Digital Signal Processing

A Quick Overview

Contents

Author: Martin Lauer, Heidelberg
email: mlauer@opencores.org
Version: 0.1

Chapter 1

Digital Signal Processing

In this chapter, the fundamentals of digital signal processing necessary for the understanding of the following chapters will be reviewed and the properties of various digital filters will be discussed. For further details the reader is referred to the literature [1, 2, 3, 4, 5, 6, 7].

In the following a one-dimensional problem is assumed. The incoming analog signal is represented as a function of a single variable, usually the time t, $f(t)$.

The most important system for digital signal processing is the linear time-invariant system (LTI), because this system is characterized by its impulse response, the output of the LTI system for any input can be derived from the knowledge of the impulse responses.

1.1 Discrete Linear Time-Invariant Systems

A discrete system is represented by the transformation of the input sequence $x[n]$ into the output sequence $y[n]$:

$$\begin{aligned} x[n] &\rightarrow y[n] \\ \delta[n] &\rightarrow h[n] \, , \end{aligned}$$

and the last equation defines the response the single impulse, the impulse response $h[n]$, of the system. The superposition property defines the linear system. If $y[n]$ is the response of a system to an input $x[n]$, then a system is called linear if the response of the system to a superposition of two signals equals the superposition of the superposition of the responses to the individual input components:

$$x_1[n] \rightarrow y_1[n]$$

$$\begin{aligned} x_2[n] &\rightarrow y_2[n] \\ c_1x_1[n] + c_2x_2[n] &\rightarrow c_1y_1[n] + c_2y_2[n] \end{aligned}$$

The time-invariant system is defined by the property that the output is delayed by the same amount of time as the input signal:

$$x[n+d] \rightarrow y[n+d] \, ,$$

which means, that the system is invariant under translation operations. If the input sequence is expressed as

$$x[n] = \sum_k x[k] \, \delta[n-k]$$

the response of the LTI system is determined by the convolution of the input sequence with the impulse response

$$\begin{aligned} y[n] &= \sum_k x[k] \, h[n-k] \\ &= x[k] * h[n-k]. \end{aligned}$$

If two systems are connected in series then the overall impulse response is the convolution [8] of the individual impulse responses $h_{\mathrm{s}}[n] = h_1[n] * h_2[n] = h_2[n] * h_1[n]$. The *inverse* system h_i is defined by $\delta[n] = h[n] * h_i[n] = h_i[n] * h[n]$ meaning that the overall response to an input sequence $x[n]$ is the input sequence itself.

1.2 Analog-Digital (ADC) Conversion

The first step of digital signal processing is the conversion of the analog data $f(t)$ into the digital domain, where the continuous signal f(t) is represented by a finite set of discrete values f[n]. The function will be represented by a grid of finite time increments Δt and finite amplitude[1] steps $\Delta A = 2^{-n}$ with $\Delta t = T = \frac{1}{f_s}$ the inverse of the sampling frequency and n the number of bits of the ADC: $f(t) \rightarrow f(m\Delta t) = f(mT) =: f_m$ with t $\in$ R and m $\in$ Z. Thus the sampling process consists of two parts: the quantization of the signal amplitude and the discretization of the time increments.

[1]The amplitude is normalized to 1.

1.3 Quantization

The *quantization* process introduces an error because of the limited number of amplitudes available for the representation of the input signal. For an ideal ADC with a constant and fixed step size ΔA the variance σ of the rounding error $(A - a)$, with A the true amplitude and a the measured ADC value, due to the quantization process is

$$\sigma^2 = \frac{1}{\Delta A} \int_{A-\Delta A/2}^{A+\Delta A/2} (A - a)^2 da = \frac{1}{3} (\frac{\Delta A}{2})^2 = \frac{\Delta A^2}{12}, \tag{1.1}$$

and thus the average deviation is only from the true amplitude is only $1/\sqrt{3}$ of the maximal rounding error. In order to reduce the error introduced by the quantization process a technique called *dithering* [9, **?**] can be employed. To increase the accuracy of a measured quantity (e.g. the pulse height of a detector signal) the results of multiple measurements of the same quantity are averaged to form the final result. The output of a digitizer, however, will be always the same digital value if the analog input signal is noiseless and prevent any gain in information. In the presence of noise adding to the analog signal the output of the digitizer will span more than one channel. In this case averaging the results of multiple conversions results in a determination of the analog input value with an accuracy *higher* than that given by the quantization steps size. By averaging N measurements the standard deviation of the mean value is given by:

$$\sigma_{<a>} \approx \frac{1}{\sqrt{N}} \sigma_a \ , \tag{1.2}$$

which shows that N has to be chose large enough to (over-)compensated the additional noise added by the dithering process. This property (also called *bit gain*) and its importance for γ energy determination will be discussed again in chapter 1.6.4 when the noise reduction capabilities of pulse shaping networks will be determined in presence of two noise sources.

A more advance technique is the *subtractive dithering*, where a DAC in front of the ADC is used to add a noise pattern to the signal which is subsequently subtracted after the A/D conversions from the ADC output. A similar technique is known as the *sliding scale* method for Wilkinson-type ADCs, which are used in standard nuclear spectroscopy setups.

1.4 Discretisation: Periodic Sampling

The sampling process is represented by the multiplication of the analog signal $f(t)$ with the Shah function [10], which is a sum of delta functions $\delta(t - t_n)$ [11], where $t_n = n\Delta t$

$$f_s(t) = f(t)\, s(t) = f(t) \sum_n \delta(t - t_n)\,, \tag{1.3}$$

with $s(t) = \sum_n \delta(t - t_n)$ the sampling function and the discrete representation f_s of the input signal $f(t)$ is achieved through an integration

$$f_s = \sum_n \int f(t)\, \delta(t - t_n)\, dt = \sum_n f(t_n) = \sum_n f(n\Delta t) \tag{1.4}$$

with $f(t_n) = \int f(t)\, \delta(t - t_n)\, dt$ the sample of $f(t)$ at t_n. Since the Fourier transformation of a multiplication yields a folding operation and the transformation of the Shah function is also a Shah function[2], the corresponding equation for the Fourier transformation of f_s, $F(\omega)$, is

$$F_s(\omega) = \frac{1}{2\pi}(F(\omega) * \omega_s \sum_{m=-\infty}^{\infty} \delta(\omega - \frac{2\pi m}{\Delta t})) \tag{1.5}$$

$$= \frac{\omega_s}{2\pi} \int F(\omega') \sum_{m=-\infty}^{\infty} \delta(\omega - m\omega_s - \omega')\, d\omega' \tag{1.6}$$

$$= \frac{\omega_s}{2\pi} \sum_{m=-\infty}^{\infty} F(\omega - m\omega_s) \tag{1.7}$$

with $\omega_s = \frac{2\pi}{\Delta t} = 2\pi f_s$ and $F(\omega)$ the Fourier transformation of $f(t)$. The individual components of $F_s(\omega)$ are shifted by multiples of ω_s and $F_s(\omega)$ is a superposition of periodic repeating copies of the original frequency spectrum $F(\omega)$. The frequency ambiguity, if $F(\omega)$ covers a wide (infinite) frequency range, is called *aliasing*. Aliasing can only be prevented if the frequency range of $F(\omega)$ is restricted to $< \frac{\omega_s}{2}$, i.e. $F(\omega) = 0$ if $|\omega| \geq \frac{\omega_s}{2}$. This expression is called *Nyquist criterion*, the corresponding frequency $\omega_n = \frac{\pi}{\Delta t}$ the *Nyquist frequency*. For the sampling of low-pass signals the sampling theorem states that the frequency of the conversion process must be twice as high as the highest frequency in the analog data. In order to fulfill the Nyquist criterion, typical ADC systems use an analog low-pass *anti-aliasing* filter prior to the A/D conversion, such that $F'(\omega) = F(\omega)\, LP(\omega)$

[2] $s(t) = \sum_n \delta(t - nT) \to S(\omega) = \omega_s \sum_k \delta(\omega - k\omega_s)$

where $LP(\omega)$ is the low-pass filter employed to fulfill the Nyquist criterion: $LP(\omega) = 0$ for $\omega \geq \frac{\omega_s}{2}$. Such a signal can be perfectly reconstructed from the sampled data points because there's no ambiguity. This is called the *sampling theorem* and the process of reconstructing the continuous signal from the sampled data points, called Witthaker reconstruction, is presented in chapter 1.8.2.

1.4.1 Finite Duration of the Sampling Process

The finite duration of the sampling process can be described by replacing the sum of delta functions with a sum of rectangular impulses of finite width W and height $\frac{T_s}{W}$:

$$s(t) = \begin{cases} T_s/W & , nT_s - \frac{W}{2} \leq t \leq nT_s + \frac{W}{2},\ n \in Z \\ 0 & , \text{otherwise} \end{cases} \tag{1.8}$$

The frequency response is that of the perfect sampling with an additional envelope of the form $\frac{\sin(x)}{x}$. For this we expand the sampling function $s(t)$ into a Fourier series:

$$s(t) = \sum_{m=-\infty}^{\infty} c_m \exp(i\omega_s mt) \tag{1.9}$$

and determine the coefficients c_m of the expansion by

$$\begin{aligned} c_m &= \frac{1}{T_s} \int_{T_s/2}^{T_s/2} s(t) \exp(-i\omega_s mt)\, dt && (1.10) \\ &= \frac{1}{T_s} \int_{W/2}^{W/2} \frac{T}{W} \exp(-i\omega_s mt)\, dt && (1.11) \\ &= \frac{2}{\omega_s mW} \sin\left(\frac{\omega_s mW}{2}\right) && (1.12) \end{aligned}$$

Then as before the sampling process is described by

$$\begin{aligned} f_s(t) &= f(t)\, s(t) && (1.13) \\ &= \sum_{m=-\infty}^{\infty} \frac{2}{\omega_s mW} \sin\left(\frac{\omega_s mW}{2}\right) f(t) \exp(i\omega_s mt). && (1.14) \\ & && (1.15) \end{aligned}$$

With the translation property of the fourier transformation the transfer function is given by

$$F_s(\omega) = \sum_{m=-\infty}^{\infty} \frac{2}{\omega mW} \sin\left(\frac{\omega mW}{2}\right) F(\omega - m\omega_s). \tag{1.16}$$

For $k = \Delta t$ this leads to $F_s(\omega) = F(\omega)$, only the term for m=0 remains and any aliasing is automatically prevented.

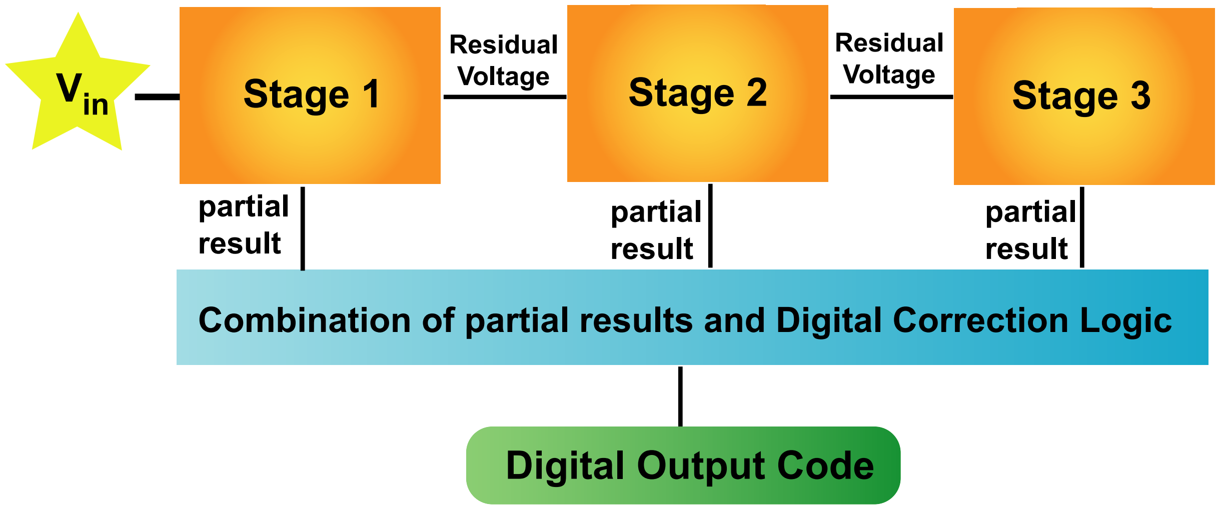
1.5 The Flash ADC Architecture

A typical DSP system consists of a sensor/detector connected to an Analog to Digital Converter (ADC or A/D Converter), whose output is connected to a Digital Signal Processor (DSP). There exist a vast amount different ADC architectures, as an example the architecture of a typical flash ADC is shown in the Fig. 1.1.

The reference voltage is divided down to ground potential using a series of resistors. A comparator compares the input voltage with the corresponding fraction of the reference voltage. If the input voltage is above this threshold of the comparator, the comparator sets its output and the subsequent encoder converts this into the digital ADC word. The output code is a result of the parallel comparison. For a n-bit ADC, $2^n - 1$ comparators are necessary. The precision and stability of the comparator thresholds and the reference voltage determine the quality of the A/D conversion result.

For fast high resolution converters a pipelined scheme is preferred. This converter type consists of several ADCs with less accuracy, i.e. less bits, that are arranged such that the following stage achieves a further refinement of the previous conversion result. Therefore a single ADC stage consists of an ADC and a DAC as shown in Fig. 1.1. An ADC is used for A/D conversion and the DAC translates the ADC result into analog data again, i.e. is responsible for the D/A conversion. The difference between the input voltage and the DAC voltage, the *residual* voltage, is transferred to the following stage which in turn performs the same operations but with the residual voltage only. In order to use identical stages, usually an amplification stage is added in between the pipeline stages. The digital outputs of the individual ADC stages are combined to form the final ADC word as shown in Fig. 1.1.

An advantage of the pipelined flash ADC is the reduced amount of comparators needed. For example a two stage pipelined 12 bit ADC needs only $2^6 - 1 + 2^6 - 1 = 2^7 - 2$ comparators compared to $2^{12} - 1$ of the non-pipelined version. However, the first ADC stage still has to have the full precision. For a two stage ADC with 6 bits per stage this means that if the threshold



(a) Pipelined flash ADC with error correction.

Figure 1.1: Single ADC stage of a pipeline flash ADC

of a comparator of the first ADC deviates by $\frac{1}{64}$ LSB[3] from the proper value then the ADC will show a missing code, i.e. the corresponding ADC result cannot be achieved, because the residual voltage exceeds the input range of following stage, which will therefore have a constant conversion result. Therefore pipelined flash ADCs implement a scheme called *error correction* to prevent missing ADC codes. Typically this works by increasing the input range of the following stage, e.g. by 1 bit, such that the residuum of the preceding stage never exceeds the input range. Now, the additional codes of the ADC can be used to detect and subsequently correct the errors of previous stage.

1.6 Digital Filter

Two types of digital filters will be presented in the following: the infinite impulse response (IIR) and the finite impulse response (FIR) filter. The difference between both filters is the use of feedback for the IIR filters, causing an infinite impulse response. Both types of filters were used and are therefore described in more detail.

[3]Least Significant Bit

1.6.1 Transfer Function

The properties of digital filters are usually displayed by plotting the magnitude $M(\omega) = |H(\omega)|$ and phase response $\Phi(\omega) = tan^{-1}(\frac{\Im(H(\omega))}{\Re(H(\omega))})$ of the, usually complex, transfer function $H(\omega) = \frac{Y(\omega)}{X(\omega)}$, which is defined as the ratio between output and input spectrum. With the *shift* theorem of the Fourier transformation[4], it is apparent that any transfer function $H(\omega)$ that can be expressed as $H(\omega) = \hat{H}(\omega)\exp(i\omega t_d) = \hat{H}(\omega)\exp(i\,\Phi(\omega))$ with $\hat{H}(\omega)$ a real function of the frequency ω has a linear phase and acts as a constant delay in the time domain. The magnitude response $|H(\omega)|$ is not affected by the linear phase term. A zero delay is achieved for real transfer functions only. The *group delay* $D(\omega)$ is defined as $D(\omega) = \frac{\partial}{\partial\omega}\Phi(\omega)$ and for the above example a group delay of $D(\omega) = t_d$ is obtained. The *phase delay* $P(\omega) = \frac{\Phi(\omega)}{\omega} = t_d$ is equal to the group delay in case of a linear phase. Since pulse shape analysis is about the extraction of timing information from the digitized detector signal, linear phase filters are preferred, because otherwise the pulse shape is distorted.

1.6.2 IIR filter

Infinite impulse response (IIR) filters are defined by the use of feedback, meaning that the output not only depends on the previous inputs but also on the previous filter outputs. Du to this remembrance of past outputs it is possible that the output of an IIR filter stays non-zero even if all the following input samples are all zero, resulting in an infinite impulse response. The equation describing a general IIR filter is

$$y[n] = \sum_{m=0}^{M} a_m x[n-m] + \sum_{m=1}^{M} b_m y[n-m] \tag{1.17}$$

The general expression for the frequency response $H(\omega)$ of IIR filters is

$$\begin{aligned} H(\omega) &= \frac{\sum_{k=0}^{N} a_k z^{-k}}{1 - \sum_{k=1}^{N} b_k z^{-k}} && (1.18)\\ &= \frac{\sum_{k=0}^{N} a_k \exp(-ik\omega)}{1 - \sum_{k=1}^{N} b_k \exp(-ik\omega)} && (1.19)\\ &= \frac{\sum_{k=0}^{N} a_k \cos(k\omega) - i\sum_{k=0}^{N} a_k \sin(k\omega)}{1 - \sum_{k=1}^{N} b_k \cos(k\omega) - i\sum_{k=1}^{N} b_k \sin(k\omega)}\,, && (1.20) \end{aligned}$$

[4] $f(t - t_d) \;\rightarrow\; F(\omega)\exp(i\omega t_d)$

where the first equation correspond to the z-transform. A zero phase response can be achieved with IIR filter by filtering the data twice, first in forward direction and then in backward direction. This time-reversal causes a zero phase as can be directly seen from the shift property of the Fourier transformation

$$f(t-d) \rightarrow H_1(\omega) = \hat{H}_1 \exp(-i\omega d) \quad (1.21)$$

$$f(t+d) \rightarrow H_2(\omega) = \hat{H}_2 \exp(i\omega d) \quad (1.22)$$

$$H(\omega) = H_1(\omega) H_2(\omega) \quad (1.23)$$

$$= \hat{H}_1 \hat{H}_2 \quad (1.24)$$

Exponential Averaging

An example for an IIR filter is the exponential averaging (EA) filter. The EA filter is a simple IIR filter with a low-pass behavior, used e.g. for noise reduction in the following aplications. The mathematical equation for this IIR filter is

$$y[n] = \alpha x[n] + (1-\alpha) y[n-1] \ , \quad (1.25)$$

with α the relaxation constant, a constant weighting factor that controls the amount of feedback. It can be seen from the filter equation 1.25 that only one storage element is needed to hold the previous filter output sample. Furthermore the algorithm requires only one multiplications if arranged such that $y[n] = \alpha(x[n] - y[n-1]) + y[n-1]$.

If α is one then the input is not attenuated and previous filter output is ignored. With decreasing α the input gets more and more attenuated and the influence of previous filter output increases, resulting in an increased smoothing and slower step response. The effect of different α on the impulse and step response is plotted in Fig. 1.3.

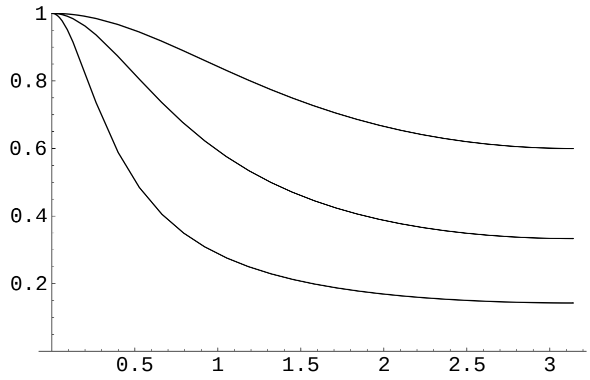
The noise variance reduction of the exponential averaging is given by [4]

$$\frac{\text{output noise variance}}{\text{input noise variance}} = \frac{\alpha}{2-\alpha}. \quad (1.26)$$

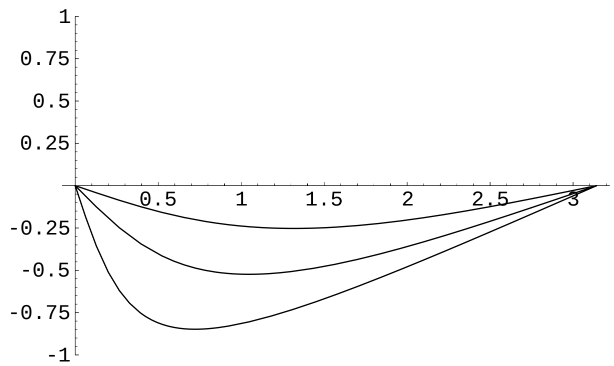
From equation 1.18 the transfer function of the exponential averaging is obtained by setting $N = 0$, $M = 1$, $a_0 = \alpha$ and $b_1 = 1 - \alpha$

$$H(\omega)_{\text{exp}} = \frac{\alpha}{1 - (1-\alpha)\cos(\omega) + i(1-\alpha)\sin(\omega)} \ . \quad (1.27)$$

The magnitude response as well as the phase response are plotted in Fig. 1.2.

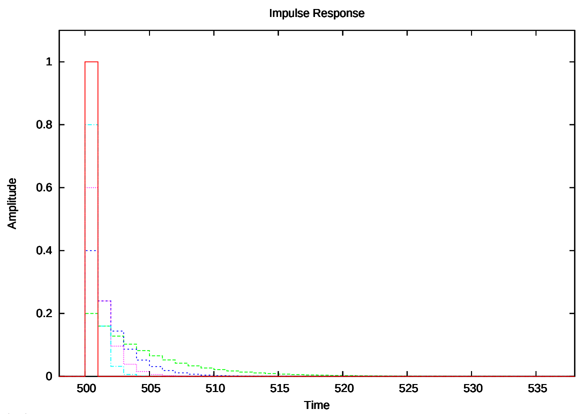


(a) Magnitude response of the exponential averaging.

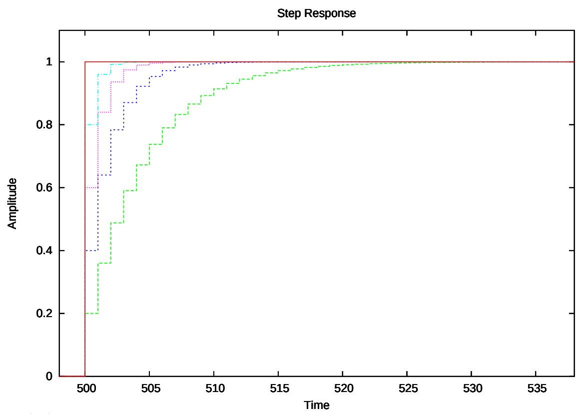


(b) Phase response of the exponential averaging.

Figure 1.2: Magnitude and phase response of the exponential averaging.



(a) Impulse response of the exponential averaging. From the impulse response it is clearly visible why the filter is called exponential averaging.



(b) Step response of the exponential averaging.

Figure 1.3: Impulse response of the exponential averaging. From the impulse response it is clearly visible why the filter is called exponential averaging.

1.6.3 FIR filter

On contrary to the IIR filter, Finite Impulse Response (FIR) filters are always stable[5] because of the lack of any feedback. The general expression for FIR filters is

$$y[n] = \sum_{k=0}^{N} c_k x[n-k]\,. \tag{1.28}$$

The output sequence $y[n]$ of a FIR filter is equal to the convolution of the input sequence with the filter's impulse response, which is given by the filter coefficients, and therefore the frequency response of the FIR filter is given by the Discrete Fourier Transform (DFT) of the impulse response (coefficients):

$$H(\omega) = \sum_{k=0}^{N} c_k \exp(-ik\omega) = \sum_{k=0}^{N} c_k \cos(k\omega) - i \sum_{k=0}^{N} c_k \sin(k\omega). \tag{1.29}$$

A linear phase response of FIR filters can easily be achieved, by writing equation 1.28 in a symmetrical way $y[n] = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} c_k x[n+k]$ such that $c_k = c_{-k}$.

The Ideal Low Pass Filter

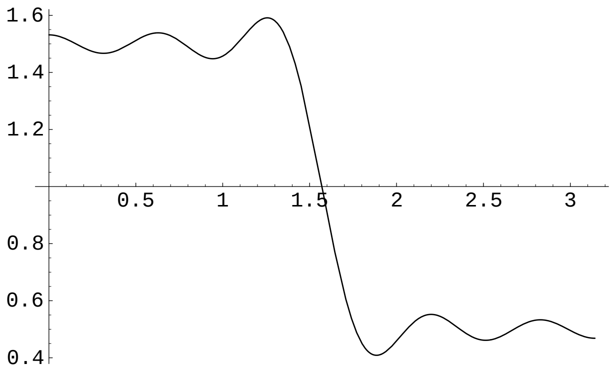
The ideal low-pass filter is an example for a FIR filter with zero phase shift. It has a rectangular transfer function

$$H(\omega) = \begin{cases} 1 & , |\omega| < \omega_c) \\ 0 & , |\omega| \geq \omega_c) \end{cases} \tag{1.30}$$

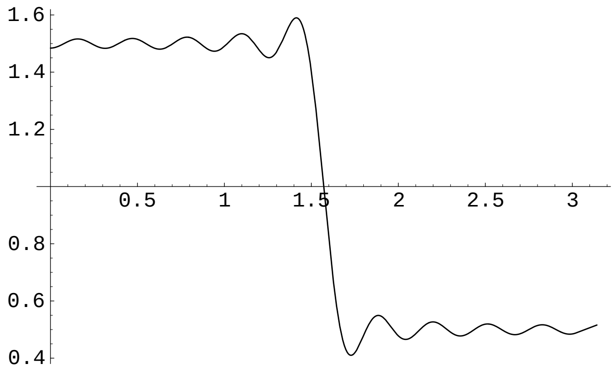
and the coefficients c[n](impulse response h[n]) of the FIR filter are given by the Fourier transformation of the transfer function

$$\begin{aligned} c[n] = h[n] &= \frac{1}{2\pi}\int_{-\infty}^{\infty} H(\omega)\exp(i\omega n)\, d\omega && (1.31) \\ &= \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} \exp(i\omega n)\, d\omega && (1.32) \\ &= \frac{\sin(\omega_c n)}{\pi n}\,, && (1.33) \end{aligned}$$

[5]If the response to a bound input sequence $|x[n]| \leq B_x < \infty$ is also bound for all n, i.e. $|y[n]| \leq B_y < \infty$, the system is called stable.



(a) $N = 9$



(b) $N = 19$

Figure 1.4: Transfer functions of the ideal low-pass filter for different filter length N.

with ω_c = the cut-off frequency. In case the ideal low-pass filter is implemented as a digital filter the cut-off frequency is defined relative to the sampling frequency $\omega_c = 2\pi\frac{f_c}{f_s}$. For a digital filter, only a finite number of $2N + 1$ coefficients can be used for the implementation. Therefore the transfer function is only an approximation of the ideal low-pass transfer function as can be seen from the discrete Fourier transformation of the FIR implementation of the low-pass

$$H_N(\omega) = \sum_{n=-N}^{N} \frac{\sin(\omega_c n)}{\pi n} \exp(-i\omega n) \,. \tag{1.34}$$

$H_N(\omega)$ is plotted in figure 1.4 for $N = 9$ and $N = 19$.

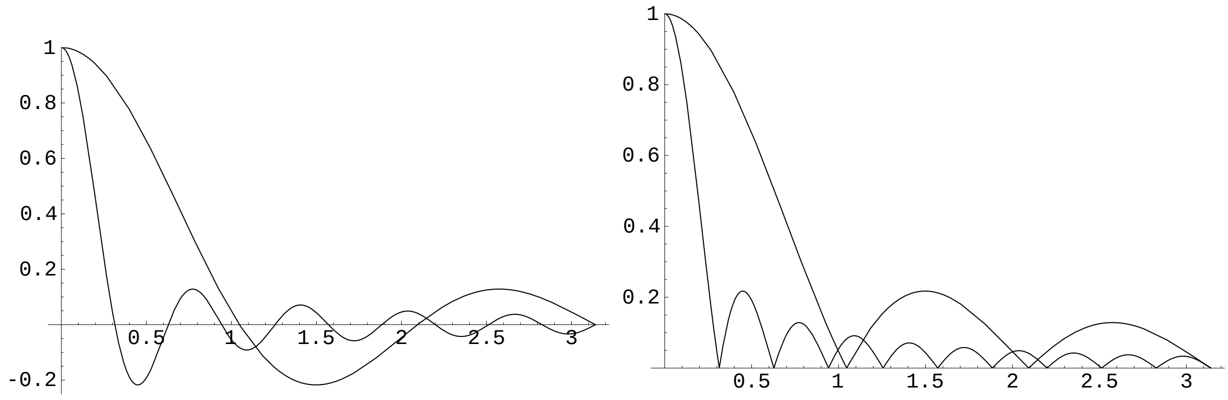
Moving Average

A moving average (MA) filter is a very simple to realize in hardware after digitization and at the same time an effective low-pass filter. Therefore this filter is presented as a second example of a FIR filter. The MA filter adds $L = 2N + 1$ samples with a constant weight

$$c[n] = \begin{cases} \frac{1}{2N+1} & , |n| \leq N \\ 0 & , |n| > N \end{cases} \tag{1.35}$$

$$y[n] = \sum_{i=-\infty}^{\infty} c[i]\, x[n+i] \tag{1.36}$$

$$= \frac{1}{2N+1} \sum_{i=-N}^{N} x[n+i] \,. \tag{1.37}$$



(a) Transfer function

(b) Magnitude response

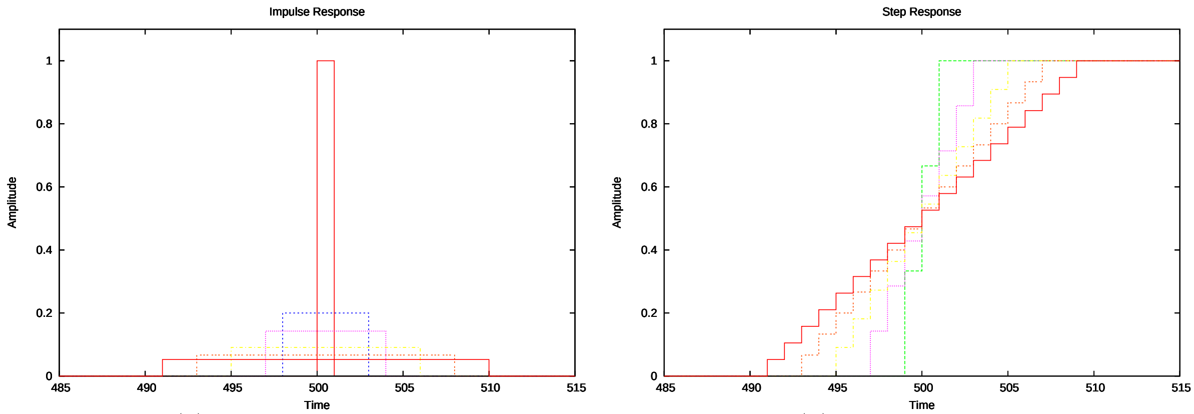
Figure 1.5: Transfer function and magnitude response of the moving averaging operation for two different filter lengths N.

The transfer function can be determined from Fourier transformation of the filter coefficients

$$\begin{aligned} H(\omega) &= \int_{-\infty}^{\infty} c(t)\exp(-i\omega t)\,dt && (1.38)\\ &= \frac{1}{2N\Delta t}\int_{-N\Delta t}^{N\Delta t}\exp(-i\omega t)\,dt && (1.39)\\ &= \frac{1}{2T_N}\int_{-T_N}^{T_N}\exp(-i\omega t)\,dt && (1.40)\\ &= \frac{\sin(\omega\,T_N)}{\omega\,T_N} && (1.41)\end{aligned}$$

which is plotted in figure 1.5[6] for $N=3$ and $N=10$. Whereas the ideal low-pass has a box-like transfer function, the MA has a box-like behavior in the time-domain. The transfer function of the MA shows an oscillating behavior, i.e. the attenuation does not constantly increase with frequency. However, if the signal power is concentrated mainly in the lower frequency range or the signal was oversampled, then the MA filter is the preferred low-pass filter because of its simple structure. The MA operation can also be expressed as a recursive filter $y[n]=y[n-1]+\frac{1}{m}(x[n]-x[n-m])$, which is the preferred equation for the implementation. The length of the MA filter can be adjusted by changing the parameter m.

[6] With the DFT the transfer function is $H(\omega) = \frac{1}{2N+1}\sum_{-N}^{N}\exp(-i\omega t) = \frac{1}{2N+1}\frac{\exp(i\omega N)-\exp(-i\omega(N+1)))}{1-\exp(-i\omega)} = \frac{1}{2N+1}\frac{\sin(\omega(2N+1)/2)}{\sin(\omega/2)}$



(a) Impulse response. (b) Step response.

Figure 1.6: Impulse and step response of the moving averaging filter.

Binomial Filter

As a third example of a low-pass FIR filter, the binomial filter will be discussed. The binomial filter is created by repeated convolution of a two point MA filter $y[n] = \frac{1}{2}(x[n] + x[n-1])$ with itself $BF = MA_2 * MA_2 * \ldots * MA_2$. For example the convolution of a step function with itself leads to a triangular shape and therefore the resulting binomial filter is $y[n] = MA_2 * MA_2 = \frac{1}{4}(x[n+1] + 2x[n] + x[n-1])$. Generally, if this process is repeated n times then the transfer function is that of the two point MA to the n-th power, i.e. $H(\omega) = \left(\frac{cos(\omega/2)}{2}\right)^n$. On contrary to the MA filter, the binomial filter shows no oscillating behavior and is therefore better low-pass filter than the MA filter. However, the filter coefficients have to stored and the filter requires multiplications with fractional precision, whereas in case of the MA filter the multiplication can be replaced by a simple shift operation of the filter length L is chosen in powers of two, i.e $L = 2^m$.

Least-Squares Smoothing - Trend Analysis

The performance of the previous FIR filters was judged using the transfer function. Now, a FIR filter will be derived on the basis of the principle of least-squares[7], stating that the best approximation $a[n]$ of the data sequence $x[n]$ is that one minimizing the squares of the deviations of the data sequence from their estimate $\sum_{n=0}^{M}(x[n] - a[n])^2$. The detector signal can be modeled as the sum of two contributions, the detector current signal and the noise.

[7] Mathematicians believe its a physical principle while physicists believe its a mathematical principle.

The noise dominates the signal at high frequencies and in order to extract the signal trend the filter should remove the high-frequency noise. The smoothing of the data based on the principle of least squares is therefore also regarded as trend analysis [?].

The detector signal $y[k]$ is modeled as the sum of the detector and noise contribution $y[k] = t[k] + n[k]$, with $t[k]$ the signal trend and $n[k]$ the noise. The FIR filter

$$t[k] = \sum_{r=-M}^{M} a_r \, y[k+r] \, , \tag{1.42}$$

to extract the signal trend $t[n]$ should be symmetric and have a gain of one and therefore the coefficients a_r have to fulfill the following conditions

$$\sum_{r=-M}^{M} a_r = 1 \tag{1.43}$$

$$a_r = a_{-r} \, . \tag{1.44}$$

The filter coefficients are determined using the following procedure [?]. The signal $y[k] = t[k] + n[k] \approx p[k] + n[k]$ is represented as the sum of a polynomial $p[k]$, which is used to approximated the signal trend, and the noise $n[k]$. In the following the input data sequence will be represented in the form of a (transposed) vector $\mathbf{y}^T = (y[-M], y[-M+1], ..., y[0], y[M])$, i.e. $\mathbf{y}$ is the data vector for the desired FIR filter of length $2M + 1$. The polynomial $\mathbf{p}$ of degree n is represented as $p[k] = b[0] + b[1]k + b[2]k^2 + ...b[n]k^n$ with $\mathbf{b}^T = (b[0], b[1], ..., b[n])$ the coefficients of the polynomial. This leads to $2M + 1$ equations for $\mathbf{p}$

$$p[-M] = b_0 + b_1(-M) + b_2(-M)^2 + ... + b_n(-M)^n \tag{1.45}$$

$$\vdots \tag{1.46}$$

$$p[0] = b_0 + b_1 0 + b_2 0^2 + ... + b_n 0^n \tag{1.47}$$

$$\vdots \tag{1.48}$$

$$p[M] = b_0 + b_1 M + b_2 M^2 + ... + b_n M^n \tag{1.49}$$

that can be expressed as $\mathbf{y} = \mathbf{F}\mathbf{b} + \mathbf{n}$ with

$$F = \begin{pmatrix} 1 & -M & (-M)^2 & \dots & (-M)^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & M & M^2 & \dots & M^n \end{pmatrix}, \qquad (1.50)$$

a $(2M+1)$x$(M+1)$ matrix.

The unknown coefficients $\mathbf{b}$ can be determined from the least-squares principle by requiring that the desired coefficients $\widehat{\mathbf{b}}$ minimize the squares of the deviation of the polynomial from the data points $(\mathbf{y}-\mathbf{p})^T(\mathbf{y}-\mathbf{p}) = (\mathbf{y}-\mathbf{F}\widehat{\mathbf{b}})^T(\mathbf{y}-\mathbf{F}\widehat{\mathbf{b}})$. The result is [?] $\widehat{\mathbf{b}} = (\mathbf{F}^T\mathbf{F})^{-1}\mathbf{F}\mathbf{y}$. The coefficients a_r of the filter can be obtained with the additional requirement that $p[0]$ models the signal at $t = 0$: $y[0] = p[0] + z[0]$ with $p[0] = \mathbf{e}_1^T\widehat{\mathbf{b}} = \widehat{b}[0]$ with $\mathbf{e}_1^T = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix}$. The polynomial $p[0]$ can now be expressed as $p[0] = \mathbf{e}_1^T\widehat{\mathbf{b}} = \mathbf{e}_1^T(\mathbf{F}^T\mathbf{F})^{-1}\mathbf{F}\,\mathbf{y}$. If the order of the polynomial and the window size is constant, then the Matrix $\mathbf{F}$ is constant, thus allowing the replacement $p[0] = \mathbf{a}^T\mathbf{y}$ with $\mathbf{a}^T$ the filter coefficients $\mathbf{a}^T = \mathbf{e}_1^T(\mathbf{F}^T\mathbf{F})^{-1}\mathbf{F} = \begin{pmatrix} a[-M] & \dots & a[0] & \dots & a[M] \end{pmatrix}$.

For a linear signal trend the FIR filter obtained with the principle of least squares equals a MA filter. This means that the MA filter is the best filter to approximate a noisy data sequence that features either constant or shows a linear rising or falling signal trend.

For a quadratic polynomial $\mathbf{p}$ and a window size $M = 3$ the corresponding filter coefficients are $a_r = \frac{1}{21}\begin{pmatrix} -2 & 3 & 6 & 7 & 6 & 3 & -2 \end{pmatrix}$.

1.6.4 Trapezoidal Shaping

The properties of the trapezoidal shaping were derived in [12, 13] and are summarized in the following. An extensive overview over the field of signal processing with semiconductor detectors is given in [14]. The noise analysis will be carried out in the time domain [15] and for the derivation two sources of noise were assumed neglecting other noise sources like $\frac{1}{f}$ noise. The only kinds of noise in this model are voltage[8] and current[9] noise. The first

[8] Voltage noise is often referred to as delta or series noise.

[9] Current noise is often referred to as step noise or parallel noise.

noise source in the model is a current sources in parallel to the detector, the signal source. The sum of these two contributions is integrated on the input capacity, i.e. detector and preamplifier capacity. In this setup the signal to noise ratio is independent of the input capacity ($\frac{q_s}{q_n} = \frac{i_s/C}{i_n/C}$). The second noise source is connected in series to the signal after the input capacity and is therefore modeled as voltage noise generator with (constant) amplitudes independent of C which leads to a signal-to-noise (SNR) ratio proportional to $\frac{1}{C}$.

Another assumption of the model is that the measurement takes place at a fixed time T_{ps} after signal start. In order to determine the cumulative effect of all noise steps prior to the time when the peak value is sampled the so-called step-noise residual function $R(t)$, which represents the effect of a single noise step that has happened $-t$ before peak capture, is used. The total mean square step and delta noise indices are then defined as

$$\langle N_s^2 \rangle = \frac{1}{S^2} \int (R(t))^2 \, dt \quad (1.51)$$

$$\langle N_\Delta^2 \rangle = \frac{1}{S^2} \int (R'(t))^2 \, dt \,, \quad (1.52)$$

with S the signal amplitude. For the example of a trapezoidal shaper with a rise and fall time of T_1 and T_2, a gap time of T_F and amplitude of $S = 1$ the corresponding noise indices are

$$\langle N_s^2 \rangle = \int_0^{T_2} \left(\frac{t}{T_2}\right)^2 dt + \int_0^{T_\mathrm{F}} (1)^2 \, dt + \int_0^{T_1} \left(\frac{t}{T_1}\right)^2 dt \quad (1.53)$$

$$= T_\mathrm{F} + \frac{\mathrm{T}_1 + \mathrm{T}_2}{3} \quad (1.54)$$

$$\langle N_\Delta^2 \rangle = \int_0^{T_2} \left(\frac{1}{T_2}\right)^2 dt + \int_0^{T_1} \left(\frac{1}{T_1}\right)^2 dt \quad (1.55)$$

$$= \frac{1}{T_1} + \frac{1}{T_2} \quad . \quad (1.56)$$

The previous equations show that the step noise contribution increases in proportion to the length of the filter rise and gap time whereas the delta noise index is inverse proportional to the filter rise time. Therefore the delta noise contribution can be reduced by increasing the peaking time of the trapezoidal shaper. Furthermore, equation 1.56 suggests that symmetrical pulse shapes yield better noise indices for a fixed total filter length.

Another important feature of the trapezoidal shaping is the insensitivity to rise time variations of the detector signal due to the flat top, eliminating the so-called ballistic deficit [16] which is limiting the energy resolution for high energy γ rays. This will be explained in section **??**.

Furthermore the trapezoidal shaper is ideally suited for high rate applications because of the finite with of the pulse shape, which is two times the peaking time plus the gap time. The gaussian shaper used in standard analog electronics shows a slow return to the baseline after an event which disturbes the baseline determination and limits the performance at high event rates.

1.7 Numerical Differentiation

Numerical differentiation is an important operation for PSA because not only the charge signal contains information about the interaction position, but also the current signal and even the derivative of the current signal. For example, the radial interaction position information for the MINIBALL detector is extracted from the second derivative of the digitized charge pulse.

In general differentiation leads to an decreased signal to noise ration (SNR). The signal amplitude decreases whereas especially the high frequency noise is almost unaffected.

The transfer function is obtained by applying the differentiation to an single frequency signal $\exp(i\omega t)$ and divide the result by the input signal

$$H(\omega) = \frac{\frac{d}{dt}\exp(i\omega t)}{\exp(i\omega t)} = i\omega \tag{1.57}$$

The transfer function is proportional to the frequency, demonstrating the high-pass behavior of the differentiation operation.

1.7.1 Differentiability of bandwidth limited Signals

A bandwidth limited signal can be represented in the time domain by [17]

$$x(t) = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} X(\omega)\exp(i\omega t)\, d\omega. \tag{1.58}$$

The nth differentiation of $x(t)$

$$\frac{d^n}{dt^n}x(t) = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} X(\omega)(i\omega)^n\exp(i\omega t)\, d\omega. \tag{1.59}$$

features an additional term $(i\omega)^n$ which could prevent the integral from convergence, but because of the bandwidth limitation, the integral always converges and therefore differentiation of bandwidth limited signals is possible.

1.7.2 Differentiation Equations

The most simple approach to numerical differentiation leads to differences of the forward, backward or central type

$$\begin{aligned} d[n]_b &= x[n] - x[n-1] && (1.60) \\ d[n]_f &= x[n+1] - x[n] && (1.61) \\ d[n]_c &= \frac{1}{2}(x[n+1] - x[n-1]) && (1.62) \\ &= \frac{1}{2}(d[n]_f + d[n]_b)\,. && (1.63) \end{aligned}$$

The advantage of the differentiation of central type, is that the filter is symmetric and therefore has a linear phase. The differentiation of forward and central type are non-causal, in the sense that the output of the differentiation filter depends on future samples. The corresponding transfer functions can be determined by applying the differentiation equations onto a single frequency signal $\exp(i\omega t)$

$$\begin{aligned} H_b[\omega] &= 2i\sin\left(\frac{\omega}{2}\right)\exp\left(-i\frac{\omega}{2}\right) && (1.64) \\ H_f[\omega] &= 2i\sin\left(\frac{\omega}{2}\right)\exp\left(i\frac{\omega}{2}\right) && (1.65) \\ H_c[\omega] &= i\sin(\omega)\,. && (1.66) \end{aligned}$$

The transfer functions are both complex and magnitude and phase response are plotted in Fig. 1.7. The differentiation is a good approximation for the true transfer function for small frequencies ω , but high frequencies are attenuated compared to the transfer function of the differentiation.

Similarly the transfer function for the second derivative

$$sd[n] = x[n+1] - 2x[n] + x[n-1] \qquad (1.67)$$

is determined to

$$H_{sd}[\omega] = 2(1 - \cos(\omega))\,, \qquad (1.68)$$

which is also a good approximation only at low frequencies[10], while high frequencies are also attenuated compared to the true transfer function $H(\omega) = -\omega^2$ which can be seen from figure 1.10.

An alternative approach to numerical differentiation [3] is to start from the desired transfer function of the differentiation $H(\omega) = i\omega$ and design a matching FIR filter with the desired transfer function. Assuming a FIR filter of the form

$$y[m] = \sum_{n=-N}^{N} c[n]\, x[m-n] \tag{1.69}$$

with $c[n]$ the coefficients that have to be determined for the desired transfer function from the coefficient equation

$$c[n] = \frac{1}{2\pi}\int_{-\infty}^{\infty} H[\omega]\exp(in\omega)\, d\omega\ . \tag{1.70}$$

For the transfer function of the differentiation this leads to the following expression for the coefficients:

$$\begin{aligned} c[n] &= \frac{1}{2\pi}\int_{-\omega_s/2}^{\omega_s/2}(i\omega)\ \exp(in\omega)\, d\omega && (1.71)\\ &= \frac{1}{2\pi}\left(\omega_s\frac{\cos(n\omega_s/2)}{n} - 2\frac{\sin(n\omega_2/2)}{n^2}\right) && (1.72)\\ &= \frac{\cos(n\pi)}{n} - \frac{\sin(n\pi)}{\pi n^2}\ , && (1.73)\end{aligned}$$

where $\omega_s = 2\pi$ was used. This leads to the following coefficients

$$\begin{aligned} c[0] &= 0 && (1.74)\\ c[n] &= \frac{(-1)^n}{n}\ \text{for n } \neq 0. && (1.75)\end{aligned}$$

The corresponding filter equations are:

$$\begin{aligned} c_3[n] &= x[n+1] - x[n-1]\\ c_7[n] &= \left(\frac{1}{3}x[n+3] - \frac{1}{2}x[n+2]\right)\\ &+ x[n+1] - x[n-1]\\ &+ \left(\frac{1}{2}x[n-2] - \frac{1}{2}x[n-3]\right)\end{aligned}$$

[10] $H_{sd}[\omega] \approx 2(1-(1-\omega^2/2+\ldots))$

$$
\begin{aligned}
c_{11}[n] &= \left(\frac{1}{5}x[n+5] - \frac{1}{4}x[n+4]\right) + \left(\frac{1}{3}x[n+3] - \frac{1}{2}x[n+2]\right) \\
&+ x[n+1] - x[n-1] \\
&+ \left(\frac{1}{2}x[n-2] - \frac{1}{2}x[n-3]\right) + \left(\frac{1}{4}x[n-4] - \frac{1}{5}x[n-5]\right) .
\end{aligned}
$$

As an example, the transfer function of the differentiation filter with $N = 11$ is plotted in figure 1.9 together with the ideal transfer function of the differentiation to demonstrate the approximation of the true transfer function.

More complex differentiation formulas are obtained by differentiating lagrangian interpolation formulas [6]

$$
\begin{aligned}
d_{l5}[n] &= \frac{1}{12}(-x[n+2] + 8x[n+1] - 8x[n-1] + x[n-2]) + \frac{h^4}{30}f^4(\lambda) \\
d_{l7}[n] &= \frac{1}{60}(x[n+3] - 9x[n+2] + 45x[n+1] \\
&- 45x[n-1] + 9x[n-2] - x[n-3]) + \frac{h^6}{160}f^{vii}(\lambda) ,
\end{aligned}
$$

where $d_{l5}[n]$ and $d_{l7}[n]$ are obtained from five and seven point interpolation formulas respectively. The last term is the error, which decreases with decreasing step size, i.e with increasing sampling frequency[11]. The transfer functions are

$$
\begin{aligned}
H_{l5}(\omega) &= \frac{1}{12}\left(16i\sin(\omega) - 2i\sin(2\omega)\right) \\
H_{l7}(\omega) &= \frac{1}{30}\left(45i\sin(\omega) - 9i\sin(2\omega) + i\sin(3\omega)\right) ,
\end{aligned}
$$

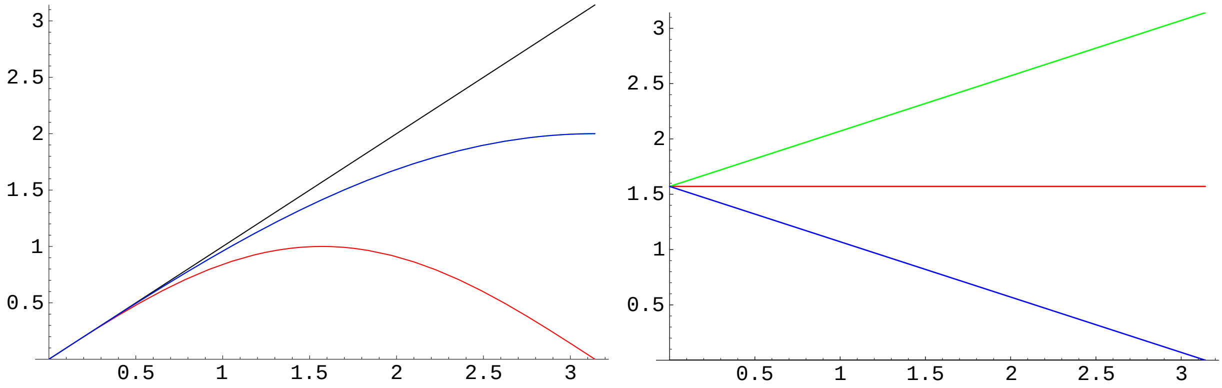
which are plotted in figure 1.11.

A better differentiation formula for bandwidth limited signals should be obtained by a smooth interpolation, i.e. an interpolation without discontinuities[12]. This should especially be true if the derivatives of the interpolated data sequences are of importance. Therefore a differentiation formula d_s based on a spline interpolation [6] of fourth order using 7 data points

$$
\begin{aligned}
d_{s7}[n] &= \frac{1}{168}(3x[n+3] - 26x[n+2] + 127x[n+1] \\
&- 127x[n-1] + 26x[n-2] - 3x[n-3]) ,
\end{aligned}
$$

[11]The error for the $d_c[n]$ differentiation formula is $\frac{h^2}{6}f'''(\lambda)$.

[12]The smoothness of the spline interpolation is shown in [5] in comparison to a polynomial interpolation.



(a) Magnitude response (b) Phase response

Figure 1.7: Magnitude and phase response of the numerical differentiation of forward (green), backward (bue) and central (red) type together with that of the ideal differentiation (black).

Figure 1.8: Current Signals obtained from a charge pulse by differentiation with various digital filters.

and the corresponding transfer function

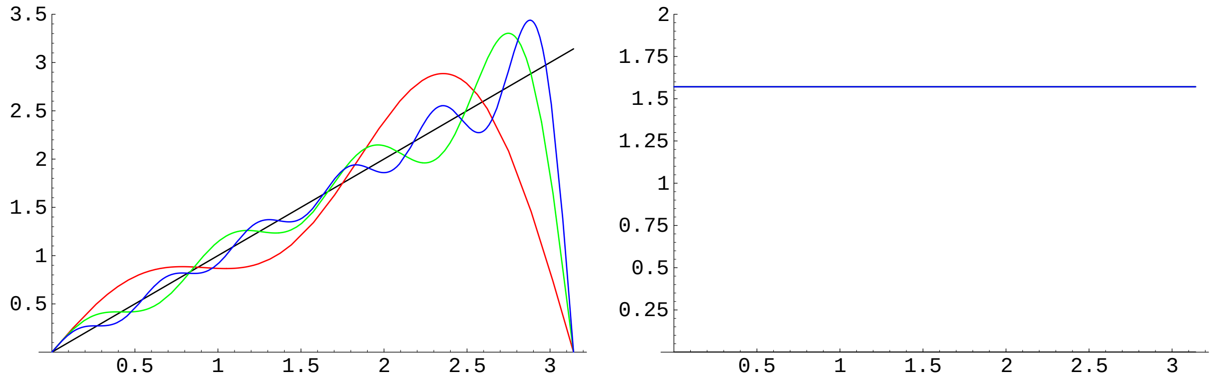
$$H_{s7}(\omega) = \frac{1}{84}\left(127i\sin(\omega) - 26i\sin(2\omega) + 3i\sin(3\omega)\right) \ , \tag{1.76}$$

are mentioned here.

1.8 Digital Resampling: Decimation and Interpolation

Digital resampling[1, 4, 17, 18, 19] is the process of changing the sampling rate *after* A/D conversion. The process of decreasing the sampling rate is called *decimation* and the process of increasing the sampling rate is called *interpolation*.

In order to understand the effect of an digital increase of the sampling rate, it is best to consider a two channel DSP system with 40 MHz ADCs fed by a common clock. Single frequency sine waves are assumed to be used as input signals. The purpose of the experiment is the determination of the phase shift or delay between the two sine waves for DSP channels. If 50 MHz sine waves are used the experiments will fail because the signal will be removed by the Nyquist filter. If 5 MHz sines waves are used the sine wave will be sampled with an oversampling ratio of 4. The phase shift



(a) Magnitude response (b) Phase response

Figure 1.9: Transfer function of the differentiation filter for different N, $N = 3$ (red), $N = 7$ (green) and $N = 11$ (blue), compared to that of the ideal differentiation (black).

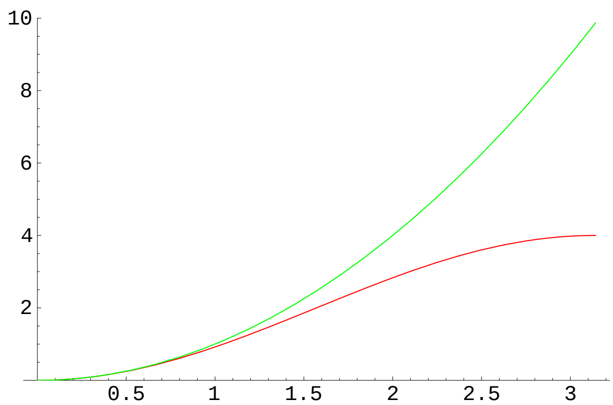
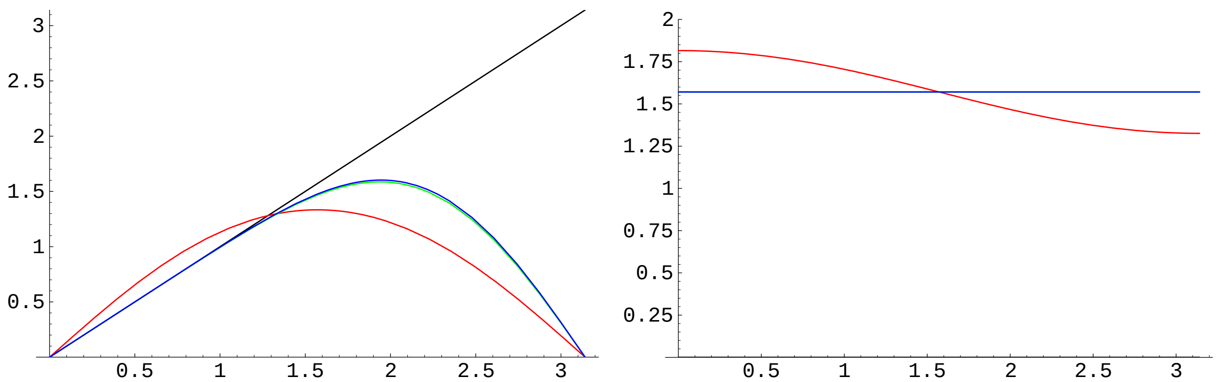


Figure 1.10: Transfer function of the numerical second derivative (red) compared to that of the ideal differentiation (green).



(a) Magnitude response (b) Phase response

Figure 1.11: Transfer function of numerical differentiation formulas based on lagrangan, $d_{l5}[n]$ (red) and $d_{l7}[n]$ (green), and spline,$d_{s7}[n]$ (blue), interpolation in comparison to the ideal differentiation (black).

Figure 1.12: Original and decimated detector signal.

will be determine by measuring the time difference when the signal crosses a certain threshold (or use the detection of the zero crossing) between the two channels. If the signal is resampled to 160 MHz, i.e. the signal is oversampled by a factor of 16, then the phase difference between the two channels can be determined with a four time higher precision, because the time difference between the samples decreased by a factor of four.

The main purpose of digital resampling is the simplification of the further digital signal processing.

1.8.1 Decimation

The process of *downsampling* an ADC output sequence $x[n] = x(nT)$ works by simply sampling the original output sequence at a new rate MT:

$$x_d[n] = x[nM]\,. \tag{1.77}$$

However, if the bandwidth of the original data exploits the full range allowed by the Nyquist theorem, then the new data sequence will be subjected to aliasing, because now the original frequency band will overlap with the mirror signals at multiples of the new sampling frequency $\omega_s^d = \frac{\omega_s}{M}$. Therefore a system is needed that consists not only of a resampler but also of a low-pass filter applied to the original data sequence that ensures that aliasing is prevented for the new data sequence, i.e $H(\omega) = 0$ for $\omega > \omega_d$ with $\omega_c^d = \frac{\omega_c}{M}$ the Nyquist frequency of the decimated signal. Such a system, a low-pass filter followed by a downsampler, is called *decimator*. The reason for aliasing can be seen from the Fourier transformation of the downsampling processes by replacing the frequencies in equation 1.7 with the down sampled frequencies

$$F_k(\omega) = \frac{\omega_s}{2M\pi} \sum_{k=-\infty}^{\infty} F((\frac{\omega}{M}) - k(\frac{\omega_s}{M})) \tag{1.78}$$

$$= \frac{\omega_c^d}{\pi} \sum_{k=-\infty}^{\infty} F(\frac{\omega}{M} - k\omega_s^d). \tag{1.79}$$

Again, the resulting spectrum features copies of the original signal, which is scaled in frequency by the factor M, with a period of $\omega_s^d = \frac{\omega_s}{M}$. A decimation unit will be presented a section **??**.

1.8.2 Upsampling

Similarly to the decimation process which is used for data rate reduction, a process for increasing the data rate of the ADC can be developed. The process of upsampling should yield a sequence $x_i[n]$ of data points $x_i[n] = x(nT_i)$[13] with the new sampling interval $T_i = \frac{T_o}{L}$ from the original data sequence $x_o[n] = x(nT_o)$ with T_o the original sampling interval. The implementation consists of a *sample rate expander* which implements the zero-padding operation

$$x_e[n] = \sum_{k=-\infty}^{\infty} x[k]\delta_{n,kL} . \tag{1.80}$$

As before, the resampling process produces copies of the original signal at multiples of the new sampling frequency while at the same time the frequency is scaled by the factor L as can be seen by replacing the original sampling frequency ω_s with the new sampling frequency $\omega_s^i = L\omega_s$ in equation 1.7

$$F^i(\omega) = \frac{L\omega_s}{2\pi} \sum_{k=-\infty}^{\infty} F(L\omega - kL\omega_s)) \tag{1.81}$$

$$= \frac{\omega_c^i}{\pi} \sum_{k=-\infty}^{\infty} F(L\omega - k\omega_s^i) , \tag{1.82}$$

with $\omega_c^i = \omega_s^i/2$ the corresponding Nyquist frequency.

Therefore the expander is followed by an (ideal) low-pass filter[14] set to the original Nyquist frequency $\omega_c = \frac{\omega_s^i}{2L}$ to remove the additional copies from the frequency spectrum. In the time-domain the low-pass filter works as an interpolator, i.e. the zeros that were added in between the original data points will be modified such that the Nyquist condition is fulfilled. However, an ideal low-pass filter cannot be realized and therefore an ideal interpolation cannot be achieved. With the impulse response of the ideal low-pass 1.33 and 1.80 it can be seen that the interpolation condition is fulfilled

$$x_i[n] = \sum_{k=-\infty}^{\infty} x[k]\frac{\sin(\pi(n-kL)/L)}{\pi(n-kL)/L} \tag{1.83}$$

[13] $x(nT)$ is the value of the continuous function $x(t)$ at time $t = nT$, while $x[n]$ is the value of the discrete data set $x[k]$ at time $k = n$.

[14] The low-pass filter needs a gain of L.

Figure 1.13: Upsampled sequence of data from a HPGe detector. The original sampling rate was increased by a factor of four.

$$= \begin{cases} x[\frac{n}{L}] & \text{, for } n = 0, \pm L, \pm 2L, \ldots \\ \sum_{k=-\infty}^{\infty} x[k] \frac{\sin(\pi(n-kL)/L)}{\pi(n-kL)/L} & \text{, otherwise .} \end{cases} \quad (1.84)$$

Equation 1.83 is also called the *cardinal interpolation function* or *Whittaker reconstruction formula* and it is the most common form of recovering a function from a sequence a data points.

In order to understand the concept of upsampling, i.e. zero-padding and low-pass filtering, it is best to consider a modification of equation 1.80

$$x_e[n] = \left\{ x[n] \quad , n \geq kL < n+1 \, , \right. \quad (1.85)$$

i.e. instead of inserting zeros the original sample values are repeated until a new sample is available. This process is called zero-order or nearest neighbor interpolation. The nearest neighbor interpolation can also be expressed a combination of an zero-padding operation plus a moving average of gain L, i.e. $x[n] = \sum_{k=0}^{L} -1x[k]$. Similarly, any interpolation operation can be split into zero-padding plus subsequent filtering with the transfer function of the interpolation.

The ideal interpolation process can also be understood as zero-padding in the frequency domain. Here, the pass band of the transfer function is increased to match the new sampling frequency. The content of the transfer function for frequencies in between the original cut-off frequency ω_c^o and the new cut-of frequency $\omega_c^i = L\omega_c^o$ is set to zero, which is equivalent to the application of the ideal low-pass filter with the cut-off frequency of the original data set.

Linear Interpolation

Linear interpolation is the second simplest form of interpolation. It can be expressed as

$$y(x) = y[n] + x(y[n+1] - y[n]) \quad (1.86)$$

$$= y[n](1-x) + y[n+1]x \quad (1.87)$$

with $n < x < n+1$. The transfer function can be derived by applying the symmetrical form of the linear interpolation

$$y(x) = \frac{y\left[n-\frac{1}{2}\right] + y\left[n+\frac{1}{2}\right]}{2} + x\left(y\left[n+\frac{1}{2}\right] - y\left[n-\frac{1}{2}\right]\right) , \tag{1.88}$$

with $n - \frac{1}{2} < x < n + \frac{1}{2}$ onto a single frequency signal $f(\omega, n) = \exp(i\omega n)$:

$$\begin{aligned} H(\omega) &= \frac{\exp\left(i\omega\left(n-\frac{1}{2}\right)\right) - \exp\left(i\omega\left(n+\frac{1}{2}\right)\right)}{2\exp\left(i\omega n\right)} & (1.89)\\ &+ x\frac{\exp(i\omega(n-\frac{1}{2}) + \exp(i\omega(n+\frac{1}{2})}{\exp(i\omega(n)} & (1.90)\\ &= \frac{\exp\left(-\frac{i\omega}{2}\right) - \exp\left(\frac{i\omega}{2}\right)}{2} + x\left(\exp\left(-\frac{i\omega}{2}\right) + \exp\left(\frac{i\omega}{2}\right)\right) & (1.91)\\ &= \cos\left(\frac{\omega}{2}\right) + 2xi\sin\left(\frac{\omega}{2}\right) . & (1.92) \end{aligned}$$

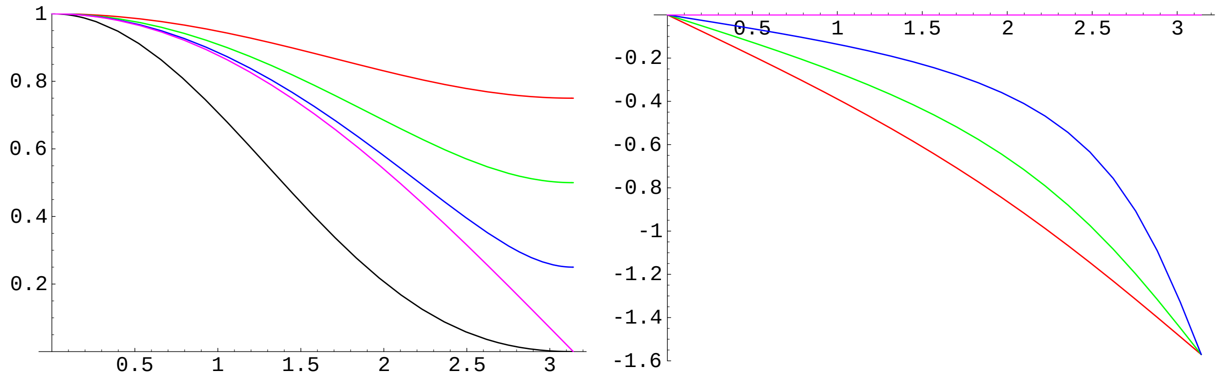
The magnitude and phase response of the linear interpolation are plotted for different x in figure 1.14. The imaginary part of the transfer function causes a phase shift. At $x = 0$ the transfer function is real (phase response is zero), but compared to the transfer function of the ideal low-pass the higher frequencies are suppressed leading to an incorrect interpolation results. Furthermore the transfer function is not zero above the cut-off frequency, possibly causing aliasing. However, if the data is oversampled, then the damping at higher frequencies is less severe and the linear interpolation would be more useful.

Linear interpolation can also be understood as a convolution of the input signal $x(n)$ with triangular pulse $L(t) = 1 - \frac{t}{T}$, which is itself a convolution of two rectangular functions (compare with the trapezoidal shaping in section 1.6.4). Therefore the overall transfer function for the linear interpolation is the quadratic sinc function $\left(\frac{sin(\omega)}{\omega}\right)^2$.

1.8.3 Quadratic Interpolation

The lagrangian equation for polynomial interpolation [20, 17, **?**] is used by the PSA DSP code and is therefore presented here. The lagrangian interpolation of the function value $y[x]$ at the position x

$$y[x] = \sum_{k=0}^{N} l_k(x)\, y[k] \tag{1.93}$$



(a) Magnitude response (b) Phase response

Figure 1.14: Magnitude and phase response of the linear interpolation. The overall transfer function is plotted in black together with the transfer function for four different x settings: $x = 0.375$ (red), $x = 0.25$ (green), $x = 0.125$ (blue) and $x = 0$ (purple).

with the coefficients defined by

$$l_k(x) = \frac{(x-0)\ldots(x-(k-1))(x-(k+1))\ldots(x-N])}{(k-0)\ldots(k-(k-1))(k-(k+1))\ldots(k-N)}\,, \tag{1.94}$$

which can also be written as

$$l_k(x) = \prod_{n=0,n\neq k}^{N} \frac{x-n}{k-n}\,. \tag{1.95}$$

This leads to the expression $l_k(j) = \delta_{kj}$ for the sample points and the coefficients $l_k(x)$ which are presented in table 1.1 for an interpolation of first (N=1), second order (N=2) and third order (N=3). In order to derive the transfer function of the quadratic interpolation, the interpolation will be expressed as an symmetric filter and therefore x has to be replace by $x+1$ in the coefficient table

$$y[x] = \frac{1}{2}x(x-1)\,y[n-1] - (x+1)(x-1)\,y[n] + \frac{1}{2}x(x+1)\,y[n+1]\,, \tag{1.96}$$

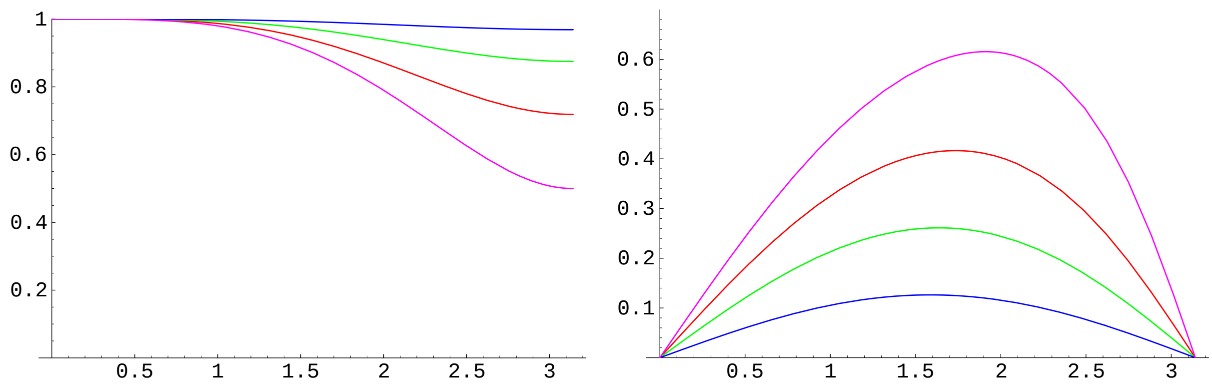
leading to the following transfer function

$$H(\omega) = (1-x^2) + x^2\cos(\omega) + x\,i\sin(\omega)\,, \tag{1.97}$$

which is plotted in figure 1.15 for different values of x.

For PSA an inverse interpolation is necessary, because the position x of the minimum has to be determined from the interpolating polynomial. This is accomplished with the additional requirement that

$$\frac{d}{dt}y[x] = \frac{d}{dt}\sum_{k=0}^{N} l_k(x)\,y[k] \equiv 0\,. \tag{1.98}$$



(a) Magnitude response (b) Phase response

Figure 1.15: Magnitude and phase response of the quadratic interpolation for four different values of x: $x = 0.5$ (purple), $x = 0.375$ (red), $x = 0.25$ (green) and $x = 0.125$ (blue).

Order	**n=0**	**n=1**	**n=2**	**n=3**
1	1-x	x	0	0
2	$\frac{1}{2}(x-1)(x-2)$	-x(x-2)	$\frac{1}{2}x(x-1)$	0
3	$-\frac{1}{6}(x-1)(x-2)(x-3)$	$\frac{1}{2}x(x-2)(x-3)$	$-\frac{1}{2}x(x-1)(x-3)$	$-\frac{1}{6}x(x-1)(x-2)$

Table 1.1: Coefficients for the Lagrange interpolation.

Appendix A

Useful Filter Equations

1. IIR (Recursive) Filter:

$$y[n] = \sum_{m=0}^{M} a_m x[n-m] + b_m y[n-m]$$

   (a) Single Pole High Pass Filter

$$\begin{aligned} a_0 &= \frac{1+x}{2} \\ a_1 &= -\frac{1+x}{2} \\ b_1 &= x \end{aligned}$$

   with $x = \exp(\frac{-1}{d}) = \exp(-2\pi f_c)$ and all other coefficients are zero such that $y[n] = a_0 x[n] + a_1 x[n-1] + b_1 y[n-1]$.

   (b) Single Pole Low Pass Filter

$$\begin{aligned} a_0 &= (1-x) \\ b_1 &= x \end{aligned}$$

   with $x = \exp(\frac{-1}{d}) = \exp(-2\pi f_c)$ and all other coefficients are zero such that $y[n] = a_0 x[n] + b_1 y[n-1]$.

   (c) 4 Stage Single Pole Low Pass Filter

$$\begin{aligned} a_0 &= (1-x)^4 \\ b_1 &= 4x \end{aligned}$$

$$\begin{aligned} b_2 &= -6x^2 \\ b_3 &= 4x \\ b_4 &= -x^4 \end{aligned}$$

with $x = \exp(\frac{-1}{d}) = \exp(-2\pi f_c)$ and all other coefficients are zero such that $y[n] = a_0 x[n] + b_1 y[n-1] + b_2 y[n-2] + b_3 y[n-3] + b_4 y[n-4]$.

(d) Bessel Filter with $f_c = \frac{1}{2} f_s$

- 1st Order Bessel Filter:

$$y[n] = x[n] + x[n-1] - y[n-1]$$

- 2nd Order Bessel Filter:

$$\begin{aligned} y[n] = x[n] &+ 2x[n-1] + x[n-2] \\ &- 2y[n-1] - y[n-2] \end{aligned}$$

- 3rd Order Bessel Filter:

$$\begin{aligned} y[n] = x[n] &+ 3x[n-1] + 3x[n-2] + x[n-3] \\ &- 3y[n-1] - 3y[n-2] - y[n-3] \end{aligned}$$

- 4th Order Bessel Filter:

$$\begin{aligned} y[n] = x[n] &+ 4x[n-1] + 6x[n-2] + 4x[n-3] + x[n-4] \\ &- 4y[n-1] - 6y[n-2] - 4y[n-3] - y[n-4] \end{aligned}$$

2. FIR Filter

(a) Moving Average

$$y[n] = \sum_{m=-M}^{M} \frac{1}{2M+1} x[n+m]$$

(b) Gaussian Filter

- Binomial Distribution with $P = 1/2$: 3 Points

$$y[n] = \frac{1}{4} x[n-1] + \frac{1}{2} x[n] + \frac{1}{4} x[n+1]$$

- Binomial Distribution with $P = 1/2$: 5 Points

$$y[n] = \frac{1}{16}x[n-2] + \frac{1}{4}x[n-1] + \frac{3}{8}x[n] + \frac{1}{4}x[n+1] + \frac{1}{16}x[n+2]$$

(c) Low Pass FIR Filter

$$\begin{aligned} c[n] &= (2f_c)\frac{\sin(2\pi f_c n)}{(2\pi f_c n)} \\ y[n] &= \sum_{m=-M}^{M} c[m]y[n-m] \end{aligned}$$

with $f_c = \frac{f_c}{f_s}$ the relative cut-off frequency.

(d) Least Squares Smoothing: 1st Order, 3 data points,

$$\begin{aligned} y[n-1] &= \frac{1}{6}(5y[n-1] + 2y[n] - y[n+1]) \\ y[n] &= \frac{1}{3}(y[n-1] + y[n] + y[n+1]) \\ y[n+1] &= \frac{1}{6}(-y[n-1] + 2y[n] + 5y[n+1]) \end{aligned}$$

(e) Least Squares Smoothing: 1st Order, 5 Data Points,

$$\begin{aligned} y[n-2] &= \frac{1}{5}(3y[n-2] + 2y[n-1] + y[n] + y[n+1]) \\ y[n-1] &= \frac{1}{10}(4y[n-2] + 3y[n-1] + 2y[n] + y[n+1]) \\ y[n] &= \frac{1}{5}(y[n-2] + y[n-1] + y[n] + y[n+1] + y[n+1]) \\ y[n+1] &= \frac{1}{10}(4y[n+2] + 3y[n+1] + 2y[n] + y[n-1]) \\ y[n+2] &= \frac{1}{5}(3y[n+2] + 2y[n+1] + y[n] + y[n-1]) \end{aligned}$$

(f) Least Squares Smoothing: 3rd Order, 5 Data Points,

$$y[n-2] = \frac{1}{70}(69y[n-2] + 4y[n-1] - 6y[n] + 4y[n+1] - y[n+2])$$

$$\begin{aligned}
y[n-1] &= \frac{1}{35}(2y[n-2]+27y[n-1]+12y[n]-8y[n+1]+2y[n+2]) \\
y[n] &= \frac{1}{35}(-3y[n-2]+12y[n-1]+17y[n]+12y[n+1]-3y[n+2]) \\
y[n+1] &= \frac{1}{35}(2y[n+2]+27y[n+1]+12y[n]-8[n-1]+2[n-2]) \\
y[n+2] &= \frac{1}{70}(69y[n+2]+4y[n+1]-6y[n]+4y[n-1]-y[n-2])
\end{aligned}$$

(g) Least Squares Smoothing: 3rd Order, 7 Data Points,

$$\begin{aligned}
y[n-3] = \frac{1}{42}(39 \quad & y[n-3] \quad +8y[n-2]-4y[n-1]-4y[n] \\
- \quad & 2y[n+3] \quad +4y[n+2]+y[n+1]) \\
y[n-2] = \frac{1}{42}(8 \quad & y[n-3] \quad +19y[n-2]+16y[n-1]+6y[n] \\
+4 \quad & y[n+3] \quad -7y[n+2]-4y[n+1]) \\
y[n-1] = \frac{1}{42}(-4 \quad & y[n-3] \quad +16y[n-2]+19y[n-1]+12y[n] \\
+ \quad & y[n+3] \quad -4y[n+2]+2y[n+1]) \\
y[n] = \frac{1}{21}(-2 \quad & y[n-3] \quad +3y[n-2]+6y[n-1]+7y[n] \\
-2 \quad & y[n+3] \quad +3y[n+2]+6y[n+1]) \\
y[n+1] = \frac{1}{42}(-4 \quad & y[n+3] \quad +16y[n+2]+19y[n+1]+12y[n] \\
+ \quad & y[n-3] \quad -4y[n-2]+2y[n-1]) \\
y[n+2] = \frac{1}{42}(8 \quad & y[n+3] \quad +19y[n+2]+16y[n+1]+6y[n] \\
+4 \quad & y[n-3] \quad -7y[n-2]-4y[n-1]) \\
y[n+3] = \frac{1}{42}(39 \quad & y[n+3] \quad +8y[n+2]-4y[n+1]-4y[n] \\
-2 \quad & y[n-3] \quad +4y[n-2]+y[n-1])
\end{aligned}$$

(h) Least Squares Smoothing: 5th Order, 7 Data Points,

$$\begin{aligned}
y[n-3] = \frac{1}{924}(923 \quad & y[n-3] \quad -6y[n-2]-15y[n-1]+20y[n] \\
- \quad & y[n+3] \quad +6y[n+2]-15y[n+1])
\end{aligned}$$

$$
\begin{aligned}
y[n-2] = \frac{1}{154}(& \; y[n-3] & +148y[n-2] + 15y[n-1] - 20y[n] \\
+ & \; y[n+3] & -6y[n+2] + 15y[n+1]) \\
y[n-1] = \frac{1}{308}(-5 & \; y[n-3] & +30y[n-2] + 233y[n-1] + 100y[n] \\
-5 & \; y[n+3] & +30y[n+2] - 75y[n+1]) \\
y[n] = \frac{1}{231}(5 & \; y[n-3] & -30y[n-2] + 75y[n-1] + 131y[n] \\
+5 & \; y[n+3] & -30y[n+2] + 75y[n+1]) \\
y[n+1] = \frac{1}{308}(-5 & \; y[n+3] & +30y[n+2] + 233y[n+1] + 100y[n] \\
-5 & \; y[n-3] & +30y[n-2] - 75y[n-1]) \\
y[n+2] = \frac{1}{154}(& \; y[n+3] & +148y[n+2] + 15y[n+1] - 20y[n] \\
+ & \; y[n-3] & -6y[n-2] + 15y[n-1]) \\
y[n+3] = \frac{1}{924}(923 & \; y[n+3] & -6y[n+2] - 15y[n+1] + 20y[n] \\
- & \; y[n-3] & +6y[n-2] - 15y[n-1])
\end{aligned}
$$

3. Interpolation with FIR Filters

   (a) Linear Interplation

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

   with $n < x < n + 1$.

4. Equations for Numerical Differentiation

   (a) 2 Points

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

   (b) 3 Points

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

(c) 5 Points

$$y[n] = \frac{1}{12}x[n+2]) + \frac{8}{12}x[n+1] - \frac{8}{12}x[n-1] - \frac{1}{12}x[n-2]$$

(d) 7 Points

$$y[n] = \frac{1}{60}x[n{+}3]{+}\frac{9}{60}x[n{+}2]){+}\frac{45}{60}x[n{+}1]{-}\frac{45}{60}x[n{-}1]{-}\frac{9}{60}x[n{-}2]{-}\frac{1}{60}x[n{-}3]$$

(e) 9 Points

$$\begin{aligned} y[n] &= \frac{3}{810}x[n+4] + \frac{32}{810}x[n+3] + \frac{168}{810}x[n+2]) + \frac{672}{810}x[n+1] \\ &\quad -\frac{672}{810}x[n-1] - \frac{168}{810}x[n-2] - \frac{32}{810}x[n-3] - \frac{3}{810}x[n-4] \end{aligned}$$

(f) Spline, 2nd Order, 5 Points

$$y[n] = \frac{1}{12}x[n+2]) + \frac{8}{12}x[n+1] - \frac{8}{12}x[n-1] + \frac{1}{12}x[n-2]$$

(g) Spline, 4th Order, 7 Points

$$y[n] = \frac{3}{168}x[n{+}3]{-}\frac{26}{168}x[n{+}2]){+}\frac{127}{168}x[n{-}1]{-}\frac{127}{168}x[n{-}1]{+}\frac{26}{168}x[n{-}2]{-}\frac{3}{168}x[$$

(h) 7 Point Filter

$$y[n] = \frac{1}{3}x[n{+}3]{+}\frac{1}{2}x[n{+}2]){+}x[n{+}1]{-}x[n{-}1]{+}\frac{1}{2}x[n{-}2]{-}\frac{1}{3}x[n{-}3]$$

(i) 11 Point Filter

$$\begin{aligned} y[n] &= \frac{1}{5}x[n+5] - \frac{1}{4}x[n+4] + \frac{1}{3}x[n+3] \\ &\quad -\frac{1}{2}x[n+2]) + x[n+1] - x[n-1] + \frac{1}{2}x[n-2] \\ &\quad -\frac{1}{3}x[n-3] + \frac{1}{4}x[n-4] - \frac{1}{5}x[n-5] \end{aligned}$$

5. Methods for Finding Extreme Values

(a) Bevington Method

$$e[n] = \frac{x[n+1] - x[n-1]}{x[n+1] - 2x[n] + x[n-1]}$$

(b) Using Quadratic Interpolation

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

Bibliography

[1] A.V. Oppenheim and R.W. Schafer, *Zeitdiskrete Signalverarbeitung*, R.Olderburg Verlag, 1999

[2] B. Jaehne, *Digitale Bildverarbeitung*, Springer-Verlag, 1997

[3] S.D. Stearns, *Digital Signal Analysis*, Prentice Hall, 1990

[4] Richard G. Lyons, *Understanding Digital Signal Processing*, Addison-Wesley Longman, Inc., Reading, MA, 1997

[5] R.W. Hamming, *Numerical methods for scientist and engineers*, Dover Publications, 1987

[6] F.B. Hildebrand, *Introduction to numerical analysis*, Dover Publications, 1987

[7] Steven W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, California Technical Publishing, `http://www.dspguide.com`

[8] `http://mathworld.wolfram.com/Convolution.html`

[9] Analog Devices Application Note 410, `http://www.analog.com`

[10] `http://mathworld.wolfram.com/ShahFuncion.html`

[11] `http://mathworld.wolfram.com/DeltaFunction.html`

[12] V. Radeka, *Trapezoidal Filtering of Signals from Large Germanium Detectors at High Rates*, IEEE Trans. Nuc. Sci. NS-19, No. 1, 412 (1972)

[13] F.S.Gouldig, *Pulse-Shaping in Low-Noise Nuclear Amplifiers: A Physical Approach to Noise Analysis*, Nucl. Instr. Meth. 100 (1972), p.493-504

[14] F.S Goulding and D.A.Landis, *Signal Processing for Semiconductor Detectors*, IEEE Trans. Nucl. Sci. NS-29 (1982), p.1125-1141

[15] F.S.Goulding NIM A 485 (2002) 653-660

[16] M.Eschenauer et a., NIM 340 (1994), 364

[17] Julius Orion Steven III Home Page `http://www-ccrma.stanford.edu/~jos/`

[18] M.Unser, Biomedical Imaging Group Homepage, `http://bigwww.epfl.ch/index.html`

[19] P. Thvenaz, T. Blu, M. Unser, *Image Interpolation and Resampling*, Handbook of Medical Imaging, Processing and Analysis, I.N. Bankman, Ed., Academic Press, San Diego CA, USA, pp. 393-420, 2000, `http://bigwww.epfl.ch/publications/thevenaz9901.pdf`

[20] P. Deuflhard, Andreas Hohmann, *Numerische Mathematik Bd.1, Eine algorithmisch orientierte Einf"uhrung*, de Gruiter

[21] Lagrange Interpolating Polynomial, `http://mathworld.wolfram.com/LagrangeInterpolatingPolynomial.html`

[22] V.Radeka, *Low-Noise Techniques in Detectors*, Ann. Rev. of Nucl. Part. Sci. 29, 1, 1982, p.764-768

[23] H.Spieler, Radiation Detectors and Signal Processing (presented at University of Heidelberg, Oct. 8-12, 2001)