Signal sampling techniques for data acquisition in process control
Laan, Marten Derk van der

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
1995

Link to publication in University of Groningen/UMCG research database

Citation for published version (APA):

Copyright
Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

Take-down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): http://www.rug.nl/research/portal. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.
Chapter 7

On the implementation of sampling schemes

In the previous chapters we have stepwise developed the general sampling scheme, and have selected some filters that can be used within this scheme. So far the results were purely theoretical or obtained by computer-simulations. However, alternative sampling operations only make sense if it is possible to implement the schemes, yielding alternative devices for A/D- and D/A-conversion.

In Section 7.1 we discuss possibilities for the implementation of the general sampling scheme. We pay particular attention to the $B_0$-spline pre-filter, which results in the local integral sampling method, whose implementation is discussed in Section 7.2. Then we study the characteristics of this method by means of measurements with a hardware prototype. Section 7.3 addresses the design and implementation of this prototype and the measurements taken with the device. In Section 7.4 we discuss an application where sampled functions are not reconstructed, but further analyzed instead, namely the evaluation of experiments in Gas-Liquid-Chromatography. The general sampling scheme with spline filters offers a comprehensive representation of the data, which considerably facilitates the analysis. Finally, some concluding remarks are given in Section 7.5.

7.1 Implementation of sampling schemes

In order to compete with traditional techniques, alternative A/D- and D/A-converters based on the general sampling scheme must be implemented as low-cost integrated circuits. Since A/D- and D/A-converters are hybrid systems, they have continuous-time parts, which require implementation in analog hardware. This is a bit troublesome, since most of the filters we have discussed cannot be realized in analog circuitry.

A makeshift solution would be to perform conversion in two stages: first use a traditional A/D-converter to convert the continuous-time signal, and then process the resulting discrete-time signal using the general sub-sampling scheme of Figure 5.1. If a continuous-time signal is to be retrieved from the samples, the conversion consists of up-sampling and digital post-filtering,
followed by D/A-conversion with a traditional converter. This approach will be discussed in more detail in Sections 7.1.1 and 7.1.2.

In some cases, an A/D converter can be implemented using analog techniques, which will be discussed in Section 7.1.3. For D/A-conversion, similar techniques can be used. For example, in [WTKM89] a D/A-converter is realized in analog circuitry which carries out second-order cardinal spline interpolation. In terms of the general sampling scheme, this is equivalent to δ-modulation and post-filtering with the C-spline, combined in a single operation. We will comment on this in Section 7.1.3.

The optimal solution for an alternative A/D-converter would be to integrate an analog pre-filter and the ideal sampler in a single module. In that case there would be no analog signal line between filter and sampler, thus reducing the sensitivity to external disturbances. For A/D-converters based on B-spline filters of low order such an implementation is possible, using analog integration techniques. Especially for the zero-order B-spline this leads to a robust implementation to be discussed in detail in Section 7.2.

7.1.1 Implementation with pre-sampling

Implementation of an A/D-converter based on the general sub-sampling scheme requires that the signal is sampled first, to obtain a discrete-time approximation of the signal. The latter signal can be filtered and down-sampled using standard digital techniques, which finally yields the ‘samples’ of the continuous-time signal. The reconstruction process consists of up-sampling, digital filtering and D/A-conversion. The overall scheme is depicted in Figure 7.1.

If the filters are properly selected, the discrete-time signal $z_k$ is an optimal approximation of the signal $x_k$. What does this mean for the error $x(t) - z(t)$? This problem was addressed in Section 5.2. We must choose a multiresolution level $j$, such that we can assume

$$x \in V_j, \quad x(t) = \sum_k x_k \phi_j(t - k)$$

to hold. Then the error $|x(t) - z(t)|$ is bounded by $\|x_k - z_k\|_2^2$.

The drawback of this method is that the costs are higher than those for standard A/D-conversion realized by an analog pre-filter and a standard A/D-converter (ADC). The scheme of Figure 7.1 has an ADC which operates on an $M$ times higher rate compared to standard conversion, requiring a more expensive device. On the other hand, an anti-alias filter prior to the ADC can be
omitted, since the aliasing effects will be removed by the digital pre-filter. Moreover, our general scheme may offer a better representation of the signal, which implies that the final sampling rate can be lower in comparison to the original scheme, while preserving the same approximation quality. This will significantly reduce the costs of further processing, storage and transport of the sampled data. Whether this compensates for the additional costs of the converters will depend on the application.

The situation for D/A-conversion is similar. The required D/A-converter (DAC) operates at a high rate, which involves extra costs. The scheme, however, does not contain analog filters. Instead, the post-filtering is carried out in the discrete-time domain prior to D/A-conversion.

### 7.1.2 Implementation of the general sub-sampling scheme

The general sub-sampling scheme consists of digital filters and up/down-samplers. All components can be easily realized in software, and by digital hardware components as well. Up/down-sampling is equivalent to digital (de-)multiplexing for which ready-to-use modules exist. The filtering can be implemented in different ways as we have seen in Chapter 5; viz., by

- FIR filtering,
- IIR filtering (spline filters), or with the
  - Pyramid algorithm.

FIR filtering is a straightforward method. If the impulse responses have infinite length, they must be truncated to an appropriate length. After this length has been determined, the filter coefficients are fixed and the processing is a pipeline process, suitable for implementation by DSP chips or systolic arrays.

In [UAE91] it is shown that the spline filters can be realized by IIR filters. This allows filtering without truncation of the impulse response. An IIR filter is obtained by splitting a given spline filter into its causal and anti-causal part. To process the anti-causal part, the signal is time-reversed and then filtered. Hence, the full signal must be available before conversion can take place. Therefore, this method is not suitable for real-time processing. The required hardware or software components are the same as in the case of FIR filtering, plus an additional buffer to store the full signal.

The last alternative is the Pyramid algorithm (also called Mallat’s algorithm), which we discussed in Section 5.2. Here filtering is carried out recursively, thereby reducing the number of filter coefficients that have to be stored. The Pyramid algorithm is more difficult to realize in hardware, due to its data dependencies between different octaves. Nevertheless, a number of VLSI architectures have been proposed (e.g., [PN93]). In [FM94], a special non-linear transform was applied to regularize the algorithm, allowing implementation by systolic arrays.

It has to be noticed that any digital implementation will introduce quantization errors due to finite-length representation of real numbers. This will affect both data and filter coefficients. Quantization of data will introduce amplitude errors, as described in Chapter 1. Quantization of filter coefficients leads to different filters, which implies that the filter combinations will
loose their $l^2$-optimality. It is, therefore, advantageous to have filters with coefficients that can be exactly represented by finite-length numbers. The discrete-time B-spline filters have this property, because they can be defined as convolutions of simple filters with rational coefficients.

### 7.1.3 Analog implementations

**A/D-conversion**

In the case of B-spline pre-filters, the A/D-converter can be realized as an analog device. For the zero-order B-spline ($B_0$-spline) this is straightforward. The (causal) impulse response of this filter, for sampling interval $\Delta$, is given by

$$h(t) = \begin{cases} 1 & 0 \leq t < \Delta \\ 0 & \text{elsewhere} \end{cases}$$

The output $y$ of the filter equals the integral of the input $x$:

$$y(t) = (x * h)(t) = \int_{t-\Delta}^{t} x(\tau) \, d\tau$$

Hence, the samples, taken at $k\Delta$, are just local integrals of the signal

$$y_k = \int_{(k-1)\Delta}^{k\Delta} x(\tau) \, d\tau.$$  

For higher order splines, the integration becomes a weighted integration which involves more complex integration hardware. Implementation of first and second order splines is possible, but for higher orders one reaches the accuracy limits of analog integration techniques.

We will come back in detail to this integrating A/D-conversion in Section 7.2.

**D/A-conversion**

An analog D/A-converter using second-order cardinal spline interpolation was realized in [WTKM89]. Since the cardinal spline consists of piecewise quadratic polynomials, convolution with this function can be rewritten as multiplication with periodic polynomial components. These components were generated using a saw-tooth generator and analog multipliers. The authors claim to achieve a good reconstruction quality. The circuit is implemented using discrete components, but VLSI implementation is also possible.

We have mentioned that cardinal spline interpolation is equivalent to the reconstruction in the general sampling scheme, using a second-order C-spline filter (see Table 4.2 on p. 79). Rather than implementing a C-spline filter, the approach can be simplified. According to Table 4.1 on p. 66, the C-spline $\phi_I$ is derived from the second-order B-spline $\beta^2$ by

$$\phi_I(t) = ((b)^{-1} * \beta^2)(t) = \sum_k (b^{-1})_k \beta^2(t-k)$$

where $b_k = \beta^2(k)$ and $(b)^{-1}$ is the convolution inverse of $b$. Due to the associativity of the convolution operator, filtering with a C-spline filter can be split into digital filtering with the
sequence \( b_k \) followed by mixed convolution with a simple B-spline filter\(^1\). The latter operation can be carried out by techniques similar to those described in [WTKM89], but with major simplifications.

### 7.2 A/D-conversion based on local integrals

The local integration, discussed in Section 7.1.3, can be implemented very efficiently using Voltage-to-Frequency-Converters (VFC’s). A VFC transfers an input signal into a pulse train with a linear relationship between input voltage and output frequency. Integration then reduces to pulse counting. The accuracy of integration depends on the number of pulses within an interval and, therefore, on the dynamic range of the VFC. This range should be high enough to retain the information content of the original signal in the outgoing pulse train.

It has to be noted that commercially available VFC’s are an order of magnitude slower than state-of-the-art A/D-converters. However, for measurement applications in technical processes the speed of these VFC’s is sufficient.

The operation principle of VFC’s is to charge an integrating capacitor with the applied input voltage until the voltage at the capacitor has reached a given level. Then, an output pulse is generated and the capacitor is discharged, to be loaded again by the input voltage. Only if the discharge time is negligibly small, a linear relationship between input voltage and output frequency is obtained. However, in practice the discharge time cannot be neglected, which introduces a non-linearity error.

A solution to this problem is based on a polarity-alternating concept, in which the capacitor is charged \( \text{and} \) discharged with the input voltage, see Figure 7.2. This concept leads to increased linearity. An architecture for VFC’s, based on the polarity-alternating concept was proposed in [Hal87], and has been realized in the CS-EL lab\(^2\) using discrete hardware components. For details, we refer to [LHK93, Kob91]. Patent application is pending [HHLS95].

The values of integrals are obtained by counting pulses. For this a separate module is required, which counts the pulses during every integration interval. At the end of each interval, the counter contains the final value of a sample, which must be stored or transferred to the analyzing computer. Before starting a new cycle the counter is reset. In [Hal84], a count/storage module based on this principle was presented. A prototype, which is able to process multiple input channels simultaneously, has been realized as an application specific integrated circuit (ASIC) [KM90].

Combination of VFC and count/storage module leads to a robust A/D-converter implementation. If converted data are directly processed by a general-purpose computer, a count/storage module may be integrated with this computer, as shown in Figure 7.3. This has two advantages:

- noiseless digital transport over medium distance, and
- free-running VFC’s, i.e., no control signals are required.

---

\(^1\)This approach is clearly described in [AUE92]

\(^2\)CS-EL lab is the electronics laboratory of the Department of Computing Science at the University of Groningen.
FIGURE 7.2: **Upper:** The operation of a standard VFC. A capacitor is charged with the input voltage (assuming a constant input). If a certain limit is reached, a pulse is generated, and the capacitor is discharged. The non-zero discharge time introduces a non-linearity in the relation between input voltage and frequency of the pulse train generated. **Lower:** The polarity-alternating VFC. The capacitor is charged and discharged using the input voltage. Switching logic is required to change the polarity of the charging current.

FIGURE 7.3: **Block diagram of an integrating A/D-converter.** For description see text.
Alternatively, VFC, count/storage module and an interface unit can be combined in a separate device, which can communicate with computers via standard protocols. This approach may be useful when monitoring data from spatially distributed processes.

### 7.3 Experiments with local integral sampling

We have mentioned that practical A/D-converters are not able to take point measurements due to internal capacities. Instead, signals are integrated during (short) intervals. One can explicitly carry out these integrations. This leads to the sampling scheme with the $B_0$-spline pre-filter, discussed in Section 7.1.3. It will be referred to as local integral sampling. We have seen that VFC’s can be used to implement such a scheme. In this section, we further investigate the properties of local integral sampling by means of measurements on a hardware prototype. The design and implementation of this device is discussed in Section 7.3.1, and the measurements are described in Section 7.3.2. Conclusions about this investigation are given in Section 7.3.3.

From the results of Chapter 5, we know that the sampling scheme with the $l^2$-optimal combination $sp0-sp0$ does not, in general, yield a very good approximation, except when the input signal is piecewise constant. However, the $B_0$-spline pre-filter provides satisfactory results in many cases, e.g., for the mixed signals (see Figure 5.12 on p. 105). We also concluded in Chapter 6, that the $B_0$-spline pre-filter is an interesting alternative anti-alias filter in digital control systems.

In practice, there is another issue. In many sampling applications there is no pre-filter at all. Instead, the sampling rate is chosen higher than strictly necessary to reduce aliasing. If samples are taken in a noisy environment, the noise is usually filtered out using a digital filter, which requires some redundancy to be present in the digital representation. In such a case, an integrating A/D-converter, with its implicit $B_0$-spline pre-filtering, can improve the quality of the acquired data. Compared to standard point measurements (or better: their approximations), the $B_0$-spline pre-filter has some inherent advantages:

- **noise suppression:**
  the $B_0$-spline filter has a low-pass character (see Figure 6.8 on p. 121), attenuating high-frequency noise;

- **suppression of power-frequency disturbances:**
  since the Fourier-transform of the $B_0$-spline is a sinc-function, the sensitivity to power-frequency noise can be reduced by choosing the integration time $\Delta$ in such a way that the zeros of the sinc-function are exactly located at multiples of the power frequency.

Integrating techniques are commonly utilized in measuring devices like digital voltage meters, and integrating A/D-converters exist as well. To investigate the local integral sampling method in more detail, we decided to set up an experiment to compare traditional sampling (through point measurements) to the integral sampling method. In order to guarantee that all other conditions stay the same, it would be best if both sampling methods were implemented in a single device. This was accomplished in a new design, which offers a varying integration duration, variable
between the theoretical limits 0 (the ideal point measurements) and \( \Delta \) (the general sampling scheme with \( B_0 \) spline pre-filtering).

A prototype has been built to study the effects of a varying integration period on the noise suppression. In the following sections, we discuss design and implementation of a hardware prototype of an integrating A/D-converter, and the results of some measurements carried out with this device. The device has been realized in our CS-EL lab.

### 7.3.1 An integrating A/D-converter

This section discusses design and implementation of a prototype of an integrating A/D-converter. The major motivations for this work are:

- to have a prototype for carrying out the experiments mentioned in the previous section;
- to gain experience with analog integration techniques, and to assess their accuracy;
- to obtain a device for the acquisition of gaschromatographic data.

For this last reason, the design specifications were derived from the characteristics of the analog output-signal of a gaschromatograph.

To enable a good comparison between point measurements and local integrals, it must be possible to vary the integration interval independently of the sampling interval \( \Delta \). Therefore, the device consists of two separate units: an analog integrator and a (standard) A/D-converter. At the beginning of each sampling interval, the integrator starts integrating the input signal. At a variable point within \( \Delta \), the output signal of the integrator is sampled and converted to a digital value. The duration of the integration is denoted by \( I_\Delta \) and is expressed in a percentage of \( \Delta \), varying from 0 (point measurements) to 100 %. Of course the value of 100 % cannot be accomplished in a practical device, due to the time needed to discharge the integrating capacitor. The timing of these events is schematically presented in Figure 7.4.

To keep the unit simple, it is not required to store the sampled data. As soon as a sample is obtained, it should be transported to a host computer via a standard interface. It is assumed that the host responds fast enough to the samples provided. Consequently, the maximum measuring time will only be bounded by the storage capacity of the host.
From the characteristics of gas chromatographic signals, we derive requirements for the basic parameters of the unit, summarized in Table 7.1.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Δ</td>
<td>sampling interval</td>
<td>20 . . . 100 ms</td>
</tr>
<tr>
<td>$U_{in}$</td>
<td>input voltage</td>
<td>$-10 \text{ mV} \ldots 1 \text{ V}$</td>
</tr>
<tr>
<td>$Z_{in}$</td>
<td>input impedance</td>
<td>$\geq 2 \text{ k}\Omega$</td>
</tr>
</tbody>
</table>

The prototype was realized using a Maxim-190 microcontroller board. This board is specially designed for data acquisition and is equipped with a Max190 (12 bits) A/D-converter, an Intel 80C32 microprocessor, an RS-232 serial interface, memory and power unit. In addition, an analog integrator was implemented with standard operational amplifiers. Due to the specific timing required, it was not possible to use the 80C32 processor for this purpose. Instead, a separate module controls the process. The overall structure of the device is given in Figure 7.5.

![Figure 7.5: The structure of the integrating A/D-converter.](image)

A point of attention is the integrator’s integration constant ($C_{int}$), i.e., the slope of the integrator’s output when its input is kept constant at 1 V. In order to exploit the full dynamic range of the internal A/D-converter, this integration constant should be adapted to the maximum amplitude of the input signal and to the moment of measuring (i.e., $I_{\Delta}$). However, the actual sample must be taken before the integrator output becomes larger than the maximum input of the ADC. This problem is depicted in Figure 7.6. A simple solution is to make this integration constant controllable by the user. If saturation occurs, this is clearly visible in the sampled data.

A variable integration constant in combination with a variable integration interval introduces another problem. In order to interpret the digital output, at least one of the two parameters must be measured during operation. Since this is not a practical situation, we decided to fix the interval $I_{\Delta}$ to a number of values, which have been chosen as 25, 50, 75, and 100 % of the sampling interval $\Delta$. A side-effect of this choice is a further design simplification.

Assuming maximum input of 1 V, a fast rate $\Delta = 20 \text{ ms}$ and $I_{\Delta} = 25 \%$, the integrator must produce its maximum output after 5 ms. For $\Delta = 100 \text{ ms}$ and $I_{\Delta} = 100 \%$, the maximum must be reached after 100 ms. The maximum output of the integrator equals the maximum input of the internal converter being 4.095 V. Together these requirements lead to $C_{int}$ to be variable between 41 and 800 V/s.
On the implementation of sampling schemes

Correct

Wrong

Figure 7.6: Saturation of the integrator output before the measurement has been taken leads to an erroneous value. The slope of the integrator output is determined by the input voltage and the integration constant $C_{\text{int}}$. **Left:** Given an integration interval $I_{\Delta}$ and assuming maximum input voltage, the integration constant must be such that the integrator output reaches the maximum input of the ADC just after the time of the measurement. Then, the digital output is almost maximal, utilizing the full range of the ADC. **Right:** If the integration interval is enlarged to $I'_{\Delta}$, and the integration constant is unchanged, the measured value is incorrect.

The resulting device has an analog input, an RS-232 compatible serial output, and three user-controllable parameters: sampling interval $\Delta$, integration interval $I_{\Delta}$, and integration constant $C_{\text{int}}$. In addition, special software on the host computer was developed to represent and store the sampled data. For the full details of the integrating A/D-converter’s design and implementation, we refer to [Gro94].

### 7.3.2 Experiments

In this section we discuss the tests of the integrating A/D-converter and the measurements that have been carried out. Besides the question whether the device meets its design specifications, a basic topic is the investigation of its linearity and the sensitivity of its analog part to external disturbances. Our a priori assumption is that this sensitivity will be reduced if the integration interval $I_{\Delta}$ is enlarged.

To test the device, we used DC inputs and harmonic inputs generated by a function generator. For some tests, white noise was superimposed to the input signal. The internal timing was monitored using an oscilloscope. The digital output was stored and afterwards processed by MATLAB.

First, the timing parameters were determined. Both sampling interval $\Delta$ and integration constant $C_{\text{int}}$ meet our requirements. Lengths of the integration intervals $I_{\Delta}$ were obtained by linear regression of measurements taken at 5 different sampling intervals. Their values are listed in Table 7.2. Of course, they differ from the ideal values (25–50–75–100) due to the time required for discharging. Important are their mutual ratios. These are good: $88.2 \approx 2 \cdot 44.0$, etc.

<table>
<thead>
<tr>
<th>Specified (in % of $\Delta$)</th>
<th>25</th>
<th>50</th>
<th>75</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measured (in % of $\Delta$)</td>
<td>21.5</td>
<td>44.0</td>
<td>66.5</td>
<td>88.2</td>
</tr>
</tbody>
</table>
To study the characteristics of the device we carried out the following measurements, for all 4 fixed values of the integration interval $I_\Delta$:

1. average digital output for varying DC input, fixed $C_{\text{int}}$, fixed sampling rate,
2. average digital output for varying $C_{\text{int}}$, fixed DC input, fixed sampling rate,
3. variation in digital output, fixed DC input,
4. FFT of digital output, harmonic input, different frequencies, different sampling rates, and
5. output variance for DC input with noise.

Some of the results of the DC input voltage measurements are presented in Figure 7.7 (varying input and fixed $C_{\text{int}}$), and in Figure 7.8 (fixed input and varying $C_{\text{int}}$). Especially the curves of Figure 7.7 show an excellent linearity.

Figure 7.9 shows a recording of the digital output for a fixed DC input voltage of 0.2 V. We may assume that the digital output varies due to internal noise. The standard deviation of the digital output for this sequence is 1.368, from which we can conclude that the A/D-conversion has at most 10 effective bits.

To test harmonic distortion, the device was fed with harmonic signals. The output data were recorded and the spectral power density was estimated using Welch’s averaged periodogram
FIGURE 7.8: Linearity of the integrating A/D converter. Varying integration constant, sampling interval and input voltage fixed. The different series are for different values of $I_\Delta$. Each of the solid lines represents the least squares fit of the corresponding measurements.

FIGURE 7.9: Digital output sequence for fixed DC input. See text.
method [Wel67]. The result for harmonic input of 10 Hz sampled with $\Delta = 18.3$ ms is given in Figure 7.10 for the different values of $I_\Delta$. The spectrum clearly shows the 10 Hz component and a DC component. The first harmonic (20 Hz) reaches $-40$ dB for $I_\Delta = 25 \%$, and slightly less for the other values of $I_\Delta$. Repeating the experiment with harmonics of 1, 2, and 4 Hz, sampled with $\Delta = 100$ ms, gave similar results. The attenuation of the second harmonic was slightly better, around $-50$ dB.

Finally, we tested the device for DC input plus white noise. The integration constant $C_{\text{int}}$ was kept constant. Table 7.3 gives the standard deviations in the digital output for $\Delta = 60$ ms. We see that the noise level decreases with increasing integration interval which is in agreement with our expectations.

<table>
<thead>
<tr>
<th>$I_\Delta$ [%]</th>
<th>21.5</th>
<th>44.0</th>
<th>66.5</th>
<th>88.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>standard deviation</td>
<td>169.5</td>
<td>80.3</td>
<td>56.6</td>
<td>44.1</td>
</tr>
</tbody>
</table>

To investigate this further, we repeated the experiments with a harmonic ‘noise’ signal with different frequencies above the Nyquist frequency. The frequency response of the $B_0$-spline filter, depicted in Figure 6.8 on p. 121, was given by

$$|H(f)| = |\text{sinc}(f\Delta)|.$$  

The first point of zero response is at a frequency of $1/\Delta$. In our case the actual duration of the integration length is shorter, and depends on the value of $I_\Delta$. For $I_\Delta = 88.2 \%$ and $\Delta = 60$ ms
TABLE 7.4: Attenuation for DC input + harmonic ‘noise’, $\Delta = 60$ ms. For every value of $I_\Delta$, the table has two entries: theoretical and measured attenuations.

<table>
<thead>
<tr>
<th>Frequency $[\text{Hz}]$</th>
<th>$I_\Delta = 21.5%$ ($f_z = 79.4$ Hz)</th>
<th>$I_\Delta = 44.0%$ ($f_z = 37.9$ Hz)</th>
<th>$I_\Delta = 66.5%$ ($f_z = 25.1$ Hz)</th>
<th>$I_\Delta = 88.2%$ ($f_z = 18.9$ Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>theor.</td>
<td>meas.</td>
<td>theor.</td>
<td>meas.</td>
</tr>
<tr>
<td>11</td>
<td>-0.3</td>
<td>0.1</td>
<td>-1.2</td>
<td>-0.7</td>
</tr>
<tr>
<td>12</td>
<td>-0.3</td>
<td>0.2</td>
<td>-1.5</td>
<td>-0.8</td>
</tr>
<tr>
<td>13</td>
<td>-0.4</td>
<td>0.2</td>
<td>-1.8</td>
<td>-1.1</td>
</tr>
<tr>
<td>14</td>
<td>-0.4</td>
<td>0.2</td>
<td>-2.0</td>
<td>-1.3</td>
</tr>
<tr>
<td>15</td>
<td>-0.5</td>
<td>0.3</td>
<td>-2.4</td>
<td>-1.5</td>
</tr>
<tr>
<td>16</td>
<td>-0.6</td>
<td>0.0</td>
<td>-2.7</td>
<td>-2.0</td>
</tr>
<tr>
<td>17</td>
<td>-0.7</td>
<td>0.2</td>
<td>-3.1</td>
<td>-2.2</td>
</tr>
<tr>
<td>18</td>
<td>-0.7</td>
<td>0.3</td>
<td>-3.5</td>
<td>-2.2</td>
</tr>
<tr>
<td>19</td>
<td>-0.8</td>
<td>0.3</td>
<td>-3.9</td>
<td>-2.7</td>
</tr>
<tr>
<td>20</td>
<td>-0.9</td>
<td>0.2</td>
<td>-4.4</td>
<td>-2.9</td>
</tr>
</tbody>
</table>

This frequency, which we denote as $f_z$, is at $18.9$ Hz. For the other (lower) values of $I_\Delta$, the first zero moves to higher frequencies. For example, if $I_\Delta = 66.5\%$ then it holds $f_z = 25.1$ Hz.

We measured the variance of the digital output for harmonic input with frequencies exceeding the Nyquist frequency of $1/(2\Delta) \approx 8.4$ Hz. These frequencies give rise to aliasing and should, therefore, be attenuated as much as possible. Table 7.4 contains both theoretical and measured attenuation, expressed in dB. The theoretical values are obtained from

$$20 \cdot 10 \log \left| \text{sinc} \frac{f}{f_z} \right|.$$

The measured data are from a single recording of approximately 80 samples. The dB-values were obtained using the standard deviation of the output to harmonic input of 1 Hz as a reference, which gives a systematic error of approximately 0.04 dB.

Due to the limited number of data, the accuracy is not very high. Nevertheless the measured attenuation values match the theoretical filter characteristics quite well. The table shows another characteristic phenomenon: For $I_\Delta = 88.2\%$ and harmonic input of 19 Hz, the frequency of the input is close to $f_z$ of the filter. Indeed this frequency is strongly attenuated.

7.3.3 Conclusions

We summarize the conclusions regarding the implementation of the integrating A/D-converter and the measurements taken by this device.

- The implementation of the integrator in discrete analog components has led to a device with good linearity and reasonable harmonic distortion.

Note that we use the actual (measured) duration of the integration interval.
The analog integrator introduces some noise, which reduces the effective resolution to at most 10 bits.

According to theory, integration must reduce the sensitivity to external disturbances outside the pass-band of the implicit B-spline filter. This is confirmed by the measurements. In particular, a harmonic disturbance is strongly attenuated if its frequency is close to one of the zeros in the Bode diagram of the filter. This feature of $B_0$-spline filters can be utilized to suppress harmonic disturbances, e.g., from the power net.

### 7.4 An example of sampled data processing: Gas-liquid-chromatography

In many measuring applications, reconstruction of a time-continuous signal from samples is not the primary goal. Instead, sampled data are further processed and/or analyzed. In this case, one needs an optimal representation, facilitating further analysis. We will see that the theory for reconstruction in the general sampling scheme can help us finding such a representation.

#### 7.4.1 Gas-liquid-chromatography

A particular application of advanced sampled data processing is found in Gas-Liquid-Chromatography (GLC). GLC is an important and frequently used analytical technique for the decomposition of gaseous and liquid compound mixtures into their components, and for the detection of very small concentrations (in the ppm or ppb ranges). Increasingly, GLC is also applied for process-integrated analyses, i.e., samples are not processed by a central laboratory in batch mode any more, but immediately when they become available, because the results are needed for real-time process control and product quality assurance. Examples for application areas are all kinds of chemical processes and environmental pollution control.

We shortly outline the operation of a gas-chromatograph: A carrier gas flows, at a constant rate, through a capillary column and a detector. At the inlet of the column there is a sample injection valve. Here, samples can be injected into the carrier gas path. A chromatographic column consists of a moving phase, in which samples are carried along the surface of a stationary phase. In gas-liquid-chromatography, the moving phase is a carrier gas and the stationary phase is a thin layer of a liquid solvent on the inner surface of a glass or metal capillary. For a sample, there is at any time equilibrium between the moving and the stationary phases along the entire column length. It is specific for a compound if, how, and for how long a time it is sorbed by the stationary phase, which constitutes the analytical principle of gas-liquid-chromatography. Components that are sorbed least onto the surface of the stationary phase remain in the column for the shortest time. Separated components are then eluted from the column and passed through the detector, which produces an electrical signal proportional to the mass of substance leaving the column in the carrier gas. This signal is amplified or attenuated, zero-corrected, and Galvanically conditioned to yield a desired output.

As output information a chromatograph yields extinction-time-functions exhibiting typical peak patterns, see Figure 7.11. Each peak corresponds to a compound present in the analyzed sample.
The time of the peak maximum, called *retention time*, is characteristic for the substance giving rise to it, and the peak’s integral is proportional to the compound’s concentration.

The following parameters have to be determined from chromatograms:

- number of peaks,
- retention times of the different peaks,
- areas of the different peaks, and
- total area under the peaks.

A detection procedure is responsible for locating the peaks. This is best illustrated by an example of an idealized peak, see Figure 7.12. The chromatogram is scanned linearly from the left to the right. As long as ordinate, slope and curvature are within certain limits (1), nothing happens. The detection procedure searches for a peak start, characterized by a certain minimal ordinate, slope and curvature (2). If slope and curvature remain above minimal values for a specified time (3), position 2 is marked as a peak start. Now the search continues for the front *inflection point*, the position where the curvature becomes negative (4). If the slope becomes negative (5), the corresponding point is marked as the top. The search continues for the rear inflection point (6) and finally, for the end of the peak which is approached when both slope and curvature
assumeme values (7) within given limits and remain within these limits for a specified time (8). The latter point is marked as the end of the peak. The area of the peak (between (2) and (8)) can now be calculated.

After top-, start- and end-time have been determined, the peak will be accepted as real chromatographic information if its width and height exceed certain limits. Otherwise, the peak is assumed to be the result of a disturbance.

In general, however, the situation is a little more complex. Peaks may be grouped, or small peaks may be superimposed on the slope of larger ones. In the first case, the common area under the peaks must be divided among the members of the group. In the second case, one has to determine which part of the area under a small peak belongs to a large peak. Another difficulty is that the zero-value of the chromatogram, i.e., the extinction when no peaks are present, varies along the chromatogram due to several physical effects. Therefore, one defines the so-called baseline, which is a smooth curve underneath the peaks. Before calculating the peak surfaces, the baseline must be subtracted from the chromatogram. This process is called baseline correction.

In most cases, a chromatogram \( x \) is uniformly sampled. Baseline correction and peak detection operate on the sampled data. The sampling interval \( \Delta \) depends on the type of chromatography, but is typically around 20 ms for gas-chromatography. This corresponds to a sampling frequency exceeding the Nyquist frequency, but the redundancy is used to remove noise by digital filtering. The curve must be smoothed in order to prepare for the numerically unstable calculation of the curve’s first two derivatives in each data point, which is necessary to determine extrema and inflection and baseline points. In the next section we will see that a well chosen representation of chromatograms will considerably facilitate both baseline correction and peak detection.

### 7.4.2 B-spline representations

We assume that a chromatogram \( x \) is measured on the interval \([0, T]\), yielding \( K \) samples \( (\Delta = T/K) \). We choose to represent the chromatogram as a quadratic spline, expanded on a basis of second-order B-splines (cf. (4.56) in Section 4.5.3):

\[
\hat{x}(t) = \sum_{k=0}^{K+1} c_k \beta^2 \left( \frac{t}{\Delta} - k + \frac{1}{2} \right). \tag{7.1}
\]

This representation is equal to (4.56), but with a dilation factor of \( \Delta \), and a translation factor of \( k - 1/2 \). The coefficients have to be chosen in such a way that the representation \( \hat{x} \) is a good approximation of \( x \). This problem is a familiar one. In fact, the representation can be regarded as the reconstruction part of the general sampling scheme. For an \( L^2 \)-optimal reconstruction, the coefficients must be obtained by pre-filtering with a \( D_2 \)-spline, followed by uniform sampling. However, in many cases sampled data are given. Then we can either choose to use the samples themselves (thereby implementing a B-spline reconstruction), or perform digital filtering on the samples to implement other reconstructions, e.g., the \( C_2 \)-spline interpolation (see Section 7.1.3). The two extra coefficients \( c_0 \) and \( c_{K+1} \) in the representation (7.1) are given the same values as their neighbors.

It has to be noted that the \( L^2 \)-optimal reconstruction may not provide the best result. Since peak areas must be preserved in the representation, an \( L^1 \)-approximation may give better results.
The general definition of B-splines was given by (4.57) on p. 76. For the second-order B-spline, this reduces to

\[
\beta^2(t) = \begin{cases} 
\frac{1}{2}(t + \frac{3}{2})^2, & t \in [-\frac{3}{2}, -\frac{1}{2}] \\
\frac{3}{4} - t^2, & t \in [-\frac{1}{2}, \frac{1}{2}] \\
\frac{1}{2}(t - \frac{3}{2})^2, & t \in [\frac{1}{2}, \frac{3}{2}] \\
0, & \text{otherwise}
\end{cases}
\]

Hence, the representation \( \hat{x}(t) \) reduces in each of the subintervals \([k-1]\Delta, k\Delta], \ k = 1 \ldots K\) of \([0, T]\), to a quadratic polynomial and its continuous first derivative \( \hat{x}'(t) \) consists of straight line segments. The latter's zeros and, hence, the extrema of \( \hat{x}(t) \), can be determined easily with little effort. This holds even more for the inflection points of \( \hat{x}(t) \), which must coincide with some of the nodal points \( k\Delta \). In contrast to standard numerical differentiation, determination of the derivatives based on the B-spline representation is numerically stable.

All baseline correction algorithms described in the literature make certain assumptions on the shape of the chromatographic signal. This is, however, unnecessary: the baseline can be seen as very smooth version of the chromatogram, with the additional property that it always lies underneath the chromatogram. For the baseline \( b \), we use a similar representation as for the chromatogram itself, but with a larger ‘sampling interval’ \( \Delta_t \)

\[
b(t) = \sum_{k=0}^{L+1} a_k \beta^2 \left( \frac{t}{\Delta_t} - k + \frac{1}{2} \right),
\]

where \( M \) and \( L \) are integers such that \( K = L \cdot M \), and \( \Delta_t = M\Delta \). The baseline will become smoother if \( M \) increases.

The coefficients \( a_k \) must be determined in such a way that the baseline remains under the chromatogram and the \( L^1 \)-norm of the difference signal is minimized. Choosing a number of control points on the chromatogram, this minimization problem reduces to a linear program, which can efficiently be solved by the so-called Simplex method [HL92]. Because B-splines satisfy a two-scale relation (see Section 4.5.3)\(^4\), they can be written in terms of dilated copies of themselves. This yields another representation of the baseline on the original scale \( \Delta \):

\[
b(t) = \sum_{k=0}^{K+1} b_k \beta^2 \left( \frac{t}{\Delta} - k + \frac{1}{2} \right).
\]

Baseline correction is now carried through by replacing the coefficients \( c_k \) in (7.1) by the new coefficients \( d_k := c_k - b_k \).

### 7.4.3 Analysis software

The algorithms mentioned above can easily be implemented on a personal computer. A software package based on this approach was developed by P.J. de Boer [dB94]. It provides a graphical user interface to facilitate further analysis. The software runs under MS-Windows and provides a

\(^4\)It was noted that the two-scale relation only holds for splines of odd order. However, due to the extra translation by \( 1/2 \), all nodal points are located at the sample times. In this case, the two-scale relation also holds for even order B-splines.
user-friendly environment. Due to the object-oriented nature of MS-Windows, it was convenient to use an object-oriented programming language. The program was written in C++, using Borland’s **ObjectWindows** library. For details on design and implementation, see [dB94].

The main task of the software is to calculate and present the parameters of chromatograms. In addition, the software has the following features:

- graphical presentation of chromatograms, with scrolling and zooming,
- presentation of peak parameters in tables,
- peak selection by the user in either table or graph; if a peak is graphically selected, the corresponding table entry is highlighted and vice versa,
- storage of raw and processed data,
- capability of processing and displaying multiple chromatograms simultaneously,
- interactive modification of the peak-detection parameters (for each chromatogram independently).
- hardcopy of figures and tables for documentation purposes, and
- full on-line help.

Some chromatographic readings were analyzed using the program and compared with the results of a commercial peak integration system (Hewlett Packard HP 3396 Series II). At first sight, the program gives satisfactory results, but extensive comparison has not been carried out, yet.

Some additional features, present in the HP 3396, but not yet included in our program, are:

- manual acceptance/rejection of peaks, and
- solvent peak\(^5\) rejection.

Moreover, analysis could further be supported by a database with the retention times of known substances.

### 7.5 Concluding remarks

In this chapter we have discussed a number of implementation techniques for the alternative sampling schemes. There are different possibilities for implementing the alternative sampling schemes, either in analog or in digital hardware. The latter choice, however, involves additional A/D- and D/A-converters, which increase the total cost. In applications, where it is important

\(^5\)The solvent gas/liquid gives rise to a large peak, which contributes to a chromatogram’s total area, causing the relative areas of the other peaks to become small. Since one is not interested in this compound, the corresponding peak must not be taken into account.
to have an optimal representation of an analog signal with a minimum sampling rate, it may be worthwhile to put in the extra effort and costs, and to use one of the optimal sampling schemes. Especially the B-spline filters are suitable for analog implementation. It is still possible to obtain an $L^2$-optimal scheme by digital filtering prior to reconstruction. The $B_0$-spline pre-filter reduces to local signal integration, for which a robust, low-cost implementation exists, based on Voltage-to-Frequency Converters. From theoretical analysis, local integration has some inherent advantages compared to standard point measurements. To verify this, a prototype of an integrating A/D-converter was built. Measurements taken by this device clearly show suppression of noise and harmonic disturbances.

In other cases, sampled data are further analyzed by digital computers. Hence, D/A-conversion is not required. By means of the Gas-Liquid-Chromatography example it was shown that the theory developed for signal reconstruction can also be applied to find a good representation of a signal. The representation of chromatograms as spline functions simplifies the extraction of analytical parameters.

To prove the effectiveness of such processing, special analysis software was developed, which carries out all pre-processing and presents the resulting data in a convenient way. Extensive tests of this software-prototype have not been carried out, yet.