ΔΣ-modulator; a 2^{nd} order ΔΣ modulator is shown in Fig. 4. First order modulators are unconditionally stable, but stability analysis must be performed for higher order modulators.

3-level and higher quantizer

The modulator can also be classified by the number of bits it has in output, which strictly depends on the output of the quantizer. The quantizer can be realized with a *N-level* comparator, thus the modulator has log_2N-bit output. A simple comparator has 2 levels and so is 1 bit quantizer; a 3-level quantizer is called a "1.5" bit quantizer; a 4-level quantizer is a 2 bit quantizer; a 5-level quantizer is called a "2.5 bit" quantizer.

Decimation structures

The conceptually simplest decimation structure is a counter that is reset to zero at the beginning of each integration period, then read out at the end of the integration period.

The multi-stage noise shaping (MASH) structure has a noise shaping property, and is commonly used in digital audio and fractional-N frequency synthesizers. It comprises two or more cascaded overflowing accumulators, each of which is equivalent to a first-order sigma delta modulator. The carry outputs are combined through summations and delays to produce a binary output, the width of which depends on the number of stages (order) of the MASH. Besides its noise shaping function, it has two more attractive properties:

- simple to implement in hardware; only common digital blocks such as accumulators, adders, and D flip-flops are required
- unconditionally stable (there are no feedback loops outside the accumulators)

A very popular decimation structure is the *sinc* filter. For 2nd order modulators, the *sinc3* filter is close to optimum.

Quantization theory formulas

When a signal is quantized, the resulting signal approximately has the second-order statistics of a signal with independent additive white noise. Assuming that the signal value is in the range of one step of the quantized value with an equal distribution, the root mean square value of this quantization noise is

$$e_{\mathrm{rms}} = \sqrt{\frac{1}{\Delta}\int_{-\Delta/2}^{+\Delta/2} e^2\,de} = \frac{\Delta}{2\sqrt{3}}$$

In reality, the quantization noise is of course not independent of the signal; this dependence is the source of idle tones and pattern noise in Sigma-Delta converters.

Oversampling ratio, where f_s is the sampling frequency and $2f_0$ is Nyquist rate

$$\mathrm{OSR} = \frac{f_s}{2f_0} = \frac{1}{2f_0\tau}$$

The rms noise voltage within the band of interest can be expressed in terms of OSR

$$\mathrm{n}_0 = \frac{e_{rms}}{\sqrt{OSR}}$$

Oversampling

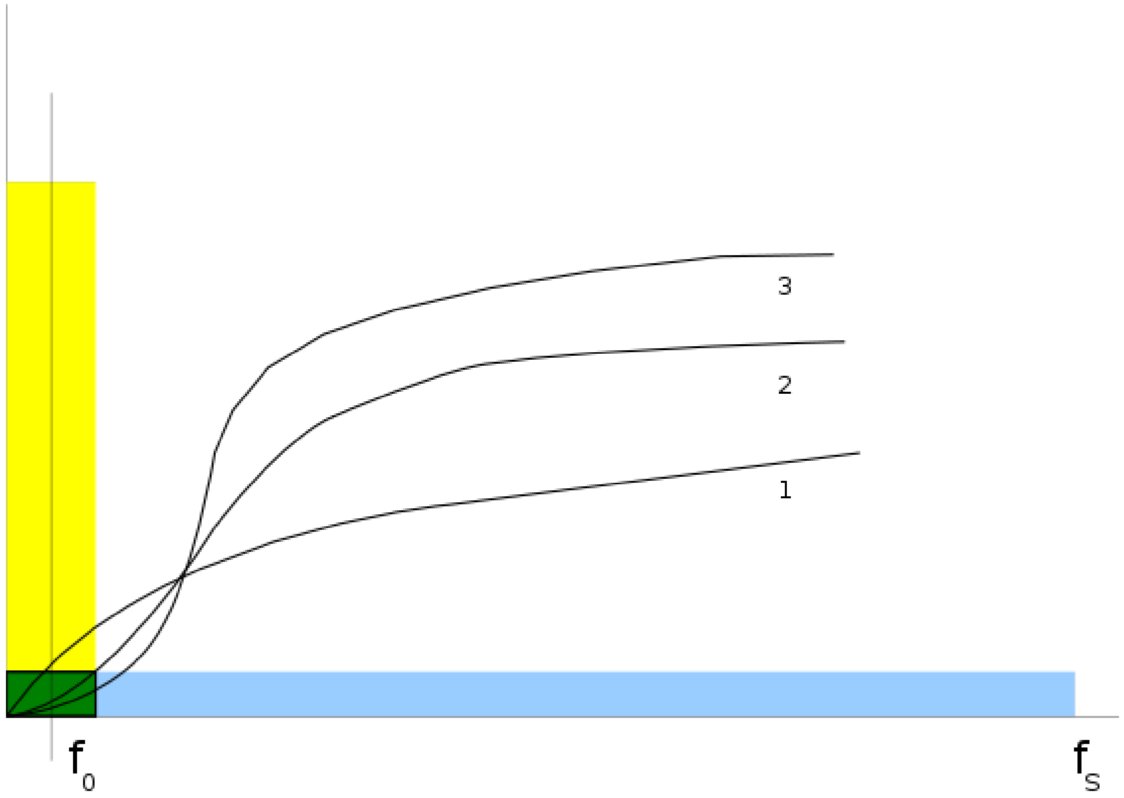


Fig. 5: Noise shaping curves and noise spectrum in ΔΣ modulator

Let's consider a signal at frequency $f_0$and a sampling frequency of $f_s$much higher than Nyquist rate (see fig. 5). ΔΣ modulation is based on the technique of oversampling to reduce the noise in the band of interest (green), which also avoids the use of high-precision analog circuits for the *anti-aliasing* filter. The quantization noise is the same both in a Nyquist converter (in yellow) and in an oversampling converter (in blue), but it is distributed over a larger spectrum. In ΔΣ-converters, noise is further reduced at low frequencies, which is the band where the signal of interest is, and it is increased at the higher frequencies, where it can be filtered. This technique is known as noise shaping.

For a first order delta sigma modulator, the noise is shaped by a filter with transfer function $H_n(z) = \left[1 - z^{-1}\right]$. Assuming that the sampling frequency $f_s \gg f_0$, the quantization noise in the desired signal bandwidth can be approximated as:

$$\mathrm{n}_0 = e_{rms} \frac{\pi}{\sqrt{3}} \left(2 f_0 \tau\right)^{\frac{3}{2}}$$

.

Similarly for a second order delta sigma modulator, the noise is shaped by a filter with transfer function $H_n(z) = \left[1 - z^{-1}\right]^2$. The in-band quantization noise can be approximated as:

$$\mathrm{n}_0 \; = \; e_{rms}\frac{\pi^2}{\sqrt{5}}\left(2f_0\tau\right)^{\frac{5}{2}}.$$

In general, for a N-order ΔΣ-modulator, the variance of the in-band quantization noise:

$$\mathrm{n}_0 \; = \; e_{rms}\frac{\pi^n}{\sqrt{2n+1}}\left(2f_0\tau\right)^{\frac{2n+1}{2}}.$$

When the sampling frequency is doubled, the signal to quantization noise is improved by $10\log(2^{2N+1})\ dB$for a N-order ΔΣ-modulator. The higher the oversampling ratio, the higher the signal-to-noise ratio and the higher the resolution in bits.

Another key aspect given by oversampling is the speed/resolution tradeoff. In fact, the decimation filter put after the modulator not only filters the whole sampled signal in the band of interest (cutting the noise at higher frequencies), but also reduces the frequency of the signal increasing its resolution. This is obtained by a sort of *averaging* of the higher data rate bitstream.

Example of decimation

Let's have, for instance, an 8:1 decimation filter and a 1-bit bitstream; if we have an input stream like 10010110, counting the number of ones, the decimation result is 4/8 = 0.5 = 100b (binary); in other words,

- the sample frequency is reduced by a factor of eight
- the serial (1-bit) input bus becomes a parallel (3-bits) output bus.

Naming

The technique was first presented in the early 1960s by prof. Haruhiko Yasuda while he was student at Waseda University, Tokyo, Japan, The name *Delta-Sigma* comes directly from the presence of a Delta modulator and an integrator, as firstly introduced by Inose et al. in their patent application. That is, the name comes from integrating or "*summing*" **differences**, which are operations usually associated with Greek letters Sigma and Delta respectively. Both names *Sigma-Delta* and *Delta-Sigma* are frequently used.

Chapter 7

Bilinear Time–Frequency Distribution

Bilinear time-frequency distributions, or **quadratic time-frequency distributions**, arise in a sub-field field of signal analysis and signal processing called time-frequency signal processing, and, in the statistical analysis of time series data. Such methods are used where one needs to deal with a situation where the frequency composition of a signal may be changing over time; this sub-field used to be called time-frequency signal analysis, and is now more often called time-frequency signal processing due to the progress in using these methods to a wide range of signal processing problems.

Background

Methods for analysing time series, in both signal analysis and time series analysis, have been developed as essentially separate methodologies applicable to, and based in, either the time or the frequency domain. A mixed approach is required in time-frequency analysis techniques which are especially effective in analyzing non-stationary signals, whose frequency distribution and magnitude vary with time. Examples of these are acoustic signals. Classes of "quadratic time-frequency distributions" (or bilinear time-frequency distributions") are used for time-frequency signal analysis. This class is similar in formulation to Cohen's class distribution function that was used in 1966 in the context of quantum mechanics. This distribution function is mathematically similar to a generalized time-frequency representation which utilizes bilinear transformations. Compared with other time-frequency analysis techniques, such as short-time Fourier transform (STFT), the bilinear-transformation (or Quadratic Time-Frequency Distributions) may not have higher clarity for most practical signals, but it provides an alternative framework to investigate new definitions and new methods. While it does suffer from an inherent cross-term contamination when analyzing multi-component signals, by using a carefully chosen window functions, the interference can be significantly mitigated, at the expense of resolution.

Mathematical definition

The definition of the class of bilinear (or quadratic) time-frequency distributions is as follows:

$$C_x(t,f)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}A_x(\eta,\tau)\Phi(\eta,\tau)\exp(-j2\pi(\eta t+\tau f))\,d\eta\,d\tau,$$

where $A_x\left(\eta,\tau\right)$is the ambiguity function (AF) which will be further discussed later, and $\Phi\left(\eta,\tau\right)$is the kernel function which is usually a low-pass function and is used to mask out the interference.

Ambiguity function

The class of bilinear (or quadratic) time-frequency distributions can be most easily understood in terms of the ambiguity function an explanation of which follows.

Consider the well known power spectral density $P_x\left(f\right)$and the signal auto-correlation function $R_x\left(\tau\right)$in the case of a stationary process. The relationship between these functions is as follows:

$$P_x(f)=\int_{-\infty}^{\infty}R_x(\tau)e^{-j2\pi f\tau}\,d\tau,$$
$$R_x(\tau)=E\left[x(t+\tau/2)x(t-\tau/2)\right].$$

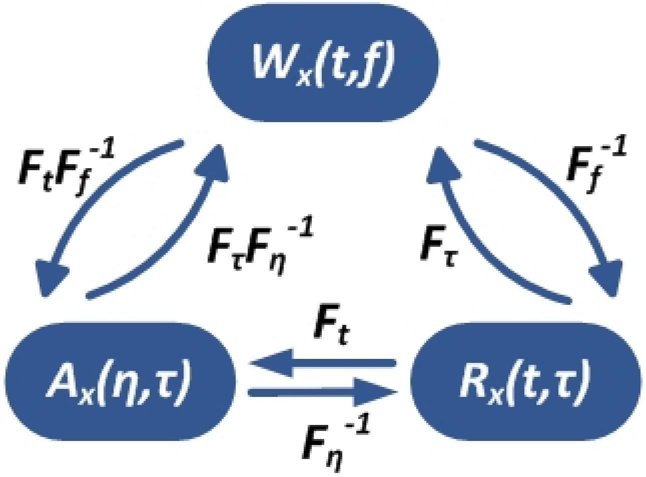
For a non-stationary signal $x\left(t\right)$, these relations can be generalized using a time-dependent power spectral density or equivalently the famous Wigner distribution function of $x\left(t\right)$as follows:

$$W_x(t,f)=\int_{-\infty}^{\infty}R_x(t,\tau)e^{-j2\pi f\tau}\,d\tau,$$
$$R_x\left(t,\tau\right)=x(t+\tau/2)*x(t-\tau/2).$$

If the Fourier transform of the auto-correlation function is taken with respect to *t* instead of τ, we get the ambiguity function as follows:

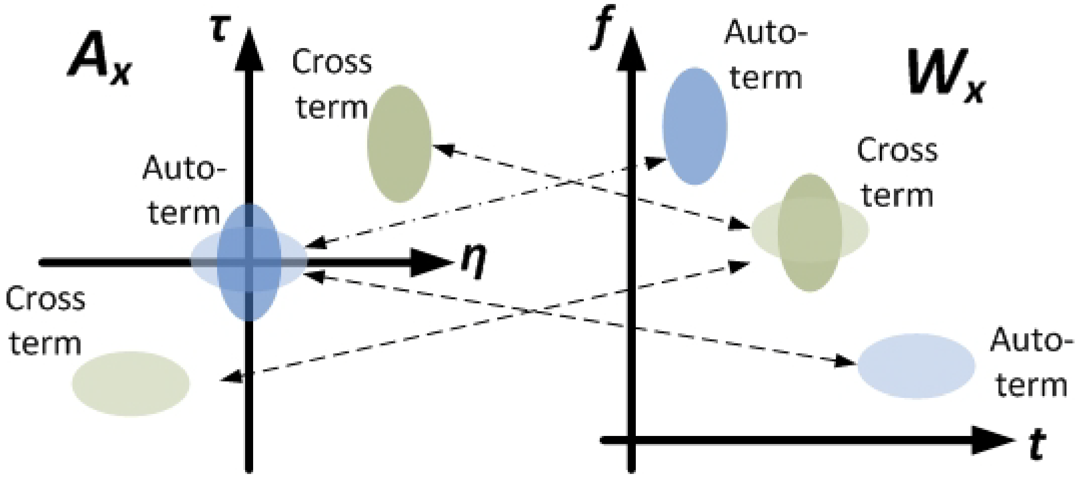
$$A_x(\eta,\tau)=\int_{-\infty}^{\infty}x(t+\tau/2)x^*(t-\tau/2)e^{-j2\pi t\eta}\,dt.$$

The relationship between the Wigner distribution function, the auto-correlation function and the ambiguity function can then be illustrated by the following figure.

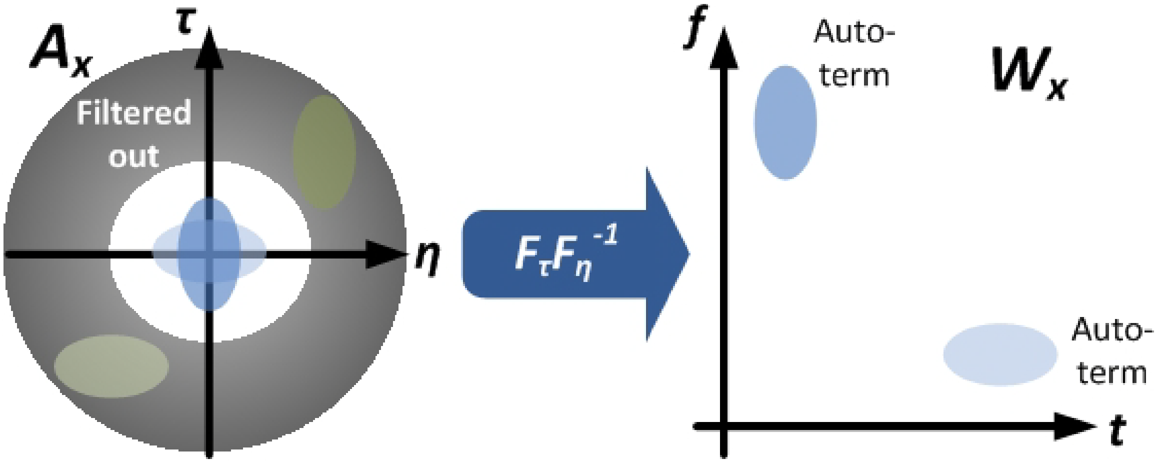


By comparing the definition of bilinear (or quadratic) time-frequency distributions with that of the Wigner distribution function, it is easily found that the latter is a special case of the former with $\Phi(\eta,\tau)=1$. Alternatively, bilinear (or quadratic) time-frequency distributions can be regarded as a masked version of the Wigner distribution function if a kernel function $\Phi(\eta,\tau)\neq 1$is chosen. A properly chosen kernel function can significantly reduce the undesirable cross-term of the Wigner distribution function.

What is the benefit of the additional kernel function? The following figure shows the distribution of the auto-term and the cross-term of a multi-component signal in both the ambiguity and the Wigner distribution function.



For multi-component signals in general, the distribution of its auto-term and cross-term within its Wigner distribution function is generally not predictable, and hence the cross-term cannot be removed easily. However, as shown in the figure, for the ambiguity function, the auto-term of the multi-component signal will inherently tend to close the origin in the η,τ plane, and the cross-term will tend to be away from the origin. With this property, the cross-term in can be filtered out effortlessly if a proper low-pass kernel function is applied in η,τ domain. The following is an example that demonstrates how the cross-term is filtered out.



Some time-frequency distributions

Wigner distribution function

Aforementioned, the Wigner distribution function is a member of the class of quadratic time-frequency distributions (QTFDs)with the kernel function $\Phi\left(\eta,\tau\right)=1$. The definition of Wigner distribution is as follows:

$$W_x(t,f)=\int_{-\infty}^{\infty}x(t+\tau/2)*x(t-\tau/2)e^{-j2\pi f\tau}\,d\tau.$$

Choi–Williams distribution function

The kernel of Choi–Williams distribution is defined as follows:

$$\Phi\left(\eta,\tau\right)=\exp\left[-\alpha\left(\eta\tau\right)^2\right],$$

where α is an adjustable parameter.

Cone-shape distribution function

The kernel of cone-shape distribution function is defined as follows:

$$\Phi\left(\eta,\tau\right)=\frac{\sin\left(\pi\eta\tau\right)}{\pi\eta\tau}\exp\left(-2\pi\alpha\tau^2\right),$$

where α is an adjustable parameter.

There are many more such QTFDs and a full list can be found in the next reference. .

Chapter 8

Delta Modulation

Delta modulation (DM or Δ-modulation)is an analog-to-digital and digital-to-analog signal conversion technique used for transmission of voice information where quality is not of primary importance. DM is the simplest form of differential pulse-code modulation (DPCM) where the difference between successive samples is encoded into n-bit data streams. In delta modulation, the transmitted data is reduced to a 1-bit data stream. Its main features are:

- the analog signal is approximated with a series of segments
- each segment of the approximated signal is compared to the original analog wave to determine the increase or decrease in relative amplitude
- the decision process for establishing the state of successive bits is determined by this comparison
- only the change of information is sent, that is, only an increase or decrease of the signal amplitude from the previous sample is sent whereas a no-change condition causes the modulated signal to remain at the same 0 or 1 state of the previous sample.

To achieve high signal-to-noise ratio, delta modulation must use oversampling techniques, that is, the analog signal is sampled at a rate several times higher than the Nyquist rate.

Derived forms of delta modulation are continuously variable slope delta modulation, delta-sigma modulation, and differential modulation. The Differential Pulse Code Modulation is the super set of DM.

Principle

Rather than quantizing the absolute value of the input analog waveform, delta modulation quantizes the difference between the current and the previous step, as shown in the block diagram in Fig. 1.

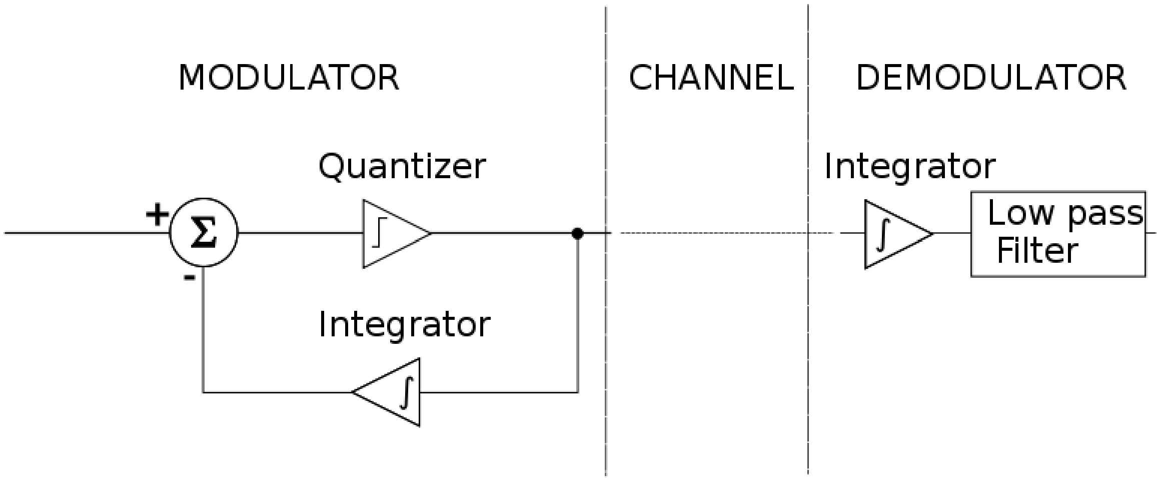


Fig. 1 - Block diagram of a Δ-modulator/demodulator

The modulator is made by a quantizer which converts the difference between the input signal and the average of the previous steps. In its simplest form, the quantizer can be realized with a comparator referenced to 0 (two levels quantizer), whose output is *1* or *0* if the input signal is positive or negative. It is also a bit-quantizer as it quantizes only a bit at a time. The demodulator is simply an integrator (like the one in the feedback loop) whose output rises or falls with each 1 or 0 received. The integrator itself constitutes a low-pass filter.

Transfer characteristics

The transfer characteristics of a delta modulated system follows a signum function,as it quantizes only two levels and also one-bit at a time.

The two sources of noise in delta modulation are "slope overload", when steps are too small to track the original waveform, and "granularity", when steps are too large. But a 1971 study shows that slope overload is less objectionable compared to granularity than one might expect based solely on SNR measures.

Output signal power

In delta modulation there is no restriction on the amplitude of the signal waveform, because the number of levels is not fixed. On the other hand, there is a limitation on the slope of the signal waveform which must be observed if slope overload is to be avoided.

However, if the signal waveform changes slowly, there is nominally no limit to the signal power which may be transmitted.

Bit-rate

If the communication channel is of limited bandwidth,there is the possibility of interference in either DM or PCM.Hence,'DM' and 'PCM' operate at same bit-rate.

Adaptive delta modulation

Adaptive delta modulation (ADM) or continuously variable slope delta modulation (CVSD) is a modification of DM in which the step size is not fixed. Rather, when several consecutive bits have the same direction value, the encoder and decoder assume that slope overload is occurring, and the step size becomes progressively larger. Otherwise, the step size becomes gradually smaller over time. ADM reduces slope error,at the expense of increasing quantizing error.This error can be reduced by using a low pass filter.

Comparison of PCM and DM

Signal-to-noise ratio of DM is larger than signal-to-noise ratio of PCM. For an ADM signal-to-noise ratio is comparable to Signal-to-noise ratio of companded PCM.

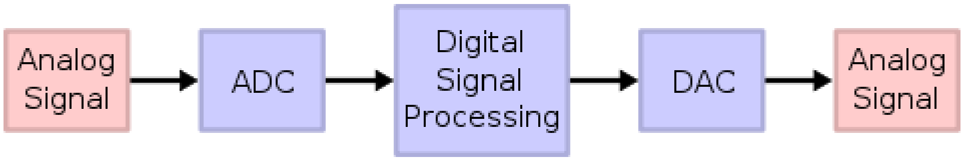
Chapter 9

Digital Signal Processor

A **digital signal processor** (**DSP**) is a specialized microprocessor with an optimized architecture for the fast operational needs of digital signal processing.

Typical characteristics

Digital signal processing algorithms typically require a large number of mathematical operations to be performed quickly and repetitively on a set of data. Signals (perhaps from audio or video sensors) are constantly converted from analog to digital, manipulated digitally, and then converted again to analog form, as diagrammed below. Many DSP applications have constraints on latency; that is, for the system to work, the DSP operation must be completed within some fixed time, and deferred (or batch) processing is not viable.



A simple digital processing system

Most general-purpose microprocessors and operating systems can execute DSP algorithms successfully, but are not suitable for use in portable devices such as mobile phones and PDAs because of power supply and space constraints. A specialized digital signal processor, however, will tend to provide a lower-cost solution, with better performance, lower latency, and no requirements for specialized cooling or large batteries.

The architecture of a digital signal processor is optimized specifically for digital signal processing. Most also support some of the features as an applications processor or microcontroller, since signal processing is rarely the only task of a system. Some useful features for optimizing DSP algorithms are outlined below.

Architecture

By the standards of general purpose processors, DSP instruction sets are often highly irregular. One implication for software architecture is that hand-optimized assembly is commonly packaged into libraries for re-use, instead of relying on unusually advanced compiler technologies to handle essential algorithms.

Hardware features visible through DSP instruction sets commonly include:

- Hardware modulo addressing, allowing circular buffers to be implemented without having to constantly test for wrapping.
- A memory architecture designed for streaming data, using DMA extensively and expecting code to be written to know about cache hierarchies and the associated delays.
- Driving multiple arithmetic units may require memory architectures to support several accesses per instruction cycle
- Separate program and data memories (Harvard architecture), and sometimes concurrent access on multiple data busses
- Special SIMD (single instruction, multiple data) operations
- Some processors use VLIW techniques so each instruction drives multiple arithmetic units in parallel
- Special arithmetic operations, such as fast multiply-accumulates (MACs). Many fundamental DSP algorithms, such as FIR filters or the Fast Fourier transform (FFT) depend heavily on multiply-accumulate performance.
- Bit-reversed addressing, a special addressing mode useful for calculating FFTs
- Special loop controls, such as architectural support for executing a few instruction words in a very tight loop without overhead for instruction fetches or exit testing
- Deliberate exclusion of a memory management unit. DSPs frequently use multi-tasking operating systems, but have no support for virtual memory or memory protection. Operating systems that use virtual memory require more time for context switching among processes, which increases latency.

Program flow

- Floating-point unit integrated directly into the datapath
- Pipelined architecture
- Highly parallel multiplier–accumulators (MAC units)
- Hardware-controlled looping, to reduce or eliminate the overhead required for looping operations

Memory architecture

- DSPs often use special memory architectures that are able to fetch multiple data and/or instructions at the same time:
  - Harvard architecture
  - Modified von Neumann architecture
- Use of direct memory access
- Memory-address calculation unit

Data operations

- Saturation arithmetic, in which operations that produce overflows will accumulate at the maximum (or minimum) values that the register can hold rather than wrapping around (maximum+1 doesn't overflow to minimum as in many general-purpose CPUs, instead it stays at maximum). Sometimes various sticky bits operation modes are available.
- Fixed-point arithmetic is often used to speed up arithmetic processing
- Single-cycle operations to increase the benefits of pipelining

Instruction sets

- Multiply-accumulate (MAC, aka fused multiply-add, FMA) operations, which are used extensively in all kinds of matrix operations, such as convolution for filtering, dot product, or even polynomial evaluation
- Instructions to increase parallelism: SIMD, VLIW, superscalar architecture
- Specialized instructions for modulo addressing in ring buffers and bit-reversed addressing mode for FFT cross-referencing
- Digital signal processors sometimes use time-stationary encoding to simplify hardware and increase coding efficiency.

History

Prior to the advent of stand-alone DSP chips discussed below, most DSP applications were implemented using bit-slice processors. The AMD 2901 bit-slice chip with its family of components was a very popular choice. There were reference designs from AMD, but very often the specifics of a particular design were application specific. These bit slice architectures would sometimes include a peripheral multiplier chip. Examples of these multipliers were a series from TRW including the TDC1008 and TDC1010, some of which included an accumulator, providing the requisite multiply-accumulate (MAC) function.

In 1978, Intel released the 2920 as an "analog signal processor". It had an on-chip ADC/DAC with an internal signal processor, but it didn't have a hardware multiplier and was not successful in the market. In 1979, AMI released the S2811. It was designed as a microprocessor peripheral, and it had to be initialized by the host. The S2811 was likewise not successful in the market.

In 1980 the first stand-alone, complete DSPs – the NEC µPD7720 and AT&T DSP1 – were presented at the International Solid-State Circuits Conference '80. Both processors were inspired by the research in PSTN telecommunications.

The Altamira DX-1 was another early DSP, utilizing quad integer pipelines with delayed branches and branch prediction.

The first DSP produced by Texas Instruments (TI), the TMS32010 presented in 1983, proved to be an even bigger success. It was based on the Harvard architecture, and so had separate instruction and data memory. It already had a special instruction set, with instructions like load-and-accumulate or multiply-and-accumulate. It could work on 16-bit numbers and needed 390 ns for a multiply-add operation. TI is now the market leader in general-purpose DSPs. Another successful design was the Motorola 56000.

About five years later, the second generation of DSPs began to spread. They had 3 memories for storing two operands simultaneously and included hardware to accelerate tight loops, they also had an addressing unit capable of loop-addressing. Some of them operated on 24-bit variables and a typical model only required about 21 ns for a MAC (multiply-accumulate). Members of this generation were for example the AT&T DSP16A or the Motorola DSP56001.

The main improvement in the third generation was the appearance of application-specific units and instructions in the data path, or sometimes as coprocessors. These units allowed direct hardware acceleration of very specific but complex mathematical problems, like the Fourier-transform or matrix operations. Some chips, like the Motorola MC68356, even included more than one processor core to work in parallel. Other DSPs from 1995 are the TI TMS320C541 or the TMS 320C80.

The fourth generation is best characterized by the changes in the instruction set and the instruction encoding/decoding. SIMD extensions were added, VLIW and the superscalar architecture appeared. As always, the clock-speeds have increased, a 3 ns MAC now became possible.

Modern DSPs

Modern signal processors yield greater performance; this is due in part to both technological and architectural advancements like lower design rules, fast-access two-level cache, (E)DMA circuitry and a wider bus system. Not all DSP's provide the same speed and many kinds of signal processors exist, each one of them being better suited for a specific task, ranging in price from about US$1.50 to US$300

Texas Instruments produce the C6000 series DSP’s, which have clock speeds of 1.2 GHz and implement separate instruction and data caches. They also have an 8 MiB 2nd level cache and 64 EDMA channels. The top models are capable of as many as 8000 MIPS (instructions per second), use VLIW (very long instruction word), perform eight operations per clock-cycle and are compatible with a broad range of external peripherals and various buses (PCI/serial/etc). TMS320C6474 chips each have three such DSP's, and the newest generation C6000 chips support floating point as well as fixed point processing.

Freescale produce a multi-core DSP family, the MSC81xx. The MSC81xx is based on StarCore Architecture processors and the latest MSC8144 DSP combines four programmable SC3400 StarCore DSP cores. Each SC3400 StarCore DSP core has a clock speed of 1 GHz.

Analog Devices produce the SHARC-based DSP and range in performance from 66 MHz/198 MFLOPS (million floating-point operations per second) to 400 MHz/2400 MFLOPS. Some models support multiple multipliers and ALUs, SIMD instructions and audio processing-specific components and peripherals. The Blackfin family of embedded digital signal processors combine the features of a DSP with those of a general use processor. As a result, these processors can run simple operating systems like μCLinux, velOSity and Nucleus RTOS while operating on real-time data.

NXP Semiconductors produce DSP's based on TriMedia VLIW technology, optimized for audio and video processing. In some products the DSP core is hidden as a fixed-function block into a SoC, but NXP also provides a range of flexible single core media processors. The TriMedia media processors support both fixed-point arithmetic as well as floating-point arithmetic, and have specific instructions to deal with complex filters and entropy coding.

Most DSP's use fixed-point arithmetic, because in real world signal processing the additional range provided by floating point is not needed, and there is a large speed benefit and cost benefit due to reduced hardware complexity. Floating point DSP's may be invaluable in applications where a wide dynamic range is required. Product developers might also use floating point DSP's to reduce the cost and complexity of software development in exchange for more expensive hardware, since it is generally easier to implement algorithms in floating point.

Generally, DSP's are dedicated integrated circuits; however DSP functionality can also be produced by using field-programmable gate array chips (FPGA’s).

Embedded general-purpose RISC processors are becoming increasingly DSP like in functionality. For example, the ARM Cortex-A8 and the OMAP3 processors include a Cortex-A8 and C6000 DSP.

Chapter 10

Dither

Dither is an intentionally applied form of noise used to randomize quantization error, preventing large-scale patterns such as "banding" in images. Dither is routinely used in processing of both digital audio and digital video data, and is often one of the last stages of audio production to compact disc.

Etymology

…one of the earliest [applications] of dither came in World War II. Airplane bombers used mechanical computers to perform navigation and bomb trajectory calculations. Curiously, these computers (boxes filled with hundreds of gears and cogs) performed more accurately when flying on board the aircraft, and less well on ground. Engineers realized that the vibration from the aircraft reduced the error from sticky moving parts. Instead of moving in short jerks, they moved more continuously. Small vibrating motors were built into the computers, and their vibration was called dither from the Middle English verb "didderen," meaning "to tremble." Today, when you tap a mechanical meter to increase its accuracy, you are applying dither, and modern dictionaries define dither as a highly nervous, confused, or agitated state. In minute quantities, dither successfully makes a digitization system a little more analog in the good sense of the word.
—Ken Pohlmann, *Principles of Digital Audio*

The term "dither" was published in books on analog computation and hydraulic controlled guns shortly after the war. The concept of dithering to reduce quantization patterns was first applied by Lawrence G. Roberts in his 1961 MIT master's thesis and 1962 article though he did not use the term *dither*. By 1964 dither was being used in the modern sense.

In digital processing and waveform analysis

Dither is often used in digital audio and video processing, where it is applied to bit-depth transitions; it is utilized in many different fields where digital processing and analysis are used — especially waveform analysis. These uses include systems using digital signal processing, such as digital audio, digital video, digital photography, seismology, RADAR, weather forecasting systems and many more.

The premise is that quantization and re-quantization of digital data yields error. If that error is repeating and *correlated* to the signal, the error that results is repeating, cyclical, and mathematically determinable. In some fields, especially where the receptor is sensitive to such artifacts, cyclical errors yield undesirable artifacts. In these fields dither results in less determinable artifacts. The field of audio is a primary example of this — the human ear functions much like a Fourier transform, wherein it hears individual frequencies. The ear is therefore very sensitive to *distortion,* or additional frequency content that "colors" the sound differently, but far less sensitive to random noise at all frequencies.

Digital audio

In audio, dither can be useful to break up periodic limit cycles, which are a common problem in digital filters. Random noise is typically less objectionable than the harmonic tones produced by limit cycles.

In 1987, Lipshitz and Vanderkooy pointed out that different noise types, with different probability density functions, behave differently when used as dither signals, and suggested optimal levels of dither signals for audio.

In an analog system, the signal is *continuous*, but in a PCM digital system, the amplitude of the signal out of the digital system is limited to one of a set of fixed values or numbers. This process is called quantization. Each coded value is a discrete step... if a signal is quantized without using dither, there will be quantization distortion related to the original input signal... In order to prevent this, the signal is "dithered", a process that mathematically removes the harmonics or other highly undesirable distortions entirely, and that replaces it with a constant, fixed noise level.

The final version of audio that goes onto a compact disc contains only 16 bits per sample, but throughout the production process a greater number of bits are typically used to represent the sample. In the end, the digital data must be reduced to 16 bits for pressing onto a CD and distributing.

There are multiple ways to do this. One can, for example, simply discard the excess bits — called *truncation.* One can also *round* the excess bits to the nearest value. Each of these methods, however, results in predictable and determinable errors in the result. Take, for example, a waveform that consists of the following values:

```
1    2    3    4    5    6    7    8
```

If we reduce our waveform by, say, 20% then we end up with the following values:

```
0.8 1.6 2.4 3.2 4.0 4.8 5.6 6.4
```

If we truncate these values we end up with the following data:

```
0    1    2    3    4    4    5    6
```

If we instead round these values we end up with the following data:

```
1    2    2    3    4    5    6    6
```

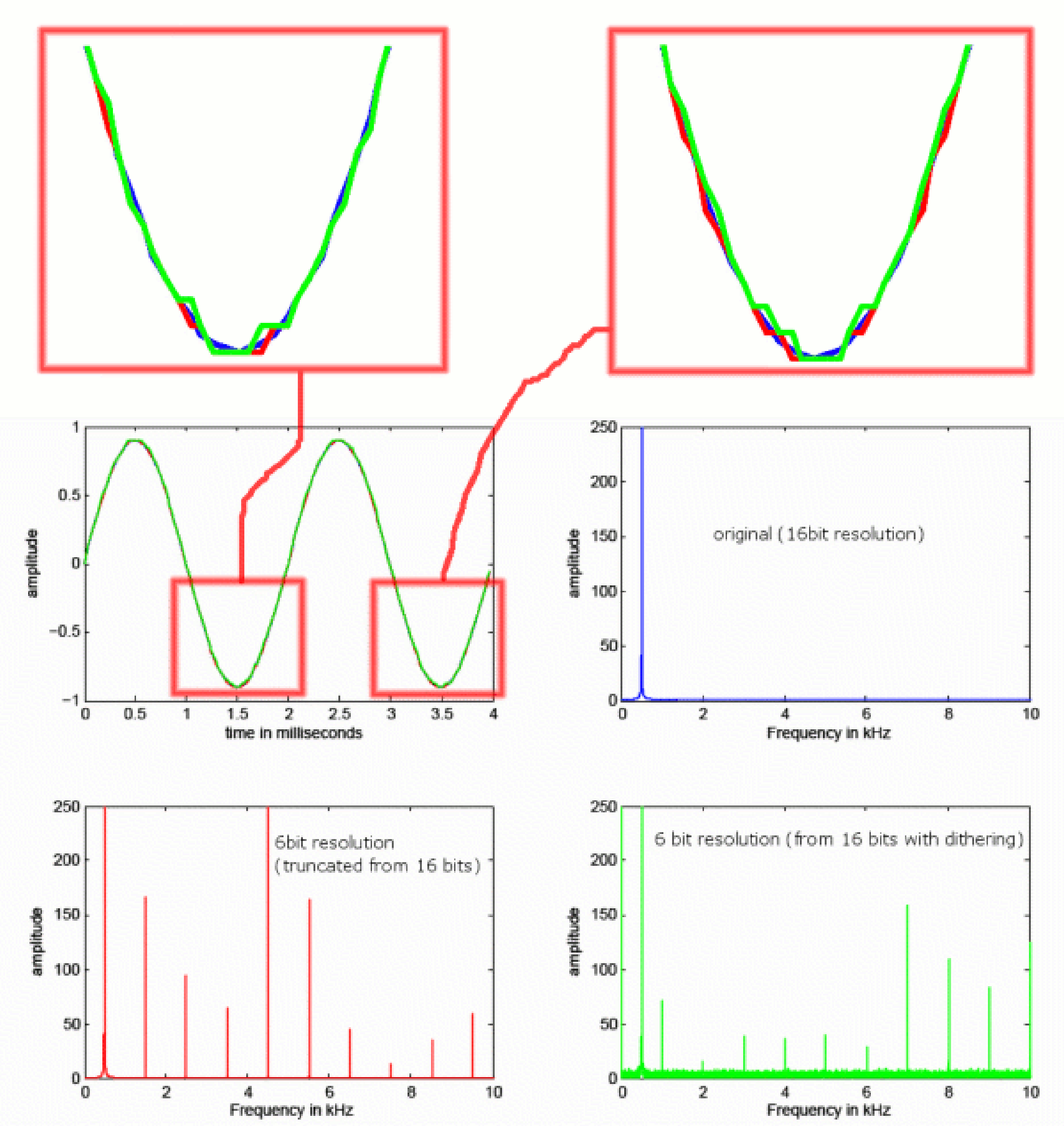
For any original waveform, the process of reducing the waveform amplitude by 20% results in regular errors. Take for example a sine wave that, for some portion, matches the values above. Every time the sine wave's value hit "3.2," the truncated result would be off by 0.2, as in the sample data above. Every time the sine wave's value hit "4.0," there would be no error since the truncated result would be off by 0.0, also shown above. The magnitude of this error changes regularly and repeatedly throughout the sine wave's cycle. It is precisely this error which manifests itself as distortion. What the ear hears as distortion is the additional content at discrete frequencies created by the regular and repeated quantization error.

A plausible solution would be to take the 2 digit number (say, 4.8) and round it one direction or the other. For example, we could round it to 5 one time and then 4 the next time. This would make the long-term average 4.5 instead of 4, so that over the long-term the value is closer to its actual value. This, on the other hand, still results in determinable (though more complicated) error. Every other time the value 4.8 comes up the result is an error of 0.2, and the other times it is –0.8. This still results in repeating, quantifiable error.

Another plausible solution would be to take 4.8 and round it so that the first four times out of five it rounded up to 5, and the fifth time it rounded to 4. This would average out to exactly 4.8 over the long term. Unfortunately, however, it still results in repeatable and determinable errors, and those errors still manifest themselves as distortion to the ear (though oversampling can reduce this).

This leads to the *dither* solution. Rather than predictably rounding up or down in a repeating pattern, what if we rounded up or down in a random pattern? If we came up with a way to randomly toggle our results between 4 and 5 so that 80% of the time it ended up on 5 then we would average 4.8 over the long run but would have random, unrepeating error in the result. This is done through dither.

We calculate a series of random numbers between 0.0 and 0.9 (ex: 0.6, 0.1, 0.3, 0.6, 0.9, etc.) and we add these random numbers to the results of our equation. Two times out of ten the result will truncate back to 4 (if 0.0 or 0.1 are added to 4.8) and the rest of the times it will truncate to 5, but each given situation has a random 20% chance of rounding to 4 or 80% chance of rounding to 5. Over the long haul this will result in results that average to 4.8 and a quantization error that is random — or noise. This "noise" result is less offensive to the ear than the determinable distortion that would result otherwise.



Reducing amplitude resolution of a 500Hz sine wave from 16 to 6 bits:

The Blue spectrum shows the original sine at 500Hz.
Truncating to 6 bits introduces harmonics/distortion (multiples of 500Hz) - red spectrum
Dithering reduces the amplitude of these distortions, but introduces background noise - green spectrum
(please note that the spectral plots above have been clipped at 250. The amplitude at 500Hz has thus been clipped and is actually much larger than can be seen here)

The sine wave at the top shows that truncation (red) always rounds values the same way
while dithering randomizes the choice of rounding up or down (green)

Audio samples:

Usage

Dither should be added to any low-amplitude or highly-periodic signal before any quantization or re-quantization process, in order to de-correlate the quantization noise with the input signal and to prevent non-linear behavior (distortion); the lesser the bit depth, the greater the dither must be. The results of the process still yield distortion, but the distortion is of a random nature so its result is effectively noise. Any bit-reduction process should add dither to the waveform before the reduction is performed.

Different types

RPDF stands for "Rectangular Probability Density Function," equivalent to a roll of a die. Any number has the same random probability of surfacing.

TPDF stands for "Triangular Probability Density Function," equivalent to a roll of two dice (the sum of two independent samples of RPDF).

Gaussian PDF is equivalent to a roll of a large number of dice. The relationship of probabilities of results follows a bell-shaped, or Gaussian curve, typical of dither generated by analog sources such as microphone preamplifiers. If the bit depth of a recording is sufficiently great, that noise will be sufficient to dither the recording.

Colored Dither is sometimes mentioned as dither that has been filtered to be different from white noise. Some dither algorithms use noise that has more energy in the higher frequencies so as to lower the energy in the critical audio band.

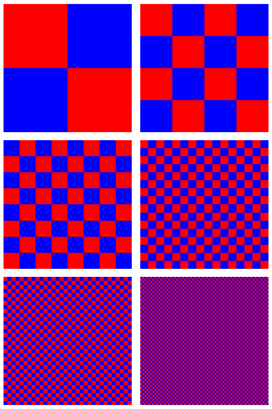
Noise shaping is a filtering process that shapes the spectral energy of quantisation error, typically to either de-emphasise frequencies to which the ear is most sensitive or separate the signal and noise bands completely. If dither is used, its final spectrum depends on whether it is added inside or outside the feedback loop of the noise shaper: if inside, the dither is treated as part of the error signal and shaped along with actual quantisation error; if outside, the dither is treated as part of the original signal and linearises quantisation without being shaped itself. In this case, the final noise floor is the sum of the flat dither spectrum and the shaped quantisation noise. While real-world noise shaping usually includes in-loop dithering, it is also possible to use it without adding dither at all, in which case the usual harmonic-distortion effects still appear at low signal levels.

Which types to use

If the signal being dithered is to undergo further processing, then it should be processed with TPDF dither that has an amplitude of two quantization steps (so that the dither values computed range from, say, –1 to +1, or 0 to 2). This is the lowest power ideal dither, in that it does not introduce noise modulation (constant noise floor) and completely eliminates the harmonic distortion from *quantization*. If colored dither is used at these intermediate processing stages then the frequency content can "bleed" into other, more noticeable frequency ranges and become distractingly audible.

If the signal being dithered is to undergo no further processing — it is being dithered to its final result for distribution — then colored dither or noise shaping is appropriate, and can effectively lower the audible noise level by putting most of that noise in a frequency range where it is less critical.

Digital photography and image processing



An illustration of dithering. Red and blue are the only colors used, but as the pixels become smaller, the patch appears violet.

Dithering is a technique used in computer graphics to create the illusion of color depth in images with a limited color palette (color quantization). In a dithered image, colors not available in the palette are approximated by a diffusion of colored pixels from within the available palette. The human eye perceives the diffusion as a mixture of the colors within it. Dithering is analogous to the halftone technique used in printing. Dithered images, particularly those with relatively few colors, can often be distinguished by a characteristic graininess, or speckled appearance.

Examples

Reducing the color depth of an image can often have significant visual side-effects. If the original image is a photograph, it is likely to have thousands, or even millions of distinct colors. The process of constraining the available colors to a specific *color palette* effectively throws away a certain amount of color information.

A number of factors can affect the resulting quality of a color-reduced image. Perhaps most significant is the color palette that will be used in the reduced image. For example, an original image (*Figure 1*) may be reduced to the 216-color "web-safe" color palette. If the original pixel colors are simply translated into the closest available color from the palette, no dithering occurs (*Figure 2*). Typically, this approach results in flat areas (contours) and a loss of detail, and may produce patches of color that are significantly different from the original. Shaded or gradient areas may appear as *color bands*, which may be distracting. The application of dithering can help to minimize such visual artifacts, and usually results in a better representation of the original (*Figure 3*). Dithering helps to reduce color banding and flatness.

One of the problems associated with using a fixed color palette is that many of the needed colors may not be available in the palette, and many of the available colors may not be needed; a fixed palette containing mostly shades of green would not be well-suited for images that do not contain many shades of green, for instance. The use of an optimized color palette can be of benefit in such cases. An optimized color palette is one in which the available colors are chosen based on how frequently they are used in the original source image. If the image is reduced based on an optimized palette, the result is often much closer to the original (*Figure 4*).

The number of colors available in the palette is also a contributing factor. If, for example, the palette is limited to only 16 colors, the resulting image could suffer from additional loss of detail, and even more pronounced problems with flatness and color banding (*Figure 5*). Once again, dithering can help to minimize such artifacts (*Figure 6*).

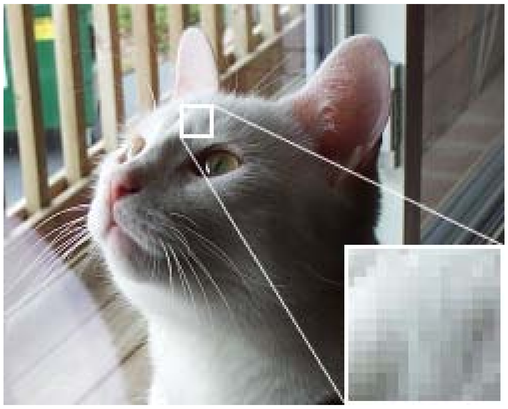


Figure 1. Original photo; note the smoothness in the detail.

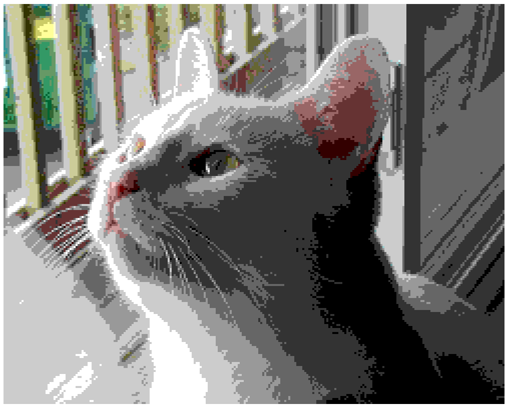


Figure 2. Original image using the web-safe color palette with no dithering applied. Note the large flat areas and loss of detail.

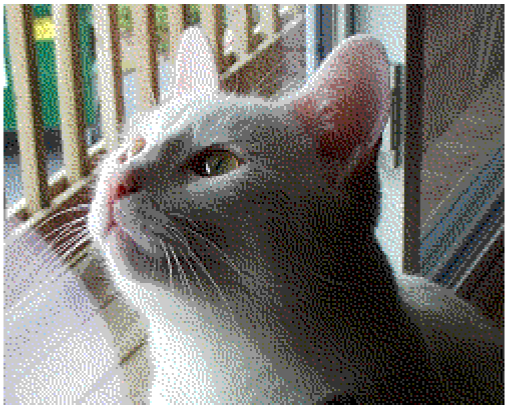


Figure 3. Original image using the web-safe color palette with Floyd–Steinberg dithering. Note that even though the same palette is used, the application of dithering gives a better representation of the original.

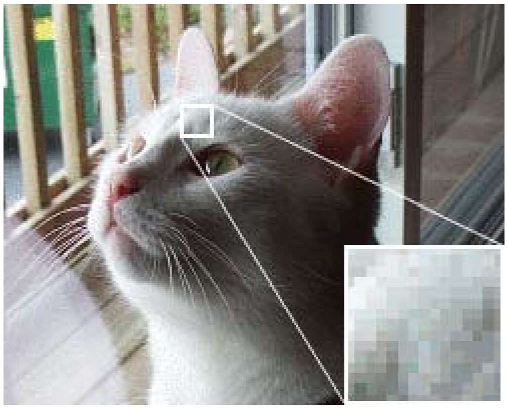


Figure 4. Here, the original has been reduced to a 256-color optimized palette with Floyd–Steinberg dithering applied. The use of an optimized palette, rather than a fixed palette, allows the result to better represent the colors in the original image.

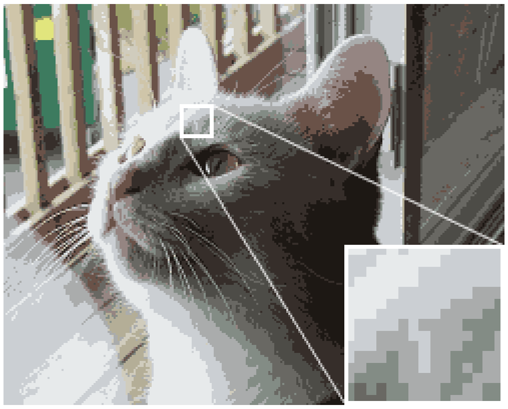


Figure 5. Depth is reduced to a 16-color optimized palette in this image, with no dithering. Colors appear muted, and color banding is pronounced.

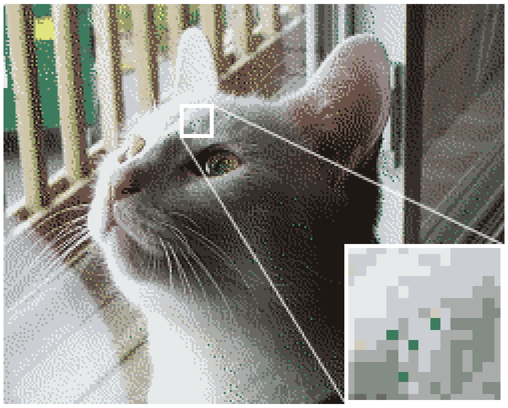


Figure 6. This image also uses the 16-color optimized palette, but the use of dithering helps to reduce banding.

Applications

Display hardware, including early computer video adapters and many modern LCDs used in mobile phones and inexpensive digital cameras, show a much smaller color range than more advanced displays. One common application of dithering is to more accurately display graphics containing a greater range of colors than the hardware is capable of showing. For example, dithering might be used in order to display a photographic image containing millions of colors on video hardware that is only capable of showing 256 colors at a time. The 256 available colors would be used to generate a dithered approximation of the original image. Without dithering, the colors in the original image might simply be "rounded off" to the closest available color, resulting in a new image that is a poor representation of the original. Dithering takes advantage of the human eye's tendency to "mix" two colors in close proximity to one another.

Some LCDs may use **temporal dithering** to achieve a similar effect. By alternating each pixel's color value rapidly between two approximate colors in the panel's color space (also known as Frame Rate Control), a display panel which natively supports 18-bit color (6 bits per channel) can represent a 24-bit "true" color image (8 bits per channel).

Dithering such as this, in which the computer's display hardware is the primary limitation on color depth, is commonly employed in software such as web browsers. Since a web browser may be retrieving graphical elements from an external source, it may be necessary for the browser to perform dithering on images with too many colors for the available display. It was due to problems with dithering that a color palette known as the "web-safe color palette" was identified, for use in choosing colors that would not be dithered on displays with only 256 colors available.

But even when the total number of available colors in the display hardware is high enough when rendering full color digital photographs, as those 15- and 16-bit RGB Hicolor 32,768/65,536 color modes, banding can be evident to the eye, especially in large areas of smooth shade transitions (although the original image file has no banding at all). Dithering the 32 or 64 RGB levels will result in a pretty good "pseudo truecolor" display approximation, which the eye cannot resolve as *grainy*. Furthermore, images displayed on 24-bit RGB hardware (8 bits per RGB primary) can be dithered to simulate somewhat higher bit depth, and/or to minimize the loss of hues available after a gamma correction. High-end still image processing software, as Adobe Photoshop, commonly uses these techniques for improved display.

Another useful application of dithering is for situations in which the graphic file format is the limiting factor. In particular, the commonly-used GIF format is restricted to the use of 256 or fewer colors in many graphics editing programs. Images in other file formats, such as PNG, may also have such a restriction imposed on them for the sake of a reduction in file size. Images such as these have a fixed color palette defining all the colors that the image may use. For such situations, graphical editing software may be responsible for dithering images prior to saving them in such restrictive formats.

Algorithms

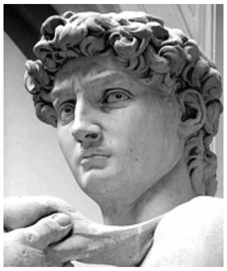
There are several algorithms designed to perform dithering. One of the earliest, and still one of the most popular, is the Floyd–Steinberg dithering algorithm, developed in 1975. One of the strengths of this algorithm is that it minimizes visual artifacts through an error-diffusion process; error-diffusion algorithms typically produce images that more closely represent the original than simpler dithering algorithms.

Dithering methods include:

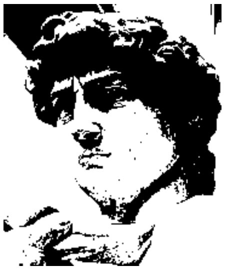
- *Thresholding* (also average dithering): each pixel value is compared against a fixed threshold. This may be the simplest dithering algorithm there is, but it results in immense loss of detail and contouring.
- *Random dithering* was the first attempt (at least as early as 1951) to remedy the drawbacks of thresholding. Each pixel value is compared against a random threshold, resulting in a staticky image. Although this method doesn't generate patterned artifacts, the noise tends to swamp the detail of the image. It is analogous to the practice of mezzotinting.
- *Patterning* dithers using a fixed pattern. For every pixel in the image the value of the pattern at the corresponding location is used as a threshold. Neighboring pixels do not affect each other, making this form of dithering suitable for use in animations. Different patterns can generate completely different dithering effects.
    - *Halftone dithering* looks similar to halftone screening in newspapers. This is a form of clustered dithering, in that dots tend to cluster together. This can help hide the adverse effects of blurry pixels found on some older output devices.

- *Ordered dithering* produces a cross-hatch pattern. This is a form of dispersed dithering. Because the dots don't cluster, the result looks much less grainy. Though simple to implement, this dithering algorithm is not easily changed to work with free-form, arbitrary palettes.

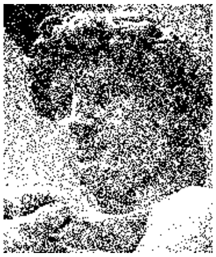
(Original)



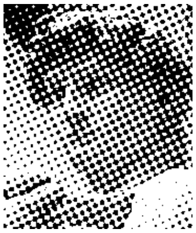
Threshold



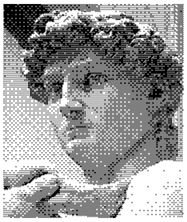
Random



Halftone

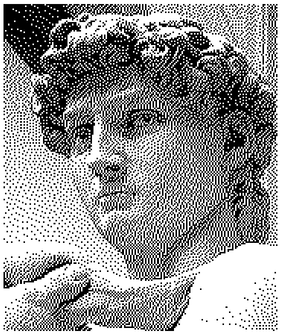


Bayer (ordered)

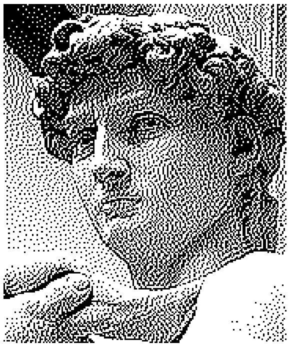


- *Error-diffusion dithering* is a feedback process that diffuses the quantization error to neighbouring pixels.
  - Floyd–Steinberg dithering only diffuses the error to neighbouring pixels. This results in very fine-grained dithering.
  - Jarvis, Judice, and Ninke dithering diffuses the error also to pixels one step further away. The dithering is coarser, but has fewer visual artifacts. It is slower than Floyd–Steinberg dithering because it distributes errors among 12 nearby pixels instead of 4 nearby pixels for Floyd–Steinberg.
  - Stucki dithering is based on the above, but is slightly faster. Its output tends to be clean and sharp.
  - Burkes dithering is a simplified form of Stucki dithering that is faster, but less clean than Stucki dithering.

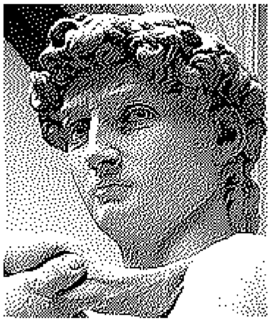
Floyd–Steinberg



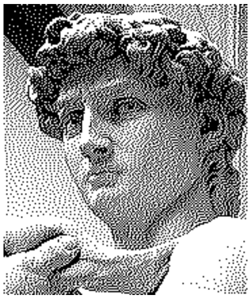
Jarvis, Judice & Ninke



Stucki



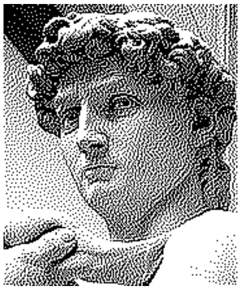
Burkes



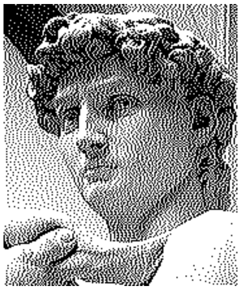
- Error-diffusion dithering (continued):
  - Sierra dithering is based on Jarvis dithering, but it's faster while giving similar results.
  - Two-row Sierra is the above method modified by Sierra to improve its speed.

- Filter Lite is an algorithm by Sierra that is much simpler and faster than Floyd–Steinberg, while still yielding similar (according to Sierra, better) results.
- Atkinson dithering resembles Jarvis dithering and Sierra dithering, but it's faster. Another difference is that it doesn't diffuse the entire quantization error, but only three quarters. It tends to preserve detail well, but very light and dark areas may appear blown out.
  - History
  - Algorithm
- Even toned screening is a patented modification of Floyd–Steinberg dithering intended to reduce visual artifacts, in particular to produce more even dot patterns in highlights and shadows.

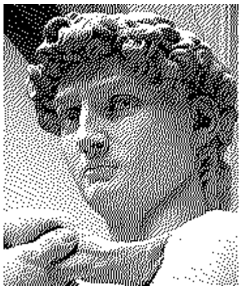
Sierra



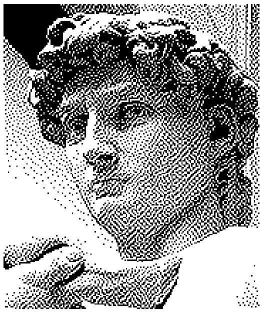
Two-row Sierra



Sierra Lite



Atkinson

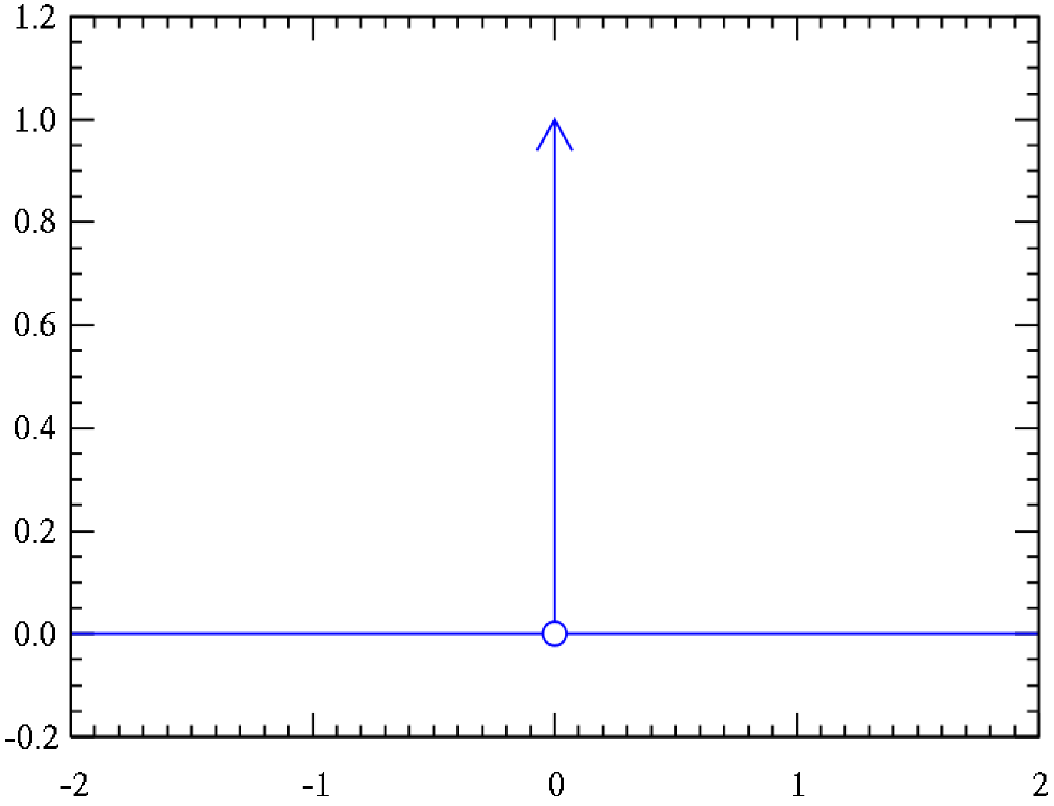


In optical fiber systems

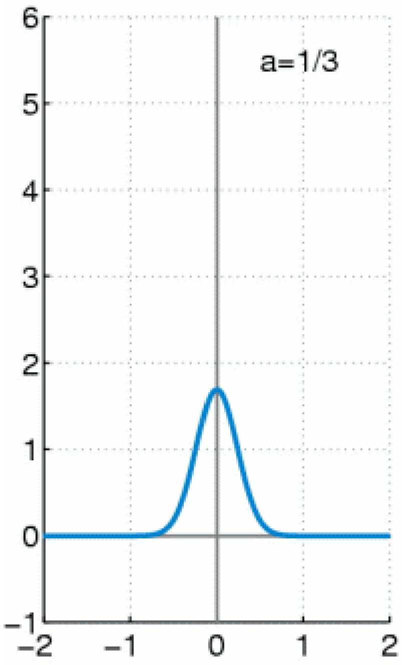
Stimulated Brillouin Scattering (SBS) is a nonlinear optical effect that limits the launched optical power in fiber optic systems. This power limit can be increased by dithering the transmit optical center frequency, typically implemented by modulating the laser's bias input.

Chapter 11

Dirac Delta Function



Schematic representation of the Dirac delta function by a line surmounted by an arrow. The height of the arrow is usually used to specify the value of any multiplicative constant, which will give the area under the function. The other convention is to write the area next to the arrowhead.



The Dirac delta function as the limit (in the sense of distributions) of the sequence of Gaussians $\delta_a(x) = \frac{1}{a\sqrt{\pi}}\mathrm{e}^{-x^2/a^2}$ as $a \to 0$.

The **Dirac delta function**, or **δ function**, is (informally) a generalized function depending on a real parameter such that it is zero for all values of the parameter except when the parameter is zero, and its integral over the parameter from −∞ to ∞ is equal to one. It was introduced by theoretical physicist Paul Dirac. In the context of signal processing it is often referred to as the **unit impulse function**. It is a continuous analog of the Kronecker delta function which is usually defined on a finite domain, and takes values 0 and 1.

From a purely mathematical viewpoint, the Dirac delta is not strictly a function, because any real function that is equal to zero everywhere but a single point must have total integral zero. While for many purposes the Dirac delta can be manipulated as a function, formally it can be defined as a distribution that is also a measure. In many applications, the Dirac delta is regarded as a kind of limit (a weak limit) of a sequence of functions having a tall spike at the origin. The approximating functions of the sequence are thus "approximate" or "nascent" delta functions.

Overview

The graph of the delta function is usually thought of as following the whole *x*-axis and the positive *y*-axis. (This informal picture can sometimes be misleading, for example in the limiting case of the sinc function.)

Despite its name, the delta function is not truly a function, at least not a usual one with domain in reals. For example, the objects *f*(*x*) = δ(*x*) and *g*(*x*) = 0 are equal everywhere except at *x* = 0 yet have integrals that are different. According to Lebesgue integration theory, if *f* and *g* are functions such that *f* = *g* almost everywhere, then *f* is integrable if and only if *g* is integrable and the integrals of *f* and *g* are identical. Rigorous treatment of the Dirac delta requires measure theory, the theory of distributions, or a hyperreal framework.

The Dirac delta is used to model a tall narrow spike function (an *impulse*), and other similar abstractions such as a point charge, point mass or electron point. For example, to calculate the dynamics of a baseball being hit by a bat, one can approximate the force of the bat hitting the baseball by a delta function. In doing so, one not only simplifies the equations, but one also is able to calculate the motion of the baseball by only considering the total impulse of the bat against the ball rather than requiring knowledge of the details of how the bat transferred energy to the ball.

In applied mathematics, the delta function is often manipulated as a kind of limit (a weak limit) of a sequence of functions, each member of which has a tall spike at the origin: for example, a sequence of Gaussian distributions centered at the origin with variance tending to zero.

An infinitesimal formula for an infinitely tall, unit impulse delta function (infinitesimal version of Cauchy distribution) explicitly appears in an 1827 text of Augustin Louis Cauchy. Siméon Denis Poisson considered the issue in connection with the study of wave propagation as did Gustav Kirchhoff somewhat later. Kirchhoff and Hermann von Helmholtz also introduced the unit impulse as a limit of Gaussians, which also corresponded to Lord Kelvin's notion of a point heat source. At the end of the 19th century, Oliver Heaviside used formal Fourier series to manipulate the unit impulse. The Dirac delta function as such was introduced as a "convenient notation" by Paul Dirac in his influential 1927 book *Principles of Quantum Mechanics*. He called it the "delta function" since he used it as a continuous analogue of the discrete Kronecker delta.

Definitions

The Dirac delta can be loosely thought of as a function on the real line which is zero everywhere except at the origin, where it is infinite,

$$\delta(x) = \begin{cases} +\infty, & x = 0 \\ 0, & x \neq 0 \end{cases}$$

and which is also constrained to satisfy the identity

$$\int_{-\infty}^{\infty} \delta(x)\, dx = 1.$$

This is merely a heuristic definition. The Dirac delta is not a true function, as no function has the above properties. Moreover there exist descriptions of the delta function which differ from the above conceptualization. For example, sinc(x/a)/a (where sinc is the sinc function) becomes the delta function in the limit as $a \to 0$, yet this function does not approach zero for values of x outside the origin, rather it oscillates between $1/x$ and $-1/x$ more and more rapidly as a approaches zero.

The Dirac delta function can be rigorously defined either as a distribution or as a measure.

As a measure

One way to rigorously define the delta function is as a measure, which accepts as an argument a subset A of the real line **R**, and returns $\delta(A) = 1$ if $0 \in A$, and $\delta(A) = 0$ otherwise. If the delta function is conceptualized as modeling an idealized point mass at 0, then $\delta(A)$ represents the mass contained in the set A. One may then define the integral against δ as the integral of a function against this mass distribution. Formally, the Lebesgue integral provides the necessary analytic device. The Lebesgue integral with respect to the measure δ satisfies

$$\int_{-\infty}^{\infty} f(x)\,\delta\{dx\} = f(0)$$

for all continuous compactly supported functions f. The measure δ is not absolutely continuous with respect to the Lebesgue measure — in fact, it is a singular measure. Consequently, the delta measure has no Radon–Nikodym derivative — no true function for which the property

$$\int_{-\infty}^{\infty} f(x)\delta(x)\,dx = f(0)$$

holds. As a result, the latter notation is a convenient abuse of notation, and not a standard (Riemann or Lebesgue) integral.

As a probability measure on **R**, the delta measure is characterized by its cumulative distribution function, which is the Heaviside function (or unit step function)

$$H(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0. \end{cases}$$

This means that $H(x)$ is the integral of the cumulative indicator function $\mathbf{1}_{(-\infty,\,x]}$ with respect to the measure δ; to wit,

$$H(x) = \int_{\mathbb{R}} \mathbf{1}_{(-\infty,x]}(t)\,\delta\{dt\} = \delta(-\infty,x].$$

Thus in particular the integral of the delta function against a continuous function can be properly understood as a Stieltjes integral:

$$\int_{-\infty}^{\infty} f(x)\delta\{dx\} = \int_{-\infty}^{\infty} f(x)\,dH(x).$$

All higher moments of δ are zero. In particular, characteristic function and moment generating function are both equal to one.

As a distribution

In the theory of distributions a generalized function is thought of not as a function itself, but only in relation to how it affects other functions when it is "integrated" against them. In keeping with this philosophy, to define the delta function properly, it is enough to say what the "integral" of the delta function against a sufficiently "good" test function is. If the delta function is already understood as a measure, then the Lebesgue integral of a test function against that measure supplies the necessary integral.

A typical space of test functions consists of all smooth functions on **R** with compact support. As a distribution, the Dirac delta is a linear functional on the space of test functions and is defined by

$$\delta[\varphi] = \varphi(0) \qquad \textbf{(\textit{1})}$$

for every test function φ.

For δ to be properly a distribution, it must be "continuous" in a suitable sense. In general, for a linear functional *S* on the space of test functions to define a distribution, it is necessary and sufficient that, for every positive integer *N* there is an integer M_N and a constant C_N such that for every test function φ, one has the inequality

$$|S[\phi]| \le C_N \sum_{k=0}^{M_N} \sup_{x\in[-N,N]} |\phi^{(k)}(x)|.$$

With the δ distribution, one has such an inequality (with $C_N = 1$) with $M_N = 0$ for all *N*. Thus δ is a distribution of order zero. It is, furthermore, a distribution with compact support (the support being {0}).

The delta distribution can also be defined in a number of equivalent ways. For instance, it is the distributional derivative of the Heaviside step function. This means that, for every test function φ, one has

$$\delta[\phi] = -\int_{-\infty}^{\infty} \phi'(x)H(x)\,dx.$$

Intuitively, if integration by parts were permitted, then the latter integral should simplify to

$$\int_{-\infty}^{\infty} \phi(x)H'(x)\,dx = \int_{-\infty}^{\infty} \phi(x)\delta(x)\,dx,$$

and indeed, a form of integration by parts is permitted for the Stieltjes integral, and in that case one does have

$$-\int_{-\infty}^{\infty} \phi'(x)H(x)\,dx = \int_{-\infty}^{\infty} \phi(x)\,dH(x).$$

In the context of measure theory, the Dirac measure gives rise to a distribution by integration. Conversely, equation (**1**) defines a Daniell integral on the space of all compactly supported continuous functions φ which, by the Riesz representation theorem, can be represented as the Lebesgue integral of φ with respect to some Radon measure.

Generalizations

The delta function can be defined in *n*-dimensional Euclidean space $\mathbf{R}^n$ as the measure such that

$$\int_{\mathbf{R}^n} f(\mathbf{x})\delta\{d\mathbf{x}\} = f(\mathbf{0})$$

for every compactly supported continuous function *f*. As a measure, the *n*-dimensional delta function is the product measure of the 1-dimensional delta functions in each variable separately. Thus, formally, with **x** = $(x_1, x_2, \ldots, x_n)$, one has

$$\delta(\mathbf{x}) = \delta(x_1)\delta(x_2)\ldots\delta(x_n). \qquad \textbf{(}\mathbf{\textit{2}}\textbf{)}$$

The delta function can also be defined in the sense of distributions exactly as above in the one-dimensional case. However, despite widespread use in engineering contexts, (**2**) should be manipulated with care, since the product of distributions can only be defined under quite narrow circumstances.

The notion of a **Dirac measure** makes sense on any set whatsoever. Thus if *X* is a set, $x_0 \in X$ is a marked point, and Σ is any sigma algebra of subsets of *X*, then the measure defined on sets $A \in \Sigma$ by

$$\delta_{x_0}(A) = \begin{cases} 1 & \text{if } \mathrm{x_0 \in A} \\ 0 & \text{if } \mathrm{x_0 \notin A} \end{cases}$$

is the delta measure or unit mass concentrated at x_0.

Another common generalization of the delta function is to a differentiable manifold where most of its properties as a distribution can also be exploited because of the differentiable structure. The delta function on a manifold M centered at the point $x_0 \in M$ is defined as the following distribution:

$$\delta_{x_0}[\phi] = \phi(x_0) \qquad \textbf{(\textit{3})}$$

for all compactly supported smooth real-valued functions φ on M. A common special case of this construction is when M is an open set in the Euclidean space $\mathbf{R}^n$.

On a locally compact Hausdorff space X, the Dirac delta measure concentrated at a point x is the Radon measure associated with the Daniell integral (**3**) on compactly supported continuous functions φ. At this level of generality, calculus as such is no longer possible, however a variety of techniques from abstract analysis are available. For instance, the mapping $x_0 \mapsto \delta_{x_0}$is a continuous embedding of X into the space of finite Radon measures on X, equipped with its vague topology. Moreover, the convex hull of the image of X under this embedding is dense in the space of probability measures on X.

Properties

Scaling and symmetry

The delta function satisfies the following scaling property for a non-zero scalar α:

$$\int_{-\infty}^{\infty} \delta(\alpha x)\,dx = \int_{-\infty}^{\infty} \delta(u)\,\frac{du}{|\alpha|} = \frac{1}{|\alpha|}$$

and so

$$\delta(\alpha x) = \frac{\delta(x)}{|\alpha|}. \qquad \textbf{(\textit{4})}$$

In particular, the delta function is an even distribution, in the sense that

$$\delta(- x) = \delta(x)$$

which is homogeneous of degree −1.

Algebraic properties

The distributional product of δ with *x* is equal to zero:

$$x\delta(x) = 0.$$

Conversely, if $xf(x) = xg(x)$, where *f* and *g* are distributions, then

$$f(x) = g(x) + c\delta(x)$$

for some constant *c*.

Translation

The integral of the time-delayed Dirac delta is given by**:**

$$\int\limits_{-\infty}^{\infty} f(t)\delta(t-T)\,dt = f(T)$$

This is sometimes referred to as the *sifting property* or the *sampling property*. The delta function is said to "sift out" the value at $t = T$.

It follows that the effect of convolving a function $f(t)$ with the time-delayed Dirac delta is to time-delay $f(t)$ by the same amount**:**

$$\begin{aligned}(f * \delta(t-T)) &\stackrel{\mathrm{def}}{=} \int\limits_{-\infty}^{\infty} f(\tau)\cdot\delta(t-T-\tau)\,d\tau \\ &= \int\limits_{-\infty}^{\infty} f(\tau)\cdot\delta(\tau-(t-T))\,d\tau\end{aligned}$$

(using (**4**): $\delta(-x) = \delta(x)$)

$$= f(t-T)$$

This holds under the precise condition that *f* be a tempered distribution. As a special case, for instance, we have the identity (understood in the distribution sense)

$$\int_{-\infty}^{\infty} \delta(\xi - x)\delta(x-\eta)\,dx = \delta(\xi-\eta).$$

Composition with a function

More generally, the delta distribution may be composed with a smooth function *g*(*x*) in such a way that the familiar change of variables formula holds, that

$$\int_{\mathbf{R}} \delta\big(g(x)\big) f\big(g(x)\big) |g'(x)| \, dx = \int_{g(\mathbf{R})} \delta(u) f(u) \, du$$

provided that *g* is a continuously differentiable function with *g*′ nowhere zero. That is, there is a unique way to assign meaning to the distribution $\delta \circ g$so that this identity holds for all compactly supported test functions *f*. This distribution satisfies δ(*g*(*x*)) = 0 if *g* is nowhere zero, and otherwise if *g* has a real root at x_0, then

$$\delta(g(x)) = \frac{\delta(x - x_0)}{|g'(x_0)|}.$$

It is natural therefore to *define* the composition δ(*g*(*x*)) for continuously differentiable functions *g* by

$$\delta(g(x)) = \sum_i \frac{\delta(x - x_i)}{|g'(x_i)|}$$

where the sum extends over all roots of *g*(*x*), which are assumed to be simple. Thus, for example

$$\delta(x^2 - \alpha^2) = \frac{1}{2|\alpha|} [\delta(x + \alpha) + \delta(x - \alpha)]$$

In the integral form the generalized scaling property may be written as

$$\int_{-\infty}^{\infty} f(x) \, \delta(g(x)) \, dx = \sum_i \frac{f(x_i)}{|g'(x_i)|}$$

Properties in *n* dimensions

The delta distribution in an *n*-dimensional space satisfies the following scaling property instead:

$$\delta(\alpha \mathbf{x}) = |\alpha|^{-n} \delta(\mathbf{x})$$

so that δ is a homogeneous distribution of degree −*n*. Under any reflection or rotation ρ, the delta function is invariant:

$$\delta(\rho \mathbf{x}) = \delta(\mathbf{x}).$$

As in the one-variable case, it is possible to define the composition of δ with a bi-Lipschitz function $g : \mathbf{R}^n \to \mathbf{R}^n$ uniquely so that the identity

$$\int_{\mathbf{R}^n} \delta(g(\mathbf{x}))\, f(g(\mathbf{x}))\,|\det g'(\mathbf{x})|\, d\mathbf{x} = \int_{g(\mathbf{R}^n)} \delta(u) f(u)\, du$$

for all compactly supported functions f.

Using the coarea formula from geometric measure theory, one can also define the composition of the delta function with a submersion from one Euclidean space to another one of different dimension; the result is a type of current. In the special case of a continuously differentiable function $g : \mathbf{R}^n \to \mathbf{R}$ such that the gradient of g is nowhere zero, the following identity holds

$$\int_{\mathbb{R}^n} f(\mathbf{x})\, \delta(g(\mathbf{x}))\, d\mathbf{x} = \int_{g^{-1}(0)} \frac{f(\mathbf{x})}{|\nabla g|}\, d\sigma(\mathbf{x})$$

where the integral on the right is over $g^{-1}(0)$, the $n-1$ dimensional surface defined by $g(\mathbf{x}) = 0$ with respect to the Minkowski content measure. This is known as a simple layer integral.

Fourier transform

The delta function is a tempered distribution, and therefore it has a well-defined Fourier transform. Formally, one finds

$$\hat{\delta}(\xi) = \int_{-\infty}^{\infty} e^{-2\pi i x \xi} \delta(x)\, dx = 1.$$

Properly speaking, the Fourier transform of a distribution is defined by imposing self-adjointness of the Fourier transform under the duality pairing $\langle \cdot, \cdot \rangle$of tempered distributions with Schwartz functions. Thus $\hat{\delta}$is defined as the unique tempered distribution satisfying

$$\langle \hat{\delta}, \phi \rangle = \langle \delta, \hat{\phi} \rangle$$

for all Schwartz functions φ. And indeed it follows from this that $\hat{\delta} = 1.$

As a result of this identity, the convolution of the delta function with any other tempered distribution S is simply S:

$$S * \delta = S.$$

That is to say that δ is an identity element for the convolution on tempered distributions, and in fact the space of compactly supported distributions under convolution is an associative algebra with identity the delta function. This property is fundamental in signal processing, as convolution with a tempered distribution is a linear time-invariant system, and applying the linear time-invariant system measures its impulse response. The impulse response can be computed to any desired degree of accuracy by choosing a suitable approximation for δ, and once it is known, it characterizes the system completely.

The inverse Fourier transform of the tempered distribution $f(\xi) = 1$ is the delta function. Formally, this is expressed

$$\int_{-\infty}^{\infty} 1 \cdot e^{2\pi i x \xi}\, d\xi = \delta(x)$$

and more rigorously, it follows since

$$\langle 1, f^{\vee} \rangle = f(0) = \langle \delta, f \rangle$$

for all Schwartz functions f.

In these terms, the delta function provides a suggestive statement of the orthogonality property of the Fourier kernel on **R**. Formally, one has

$$\int_{-\infty}^{\infty} e^{i2\pi\xi_1 t} \left[e^{i2\pi\xi_2 t}\right]^* dt = \int_{-\infty}^{\infty} e^{-i2\pi(\xi_2 - \xi_1)t}\, dt = \delta(\xi_1 - \xi_2).$$

This is, of course, shorthand for the assertion that the Fourier transform of the tempered distribution

$$f(t) = e^{i2\pi\xi_1 t}$$

is

$$\hat{f}(\xi_2) = \delta(\xi_1 - \xi_2)$$

which again follows by imposing self-adjointness of the Fourier transform.

By analytic continuation of the Fourier transform, the Laplace transform of the delta function is found to be

$$\int_0^{\infty} \delta(t - a) e^{-st}\, dt = e^{-sa}.$$

Distributional derivatives

The distributional derivative of the Dirac delta distribution is the distribution δ′ defined on compactly supported smooth test functions φ by

$$\delta'[\varphi] = -\delta[\varphi'] = -\varphi'(0).$$

The first equality here is a kind of integration by parts, for if δ were a true function then

$$\int_{-\infty}^{\infty} \delta'(x)\varphi(x)\,dx = -\int_{-\infty}^{\infty} \delta(x)\varphi'(x)\,dx.$$

The k^{th} derivative of δ is defined similarly as the distribution given on test functions by

$$\delta^{(k)}[\varphi] = (-1)^k \varphi^{(k)}(0).$$

In particular δ is an infinitely differentiable distribution.

The first derivative of the delta function is the distributional limit of the difference quotients:

$$\delta'(x) = \lim_{h\to 0} \frac{\delta(x+h)-\delta(x)}{h}$$

More properly, one has

$$\delta' = \lim_{h\to 0} \frac{1}{h}(\tau_h\delta - \delta)$$

where τ_h is the translation operator, defined on functions by $\tau_h\varphi(x) = \varphi(x+h)$, and on a distribution S by

$$(\tau_h S)[\varphi] = S[\tau_{-h}\varphi].$$

In the theory of electromagnetism, the first derivative of the delta function represents a point magnetic dipole situated at the origin. Accordingly, it is referred to as a dipole or the doublet function.

The derivative of the delta function satisfies a number of basic properties, including:

$$\frac{d}{dx}\delta(-x) = -\frac{d}{dx}\delta(x)$$

$$x\delta'(x) = -\,\delta(x).$$

Furthermore, the convolution of δ' with a compactly supported smooth function f is

$$\delta' * f = \delta * f' = f',$$

which follows from the properties of the distributional derivative of a convolution.

Higher dimensions

More generally, on an open set U in the n-dimensional Euclidean space $\mathbf{R}^n$, the Dirac delta distribution centered at a point $a \in U$ is defined by

$$\delta_a[\varphi] = \varphi(a)$$

for all $\varphi \in S(U)$, the space of all smooth compactly supported functions on U. If $\alpha = (\alpha_1, ..., \alpha_n)$ is any multi-index and ∂^α denotes the associated mixed partial derivative operator, then the α^{th} derivative $\partial^\alpha \delta_a$ of δ_a is given by

$$\langle \partial^\alpha \delta_a, \varphi \rangle = (-1)^{|\alpha|} \langle \delta_a, \partial^\alpha \varphi \rangle = (-1)^{|\alpha|} \partial^\alpha \varphi(x) \Big|_{x=a} \text{ for all } \varphi \in S(U).$$

That is, the α^{th} derivative of δ_a is the distribution whose value on any test function φ is the α^{th} derivative of φ at a (with the appropriate positive or negative sign).

The first partial derivatives of the delta function are thought of as double layers along the coordinate planes. More generally, the normal derivative of a simple layer supported on a surface is a double layer supported on that surface, and represents a laminar magnetic monopole. Higher derivatives of the delta function are known in physics as multipoles.

Higher derivatives enter into mathematics naturally as the building blocks for the complete structure of distributions with point support. If S is any distribution on U supported on the set $\{a\}$ consisting of a single point, then there is an integer m and coefficients c_α such that

$$S = \sum_{|\alpha| \le m} c_\alpha \partial^\alpha \delta_a.$$

Representations of the delta function

The delta function can be viewed as the limit of a sequence of functions

$$\delta(x) = \lim_{\varepsilon \to 0+} \eta_\varepsilon(x),$$

where $\eta_\varepsilon(x)$ is sometimes called a **nascent delta function**. This limit is meant in a weak sense: either that

$$\lim_{\varepsilon \to 0+} \int_{-\infty}^{\infty} \eta_\varepsilon(x) f(x)\, dx = f(0) \qquad \textbf{(5)}$$

for all continuous functions f having compact support, or that this limit holds for all smooth functions f with compact support. The difference between these two slightly different modes of weak convergence is often subtle: the former is convergence in the vague topology of measures, and the latter is convergence in the sense of distributions.

Approximations to the identity

Typically a nascent delta function η_ε can be constructed in the following manner. Let η be an absolutely integrable function on **R** of total integral 1, and define

$$\eta_\varepsilon(x) = \eta(x/\varepsilon)/\varepsilon.$$

In n dimensions, one uses instead the scaling

$$\eta_\varepsilon(x) = \eta(x/\varepsilon)/\varepsilon^n.$$

Then a simple change of variables shows that η_ε also has integral 1. One shows easily that (**5**) holds for all continuous compactly supported functions f, and so η_ε converges weakly to δ in the sense of measures. If the initial $\eta = \eta_1$ is itself smooth and compactly supported then the sequence is called a mollifier.

The η_ε constructed in this way are known as an **approximation to the identity**. This terminology is because the space $L^1(\mathbf{R})$ of absolutely integrable functions is closed under the operation of convolution of functions: $f*g \in L^1(\mathbf{R})$ whenever f and g are in $L^1(\mathbf{R})$. However, there is no identity in $L^1(\mathbf{R})$ for the convolution product: no element h such that $f*h = f$ for all f. Nevertheless, the sequence η_ε does approximate such an identity in the sense that

$$f * \eta_\varepsilon \to f \quad \text{as } \varepsilon \to 0.$$

This limit holds in the sense of mean convergence (convergence in L^1). Further conditions on the η_ε, for instance that it be a mollifier associated to a compactly supported function, are needed to ensure pointwise convergence almost everywhere.

The standard mollifier is given by $\Psi(x/\varepsilon)/\varepsilon$ where Ψ is a suitably normalized bump function. For instance,

$$\eta_\varepsilon(x) = \frac{\Psi(x/\varepsilon)}{\int_{-\infty}^{\infty} \Psi(x/\varepsilon)\, dx}$$

where

$$\Psi(x) = \begin{cases} e^{-1/(1-|x|^2)} & \text{if } |x| < 1 \\ 0 & \text{if } |x| \geq 1. \end{cases}$$

In some situations such as numerical analysis, a piecewise linear approximation to the identity is desirable. This can be obtained by taking η_1 to be a hat function. With this choice of η_1, one has

$$\eta_\varepsilon(x) = \varepsilon^{-1}\max(1 - |x/\varepsilon|, 0)$$

which are all continuous and compactly supported, although not smooth and so not a mollifier.

Probabilistic considerations

In the context of probability theory, it is natural to impose the additional condition that the initial η_1 in an approximation to the identity should be positive, as such a function then represents a probability distribution. Convolution with a probability distribution is sometimes favorable because it does not result in overshoot or undershoot, as the output is a convex combination of the input values, and thus falls between the maximum and minimum of the input function. Taking η_1 to be any probability distribution at all, and letting $\eta_\varepsilon(x) = \eta_1(x/\varepsilon)/\varepsilon$ as above will give rise to an approximation to the identity. In general this converges more rapidly to a delta function if, in addition, η has mean 0 and has small higher moments. For instance, if η_1 is the uniform distribution on [−1/2, 1/2], also known as the rectangular function, then setting $\varepsilon = 1/(2n)$, with n a new parameter:

$$\eta_\varepsilon(x) = \delta_n(x) = 2n\ \mathrm{rect}(nx/2) = \begin{cases} 2n, & -\frac{1}{n} < x < \frac{1}{n} \\ 0, & \text{otherwise.} \end{cases}$$

Another example is with the Wigner semicircle distribution

$$\eta_\varepsilon(x) = \begin{cases} \frac{2}{\pi\varepsilon^2}\sqrt{\varepsilon^2 - x^2}, & -\varepsilon < x < \varepsilon \\ 0, & \text{otherwise} \end{cases}$$

This is continuous and compactly supported, but not a mollifier because it is not smooth.

Semigroups

Nascent delta functions often arise as convolution semigroups. This amounts to the further constraint that the convolution of η_ε with η_δ must satisfy

$$\eta_\varepsilon * \eta_\delta = \eta_{\varepsilon+\delta}$$

for all ε,δ > 0. Convolution semigroups in L^1 that form a nascent delta function are always an approximation to the identity in the above sense, however the semigroup condition is quite a strong restriction.

In practice, semigroups approximating the delta function arise as fundamental solutions or Green's functions to physically motivated elliptic or parabolic partial differential equations. In the context of applied mathematics, semigroups arise as the output of a linear time-invariant system. Abstractly, if *A* is a linear operator acting on functions of *x*, then a convolution semigroup arises by solving the initial value problem

$$\begin{cases} \frac{\partial}{\partial t}\eta(t,x) = A\eta(t,x), & t > 0 \\ \lim\limits_{t\to 0^+} \eta(t,x) = \delta(x) \end{cases}$$

in which the limit is as usual understood in the weak sense. Setting $\eta_\varepsilon(x) = \eta(\varepsilon, x)$ gives the associated nascent delta function.

Some examples of physically important convolution semigroups arising from such a fundamental solution include the following.

The heat kernel

The heat kernel, defined by

$$\eta_\varepsilon(x) = \frac{1}{\sqrt{2\pi\varepsilon}}\mathrm{e}^{-x^2/2\varepsilon}$$

represents the temperature in an infinite wire at time $t > 0$, if a unit of heat energy is stored at the origin of the wire at time $t = 0$. This semigroup evolves according to the one-dimensional heat equation:

$$\frac{\partial u}{\partial t} = \frac{1}{2}\frac{\partial^2 u}{\partial x^2}.$$

In probability theory, $\eta_\varepsilon(x)$ is a normal distribution of variance ε and mean 0. It represents the probability density at time $t = \varepsilon$ of the position of a particle starting at the origin following a standard Brownian motion. In this context, the semigroup condition is then an expression of the Markov property of Brownian motion.

In higher dimensional Euclidean space $\mathbf{R}^n$, the heat kernel is

$$\eta_\varepsilon = \frac{1}{(2\pi\varepsilon)^{n/2}}\mathrm{e}^{-x\cdot x/2\varepsilon},$$

and has the same physical interpretation, *mutatis mutandis*. It also represents a nascent delta function in the sense that $\eta_\varepsilon \to \delta$ in the distribution sense as $\varepsilon \to 0$.

The Poisson kernel

$$\eta_\varepsilon(x) = \frac{1}{\pi}\frac{\varepsilon}{\varepsilon^2 + x^2} = \int_{-\infty}^{\infty} \mathrm{e}^{2\pi\mathrm{i}\xi x - |\varepsilon\xi|}\, d\xi$$

is the fundamental solution of the Laplace equation in the upper half-plane. It represents the electrostatic potential in a semi-infinite plate whose potential along the edge is held at fixed at the delta function. The Poisson kernel is also closely related to the Cauchy distribution. This semigroup evolves according to the equation

$$\frac{\partial u}{\partial t} = -(-\partial^2/\partial x^2)^{1/2} u(t,x)$$

where the operator is rigorously defined as the Fourier multiplier

$$\mathcal{F}\left[(-\partial^2/\partial x^2)^{1/2} f\right](\xi) = |2\pi\xi|\mathcal{F}f(\xi).$$

Oscillatory integrals

In areas of physics such as wave propagation and wave mechanics, the equations involved are hyperbolic and so may have more singular solutions. As a result, the nascent delta functions that arise as fundamental solutions of the associated Cauchy problems are generally oscillatory integrals. An example, which comes from a solution of the Euler–Tricomi equation of transonic gas dynamics, is the rescaled Airy function

$$\mathrm{Ai}(x/\varepsilon^{1/3})/\varepsilon^{1/3}.$$

Although using the Fourier transform, it is easy to see that this generates a semigroup in some sense, it is not absolutely integrable and so cannot define a semigroup in the above strong sense. Many nascent delta functions constructed as oscillatory integrals only converge in the sense of distributions (an example is the Dirichlet kernel below), rather than in the sense of measures.

Another example is the Cauchy problem for the wave equation in $\mathbf{R}^{1+1}$:

$$c^{-2}\frac{\partial^2 u}{\partial t^2} - \Delta u = 0$$
$$u = 0, \quad \frac{\partial u}{\partial t} = \delta \quad \text{when t} = 0.$$

The solution *u* represents the displacement from equilibrium of an infinite elastic string, with an initial disturbance at the origin.

Other approximations to the identity of this kind include the sinc function

$$\eta_\varepsilon(x) = \frac{1}{\pi x}\sin\left(\frac{x}{\varepsilon}\right) = \frac{1}{2\pi}\int_{-1/\varepsilon}^{1/\varepsilon}\cos(kx)\ dk$$

and the Bessel function

$$\eta_\varepsilon(x) = \frac{1}{\varepsilon}J_{1/\varepsilon}\left(\frac{x+1}{\varepsilon}\right).$$

Plane wave decomposition

One approach to the study of a linear partial differential equation

$$L[u] = f,$$

where *L* is a differential operator on $\mathbf{R}^n$, is to seek first a fundamental solution, which is a solution of the equation

$$L[u] = \delta.$$

When *L* is particularly simple, this problem can often be resolved using the Fourier transform directly (as in the case of the Poisson kernel and heat kernel already mentioned). For more complicated operators, it is sometimes easier first to consider an equation of the form

$$L[u] = h$$

where *h* is a plane wave function, meaning that it has the form

$$h = h(x \cdot \xi)$$

for some vector ξ. Such an equation can be resolved (if the coefficients of *L* are analytic functions) by the Cauchy–Kovalevskaya theorem or (if the coefficients of *L* are constant) by quadrature. So, if the delta function can be decomposed into plane waves, then one can in principle solve linear partial differential equations.

Such a decomposition of the delta function into plane waves was part of a general technique first introduced essentially by Johann Radon, and then developed in this form by Fritz John (1955). Choose *k* so that *n* + *k* is an even integer, and for a real number *s*, put

$$g(s) = \operatorname{Re}\left[\frac{-s^k\log(-is)}{k!(2\pi i)^n}\right] = \begin{cases} \frac{|s|^k}{4k!(2\pi i)^{n-1}} & n \text{ odd} \\ -\frac{|s|^k\log|s|}{k!(2\pi i)^n} & n \text{ even.}\end{cases}$$

Then δ is obtained by applying a power of the Laplacian to the integral with respect to the unit sphere measure dω of $g(x\cdot\xi)$ for ξ in the unit sphere S^{n-1}:

$$\delta(x) = \Delta_x^{(n+k)/2} \int_{S^{n-1}} g(x\cdot\xi)\,d\omega_\xi.$$

The Laplacian here is interpreted as a weak derivative, so that this equation is taken to mean that, for any test function φ,

$$\varphi(x) = \int_{\mathbf{R}^n} \varphi(y)\,dy\, \Delta_x^{(n+k)/2} \int_{S^{n-1}} g((x-y)\cdot\xi)\,d\omega_\xi.$$

The result follows from the formula for the Newtonian potential (the fundamental solution of Poisson's equation). This is essentially a form of the inversion formula for the Radon transform, because it recovers the value of $\varphi(x)$ from its integrals over hyperplanes. For instance, if n is odd and $k = 1$, then the integral on the right hand side is

$$\begin{aligned} & c_n \Delta_x^{(n+1)/2} \iint_{S^{n-1}} \varphi(y)|(y-x)\cdot\xi|\,d\omega_\xi\,dy \\ &= c_n \Delta_x^{(n+1)/2} \int_{S^{n-1}} d\omega_\xi \int_{-\infty}^{\infty} |p| R\varphi(\xi, p + x\cdot\xi)\,dp \end{aligned}$$

where $R\varphi(\xi,p)$ is the Radon transform of φ:

$$R\varphi(\xi,p) = \int_{x\cdot\xi=p} f(x)\,d^{n-1}x.$$

An alternative equivalent expression of the plane wave decomposition, from Gel'fand & Shilov (1966–1968, I, §3.10), is

$$\delta(x) = \frac{(n-1)!}{(2\pi i)^n} \int_{S^{n-1}} (x\cdot\xi)^{-n}\,d\omega_\xi$$

for n even, and

$$\delta(x) = \frac{1}{2(2\pi i)^{n-1}} \int_{S^{n-1}} \delta^{(n-1)}(x\cdot\xi)\,d\omega_\xi$$

for n odd.

Fourier kernels

In the study of Fourier series, a major question consists of determining whether and in what sense the Fourier series associated with a periodic function converges to the function. The n^{th} partial sum of the Fourier series of a function f of period 2π is defined by convolution (on the interval [−π,π]) with the Dirichlet kernel:

$$D_N(x) = \sum_{n=-N}^{N} e^{inx} = \frac{\sin\left((N+\frac{1}{2})x\right)}{\sin(x/2)}.$$

Thus,

$$s_N(f)(x) = D_N * f(x) = \sum_{n=-N}^{N} a_n e^{inx}$$

where

$$a_n = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(y)e^{-iny}\,dy.$$

A fundamental result of elementary Fourier series states that the Dirichlet kernel tends to the a multiple of the delta function as $N \to \infty$. This is interpreted in the distribution sense, that

$$s_N(f)(0) = \int_{\mathbf{R}} D_N(x)f(x)\,dx \to 2\pi f(0)$$

for every compactly supported *smooth* function f. Thus, formally one has

$$\delta(x) = \frac{1}{2\pi}\sum_{n=-\infty}^{\infty} e^{inx}$$

on the interval [−π,π].

In spite of this, the result does not hold for all compactly supported *continuous* functions: that is D_N does not converge weakly in the sense of measures. The lack of convergence of the Fourier series has led to the introduction of a variety of summability methods in order to produce convergence. The method of Cesàro summation leads to the Fejér kernel

$$F_N(x) = \sum_{n=0}^{N} D_n(x) = \frac{1}{N}\left(\frac{\sin\frac{Nx}{2}}{\sin\frac{x}{2}}\right)^2.$$

The Fejér kernels tend to the delta function in a stronger sense that

$$\int_{\mathbf{R}} F_N(x) f(x)\, dx \to 2\pi f(0)$$

for every compactly supported continuous function f. The implication is that the Fourier series of any continuous function is Cesàro summable to the value of the function at every point.

Hilbert space theory

The Dirac delta distribution is a densely defined unbounded linear functional on the Hilbert space L^2 of square integrable functions. Indeed, smooth compactly support functions are dense in L^2, and the action of the delta distribution on such functions is well-defined. In many applications, it is possible to identify subspaces of L^2 and to give a stronger topology on which the delta function defines a bounded linear functional.

Sobolev spaces

The Sobolev embedding theorem for Sobolev spaces on the real line **R** implies that any square-integrable function f such that

$$\|f\|_{H^1}^2 = \int_{-\infty}^{\infty} |\hat{f}(\xi)|^2 (1 + |\xi|^2)\, d\xi < \infty$$

is automatically continuous, and satisfies in particular

$$\delta[f] = |f(0)| < C\|f\|_{H^1}.$$

Thus δ is a bounded linear functional on the Sobolev space H^1. Equivalently δ is an element of the continuous dual space H^{-1} of H^1. More generally, in n dimensions, one has $\delta \in H^{-s}(\mathbf{R}^n)$ provided $s > n / 2$.

Spaces of holomorphic functions

In complex analysis, the delta function enters via Cauchy's integral formula which asserts that if D is a domain in the complex plane with smooth boundary, then

$$f(z) = \frac{1}{2\pi i} \oint_{\partial D} \frac{f(\zeta)\, d\zeta}{\zeta - z}, \quad z \in D$$

for all holomorphic functions f in D that are continuous on the closure of D. As a result, the delta function δ_z is represented on this class of holomorphic functions by the Cauchy integral:

$$\delta_z[f] = f(z) = \frac{1}{2\pi i} \oint_{\partial D} \frac{f(\zeta)\, d\zeta}{\zeta - z}.$$

More generally, let $H^2(\partial D)$ be the Hardy space consisting of the closure in $L^2(\partial D)$ of all holomorphic functions in D continuous up to the boundary of D. Then functions in $H^2(\partial D)$ uniquely extend to holomorphic functions in D, and the Cauchy integral formula continues to hold. In particular for $z \in D$, the delta function δ_z is a continuous linear functional on $H^2(\partial D)$. This is a special case of the situation in several complex variables in which, for smooth domains D, the Szegő kernel plays the role of the Cauchy integral.

Resolutions of the identity

Given a complete orthonormal basis set of functions $\{\varphi_n\}$ in a separable Hilbert space, for example, the normalized eigenvectors of a compact self-adjoint operator, any vector f can be expressed as:

$$f = \sum_{n=1}^{\infty} \alpha_n \varphi_n \ .$$

The coefficients $\{\alpha_n\}$ are found as:

$$\alpha_n = \langle \varphi_n ,\ f \rangle \ ,$$

which may be represented by the notation:

$$\alpha_n = \varphi_n^\dagger \ f \ ,$$

a form of the bra-ket notation of Dirac. Adopting this notation, the expansion of f takes the dyadic form:

$$f = \sum_{n=1}^{\infty} \ \varphi_n \left(\varphi_n^\dagger \ f \right) .$$

Letting I denote the identity operator on the Hilbert space, the expression

$$I = \sum_{n=1}^{\infty} \ \varphi_n \varphi_n^\dagger ,$$

is called a resolution of the identity. When the Hilbert space is the space $L^2(D)$ of square-integrable functions on a domain D, the quantity:

$$\varphi_n \varphi_n^\dagger ,$$

is an integral operator, and the expression for *f* can be rewritten as:

$$f(x) = \sum_{n=1}^{\infty} \int_D \left(\varphi_n(x)\varphi_n^*(\xi)\right) f(\xi)\, d\xi.$$

The right-hand side converges to *f* in the L^2 sense. It need not hold in a pointwise sense, even when *f* is a continuous function. Nevertheless, it is common to abuse notation and write

$$f(x) = \int \delta(x-\xi) f(\xi)\, d\xi,$$

resulting in the representation of the delta function:

$$\delta(x-\xi) = \sum_{n=1}^{\infty} \varphi_n(x)\varphi_n^*(\xi).$$

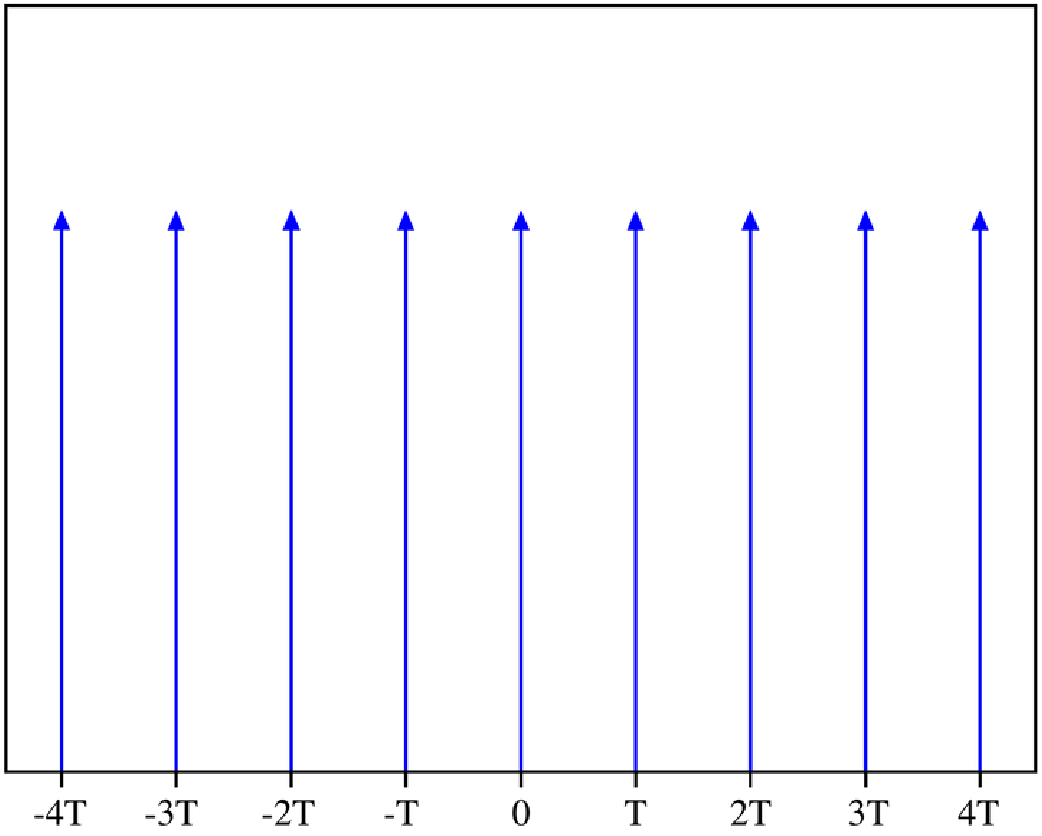
With a suitable rigged Hilbert space $(\Phi, L^2(D), \Phi^*)$ where $\Phi \subset L^2(D)$ contains all compactly supported smooth functions, this summation may converge in Φ^*, depending on the properties of the basis φ_n. In most cases of practical interest, the orthonormal basis comes from an integral or differential operator, in which case the series converges in the distribution sense.

Infinitesimal delta functions

Cauchy used an infinitesimal α to write down a unit impulse, infinitely tall and narrow Dirac-type delta function δ_α satisfying $\int F(x)\delta_\alpha(x) = F(0)$ in a number of articles in 1827. Cauchy defined an infinitesimal in Cours d'Analyse (1827) in terms of a sequence tending to zero. Namely, such a null sequence becomes an infinitesimal in Cauchy's and Lazare Carnot's terminology.

Modern set-theoretic approaches allow one to define infinitesimals via the ultrapower construction, where a null sequence becomes an infinitesimal in the sense of an equivalence class modulo a relation defined in terms of a suitable ultrafilter. The article by Yamashita (2007) contains a bibliography on modern Dirac delta functions in the context of an infinitesimal-enriched continuum provided by the hyperreals.

Dirac comb



A Dirac comb is an infinite series of Dirac delta functions spaced at intervals of *T*

A so-called uniform "pulse train" of Dirac delta measures, which is known as a Dirac comb, or as the Shah distribution, creates a sampling function, often used in digital signal processing (DSP) and discrete time signal analysis. The Dirac comb is given as the infinite sum, whose limit is understood in the distribution sense,

$$\Delta(x) = \sum_{n=-\infty}^{\infty} \delta(x-n),$$

which is a sequence of point masses at each of the integers.

Up to an overall normalizing constant, the Dirac comb is equal to its own Fourier transform. This is significant because if *f* is any Schwartz function, then the periodization of *f* is given by the convolution

$$(f * \Delta)(x) = \sum_{n=-\infty}^{\infty} f(x-n).$$

In particular,

$$(f * \Delta)^\wedge = \hat{f}\widehat{\Delta} = \hat{f}\Delta$$

is precisely the Poisson summation formula.

Sokhatsky–Weierstrass theorem

The Sokhatsky–Weierstrass theorem, important in quantum mechanics, relates the delta function to the distribution p.v.1/x, the Cauchy principal value of the function 1/x, defined by

$$\left\langle \mathrm{p.\,v.}\frac{1}{x}, \phi \right\rangle = \lim_{\epsilon\to 0^+} \int_{|x|>\epsilon} \frac{\phi(x)}{x}\,dx.$$

Sokhostsky's formula states that

$$\lim_{\epsilon\to 0^+} \frac{1}{x \pm i\epsilon} = \mathrm{p.\,v.}\frac{1}{x} \mp i\pi\delta(x),$$

Here the limit is understood in the distribution sense, that for all compactly supported smooth functions f,

$$\lim_{\epsilon\to 0^+} \int_{-\infty}^{\infty} \frac{f(x)}{x \pm i\epsilon}\,dx = \mp i\pi f(0) + \lim_{\epsilon\to 0^+} \int_{|x|>\epsilon} \frac{f(x)}{x}\,dx.$$

Relationship to the Kronecker delta

The Kronecker delta δ_{ij} is the quantity defined by

$$\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \ne j \end{cases}$$

for all integers i, j. This function then satisfies the following analog of the sifting property: if $(a_i)_{i\in\mathbb{Z}}$be any doubly infinite sequence, then

$$\sum_{i=-\infty}^{\infty} a_i \delta_{ik} = a_k.$$

Similarly, for any real or complex valued continuous function f on $\mathbb{R}$, the Dirac delta satisfies the sifting property

$$\int_{-\infty}^{\infty} f(x)\delta(x-x_0)\,dx = f(x_0).$$

This exhibits the Kronecker delta function as a discrete analog of the Dirac delta function.

Applications to probability theory

In probability theory and statistics, the Dirac delta function is often used to represent a discrete distribution, or a partially discrete, partially continuous distribution, using a probability density function (which is normally used to represent fully continuous distributions). For example, the probability density function $f(x)$of a discrete distribution consisting of points $\mathbf{x} = \{x_1, \dots, x_n\}$, with corresponding probabilities $p_1, \dots, p_n$, can be written as

$$f(x) = \sum_{i=1}^{n} p_i \delta(x - x_i)$$

As another example, consider a distribution which 6/10 of the time returns a standard normal distribution, and 4/10 of the time returns exactly the value 3.5 (i.e. a partly continuous, partly discrete mixture distribution). The density function of this distribution can be written as

$$f(x) = 0.6\,\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} + 0.4\,\delta(x-3.5).$$

The delta function is also used in a completely different way to represent the local time of a diffusion process (like Brownian motion). The local time of a stochastic process *B*(*t*) is given by

$$\ell(x,t) = \int_0^t \delta(x - B(s))\,ds$$

and represents the amount of time that the process spends at the point *x* in the range of the process. More precisely, in one dimension this integral can be written

$$\ell(x,t) = \lim_{\epsilon\to 0^+} \frac{1}{2\epsilon} \int_0^t \mathbf{1}_{[x-\epsilon,x+\epsilon]}(B(s))\,ds$$

where $\mathbf{1}_{[x-\epsilon,x+\epsilon]}$is the indicator function of the interval [*x*−*ε*,*x*+*ε*].

Application to quantum mechanics

We give an example of how the delta function is expedient in quantum mechanics. The wave function of a particle gives the probability amplitude of finding a particle within a given region of space. Wave functions are assumed to be elements of the Hilbert space L^2 of square-integrable functions, and the total probability of finding a particle within a given interval is the integral of the magnitude of the wave function squared over the interval. A set $\{\varphi_n\}$ of wave functions is orthonormal if they are normalized by

$$\langle\phi_n|\phi_m\rangle = \delta_{nm}$$

where δ here refers to the Kronecker delta. A set of orthonormal wave functions is complete in the space of square-integrable functions if any wave function ψ can be expressed as a combination of the φ_n:

$$\psi = \sum c_n\phi_n,$$

with $c_n = \langle\phi_n|\psi\rangle$. Complete orthonormal systems of wave functions appear naturally as the eigenfunctions of the Hamiltonian (of a bound system) in quantum mechanics that measures the energy levels, which are called the eigenvalues. The set of eigenvalues, in this case, is known as the spectrum of the Hamiltonian. In bra-ket notation, as above, this equality implies the resolution of the identity:

$$I = \sum |\phi_n\rangle\langle\phi_n|.$$

Here the eigenvalues are assumed to be discrete, but the set of eigenvalues of an observable may be continuous rather than discrete. An example is the position observable, $Q\psi(x) = x\psi(x)$. The spectrum of the position (in one dimension) is the entire real line, and is called a continuous spectrum. However, unlike the Hamiltonian, the position operator lacks proper eigenfunctions. The conventional way to overcome this shortcoming is to widen the class of available functions by allowing distributions as well: that is, to replace the Hilbert space of quantum mechanics by an appropriate rigged Hilbert space. In this context, the position operator has a complete set of eigen-distributions, labeled by the points y of the real line, given by

$$\varphi_y(x) = \delta(x - y).$$

The eigenfunctions of position are denoted by $\phi_y = |y\rangle$in Dirac notation, and are known as position eigenstates.

Similar considerations apply to the eigenstates of the momentum operator, or indeed any other self-adjoint unbounded operator P on the Hilbert space, provided the spectrum of P is continuous and there are no degenerate eigenvalues. In that case, there is a set Ω of real numbers (the spectrum), and a collection φ_y of distributions indexed by the elements of Ω, such that

$P\varphi_y = y\varphi_y.$

That is, φ_y are the eigenvectors of P. If the eigenvectors are normalized so that

$$\langle \phi_y, \phi_{y'} \rangle = \delta(y - y')$$

in the distribution sense, then for any test function ψ,

$$\psi(x) = \int_\Omega c(y)\phi_y(x)\,dy$$

where

$$c(y) = \langle \psi, \phi_y \rangle.$$

That is, as in the discrete case, there is a resolution of the identity

$$I = \int_\Omega |\phi_y\rangle\,\langle\phi_y|\,dy$$

where the operator-valued integral is again understood in the weak sense. If the spectrum of P has both continuous and discrete parts, then the resolution of the identity involves a summation over the discrete spectrum *and* an integral over the continuous spectrum.

The delta function also has many more specialized applications in quantum mechanics, such as the delta potential models for a single and double potential well.

Application to structural mechanics

The delta function can be used in structural mechanics to describe transient loads or point loads acting on structures. The governing equation of a simple mass-spring system excited by a sudden force impulse I at time $t = 0$ can be written

$$m\frac{\mathrm{d}^2\xi}{\mathrm{d}t^2} + k\xi = I\delta(t),$$

where m is the mass, ξ the deflection and k the spring constant.

As another example, the equation governing the static deflection of a slender beam is, according to Euler-Bernoulli theory,

$$EI\frac{\mathrm{d}^4w}{\mathrm{d}x^4} = q(x),$$

where *EI* is the bending stiffness of the beam, *w* the deflection, *x* the spatial coordinate and *q*(*x*) the load distribution. If a beam is loaded by a point force *F* at $x = x_0$, the load distribution is written

$$q(x) = F\delta(x - x_0).$$

As integration of the delta function results in the Heaviside step function, it follows that the static deflection of a slender beam subject to multiple point loads is described by a set of piecewise polynomials.

Also a point moment acting on a beam can be described by delta functions. Consider two opposing point forces *F* at a distance *d* apart. They then produce a moment *M* = *Fd* acting on the beam. Now, let the distance *d* approach the limit zero, while *M* is kept constant. The load distribution, assuming a clockwise moment acting at *x* = 0, is written

$$\begin{aligned} q(x) &= \lim_{d\to 0} \Big(F\delta(x) - F\delta(x-d) \Big) \\ &= \lim_{d\to 0} \left(\frac{M}{d}\delta(x) - \frac{M}{d}\delta(x-d) \right) \\ &= M \lim_{d\to 0} \frac{\delta(x) - \delta(x-d)}{d} \\ &= M\delta'(x). \end{aligned}$$

Point moments can thus be represented by the derivative of the delta function. Integration of the beam equation again results in piecewise polynomial deflection.