Abstract

A summary of basic digital signal processing systems is provided. Methods currently used in gamma-ray spectroscopy based on digital techniques are summarized. A list of references regarding digital spectroscopy is provided to guide the reader to relevant work.
Introduction

There is trend towards increase use of digital techniques, when analysing signals from detectors. There are several advantages of using digital techniques but also disadvantages, as compared to using traditional analog techniques. Some advantages are:

- Digital analysis methods provide an almost infinite flexibility, e.g. regarding choice of parameters for signal filtering.
- Digital filtering can be performed in a way that is difficult or even impossible with analog techniques. E.g. triangular, trapezoidal or cusp-shaped pulse shapes can be realised using relatively simple digital algorithms but they are very hard to implement using analog electronics.
- A more detailed analysis can be performed on pulses from multiple detector segments or from pixelated detectors.
- Increased stability. The signal will be stable after digitizing since it is stored in memory, it does not depend on e.g. temperature anymore.
- Perfect linearity. The digitized signal can be used in mathematical algorithms. In an analog system, there will always be sources of non linearity, e.g. from small feedback loops.

Some disadvantages are:

- The information in the digitized signal is limited by the frequency of the sampling unit (the ADC). This could be an important limit when very fast timing information is needed from the signal.
- A fast ADC and digital electronics to process the digitized signal require more power than an equivalent analog shaper.

Pulse shapes

[KNOLL] describe signals generated from detectors and show how they can be described mathematically. An exponential pulse shape is often a good approximation of the signal from the pre-amplified signal from a semiconductor detector, e.g. a High Purity Germanium (HPGe) detector. This approximation is valid when the electrons and holes in the semiconductor have similar drift mobility. The rise time of a signal is defined as the time between 10 and 90 percent of the amplitude of the (rise of the) pulse. The rise time of a pre-amplified HPGe signal is in the order of $10^1$ ns while the decay time is in the order of $10^1 \mu s$.

In semiconductor detectors with dissimilar drift mobility of the electrons and holes, e.g. CdTe, the rise of the pre-amplified pulses will contain two components (from the electrons and holes) and will depend on the point of gamma-ray interaction within the depletion field in the semiconductor.

Analog to Digital Conversion

Analog to digital conversion is the process of conversion between analog and digital signals. The digital signal is a series of discrete samples of the analog signal. The conversion process is performed by an analog to digital converted (ADC). The ADC repeatedly measures the voltage of an analog signal and produces a digital number corresponding to that voltage.

The sampling frequency of the ADC determines how often the analog signal is measured and converted to a digital word. Between the 1980's and the beginning of 21st century, the sampling frequency increased from tens of MHz to the GHz range. In the same time period, the number of bits in the digital word produced by the ADC increased from around 8 bits to around 24 bits, depending on sampling speed and internal architecture [IEEE_2005].
Let

- \( N \) = The number of bits in the ADC, the resolution.
- \( n_{ADC} \) = The digital number (integer) produced: \( 0 \leq n_{ADC} \leq 2^{N-1} \)
- \( v_{in} \) = The input voltage to be sampled by the ADC.
- \( V_U \) = The upper end of the ADC input voltage range.
- \( V_L \) = The lower end of the ADC input voltage range.

The digital number produced by the ADC can then be calculated as follows, i.e., the calculated number is rounded to the nearest integer.

\[
n_{ADC} = \text{INT} \left( (2^N - 1) \cdot \frac{(v_{in} - V_L)}{(V_U - V_L)} + \frac{1}{2} \right)
\]

The input voltage range of the ADC should match as close as possible the range of pulse heights expected from the detector. The ideal ADC will produce a linear translation between input voltage and the digital word. In reality however, there are non linearity to be taken into account. Two types of deviations from linearity are defined as follows.

**Integral Non Linearity**  
The maximum deviation from the ideal linear translation. Typically this deviation is less than 0.1 %.

**Differential Non Linearity**  
The deviation in the number of recorded events within a sub-interval of the range, from the ideal number expected in that range. Typically, this deviation is much less than 1 %.

**ADC types**

The analog to digital conversion can be realized in many different electronic design. We list some common ADC types below. E.g. ref. [KNOLL] contains a more in-depth description of ADC's.

**Flash ADC**

The flash ADC, also called a direction conversion ADC, digitize the analog signal by parallel comparison with a set of comparators. This type of ADC is usually very fast but is limited to the number of comparators that can be constructed in the circuit.

**Successive-approximation ADC**

In a successive-approximation ADC, the input analogue voltage is successively compared to an internal reference voltage (generated by a digital to analog converter). For each comparison step, the reference voltage is set to the midpoint of the current comparison interval. The digital word is stored in a successive approximation register (SAR) and is updated for each comparison. This type of ADC is relatively slow but can provide a higher number of bits.
Ramp-compare ADC

The ramp-compare ADC works by measuring the time it takes for an internally generated saw-tooth signal to reach the level of the input analogue signal. The saw-tooth signal is ramped up or down very fast, to be able to get a fast "conversion". The ramp-compare ADC is relatively easy to construct since it requires few components but it is relatively slow.

Wilkinson ADC

The Wilkinson ADC compares the input analog voltage to an internally generated voltage produced by a charging capacitor. When the voltage over the charging capacitor reaches the same level as the input analog voltage, it is then made to discharge linearly. The time it takes for the linear discharge to reach zero voltage is then measured by counting the number of cycles in an internal high-frequency oscillator clock.

Effective number of bits

All signals measured with an ADC will contain some level of noise. When the lower level of the input range of the ADC is below the noise level, the lowest significant bits of the output word will represent the noise. Subtracting these bits from the output word results in an effective number of bits that represent the resolution when converting the interesting part of the signal, without the noise. The effective number of bits (ENOB) can be calculated using

\[ ENOB = \frac{SINAD - 10 \cdot \log_{10}(3/2)}{20 \cdot \log_{10}(2)} \approx \frac{SINAD - 1.76}{6.02} \]

where SINAD is the ratio of the total signal including distortion and noise to the wanted signal. (Assignment: Derive this formula.)
Signal processing systems

Discrete time-signals

A continuous function \( f(x) \) is usually defined for any real value \( x \). Discrete time-functions \( f(n) \) however, are defined for discrete values \( n \). For time-signals that we consider here, the discrete values \( n \) are points in time. The ADC sample the continuous voltage \( u(t) \) at discrete times \( u(t_n) \) which is conveniently written as \( u(n) \), with \( n \) representing the sample number.

Some examples of basic discrete time-signals are listed below.

**Unit impulse**

The unit impulse is defined as:

\[
\delta(n) = \begin{cases} 
1, & n = 0 \\
0, & n \neq 0 
\end{cases}
\]

A unit impulse function with impulse strength \( a \) is defined as follows. Figure 1 shows the unit impulse function for some values if \( n \).

\[
x(n) = a\delta(n)
\]

*Figure 1. The unit impulse function with impulse strength \( a \).*
Unit step

The unit step is 1 for all $n \geq 0$, and is 0 elsewhere. It is illustrated in figure 2.

$$u(n) = \begin{cases} 
1, & n \geq 0 \\
0, & n < 0 
\end{cases}$$

Figure 2. The unit step function.

Unit doublet

$$\delta(n) = \delta(n) - \delta(n-1) = \begin{cases} 
1, & n = 0 \\
-1, & n = 1 \\
0, & \text{elsewhere} 
\end{cases}$$

Figure 3. The unit doublet.
Exponential signals

Exponential signals can be described with the following formula. Depending on the parameter $a$, the shape of the sequence of points varies as can be seen in figure 4. Note that $u(n) = e^{-n}$ if $a = e^{-1}$

$$u(n) = a^n$$

Figure 4. Exponential signals. The shape of the signal depend on the value and sign of the parameter $a$. 
**Linear and time-invariant systems**

A function that transforms an input sequence $x(n)$ into an output sequence $y(n)$ is called a *transfer function*. The transfer function $F$ is operating on the input sequence $x(n)$, where $-\infty < n < \infty$:

$$y(n) = F[x(n)]$$

We can illustrate the transfer function accordingly:

$$x(n) \xrightarrow{F} y(n)$$

The transfer function is *linear* if

$$y(n) = F[a_1x_1(n) + a_2x_2(n)] = a_1F[x_1(n)] + a_2F[x_2(n)]$$

The transfer functions that we consider when analysing detector signals are linear. It can be noted that the z-transform that is used in some of the operations to be defined later is a linear transform, see e.g. ref. [BETA].

If, for every $k$, we have

$$y(n) = F[x(n)] \Rightarrow y(n - k) = F[x(n - k)]$$

we then say that $F$ is *time invariant*, i.e. the transfer function $F$ does not change with time.
Discrete signal operations

In order to construct a machinery for executing algorithms on digital signals, some basic operations must be defined. We define addition, subtraction, multiplication, integration and differentiation on a sequence of digital numbers as follows. Note the use of the z-transform in some functions. The z-transform on a infinite sequence of numbers \( \{x(0), x(1), \ldots\} \) is defined by

\[
X(z) = \sum_{n=0}^{\infty} x(n)z^{-n}
\]

<table>
<thead>
<tr>
<th>Operation</th>
<th>Formula</th>
<th>Diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addition</td>
<td>( y(n) = v(n) + x(n) )</td>
<td><img src="#" alt="Addition Diagram" /></td>
</tr>
<tr>
<td>Subtraction</td>
<td>( y(n) = v(n) - x(n) )</td>
<td><img src="#" alt="Subtraction Diagram" /></td>
</tr>
<tr>
<td>Multiplication</td>
<td>( y(n) = v(n)x(n) )</td>
<td><img src="#" alt="Multiplication Diagram" /></td>
</tr>
<tr>
<td>Multiplication with a constant</td>
<td>( y(n) = m x(n) )</td>
<td><img src="#" alt="Multiplication with Constant Diagram" /></td>
</tr>
<tr>
<td>Operation</td>
<td>Equation</td>
<td>Diagram</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Delay with $k$ samples</td>
<td>$y(n) = x(n-k)$</td>
<td><img src="#" alt="Delay Diagram" /></td>
</tr>
<tr>
<td>Integration (accumulation)</td>
<td>$y(n) = y(n-1) + x(n)$</td>
<td><img src="#" alt="Integration Diagram" /></td>
</tr>
<tr>
<td>Differentiation</td>
<td>$y(n) = x(n) - x(n-1)$</td>
<td><img src="#" alt="Differentiation Diagram" /></td>
</tr>
</tbody>
</table>
**Convolution**

For two continuous functions $x(t)$ and $h(t)$ in the time domain, the convolution between the two is denoted by the $*$ sign and is defined by

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

In the discrete time domain, the convolution is correspondingly defined by

$$y(n) = x(n) * h(n) = \sum_{i=-\infty}^{\infty} x(i)h(n - i)$$

A system where the output depends only on current and past input values is called a causal system. For a causal system, the convolution integral or sum is restricted to the current and past times:

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

$$y(n) = \sum_{i=-\infty}^{n} x(i)h(n - i)$$

The convolution operation have the following properties for linear and time-invariant systems.

<table>
<thead>
<tr>
<th>Property</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commutative</td>
<td>$h(n) * x(n) = x(n) * h(n)$</td>
</tr>
<tr>
<td>Associative</td>
<td>$[x(n) * h_1(n)] * h_2(n) = x(n) * [h_1(n) * h_2(n)]$</td>
</tr>
<tr>
<td>Distributive</td>
<td>$x(n) * [h_1(n) + h_2(n)] = x(n) * h_1(n) + x(n) * h_2(n)$</td>
</tr>
</tbody>
</table>

**Impulse response, IIR and FIR**

Let $x(n)$ be the unit impulse, $\delta(n)$. We then have $y(n) = x(n) * h(n) = h(n)$. In this case, $h(n)$ is called the impulse response.

Assuming that the impulse response $h(n)$ is known for infinitely number of samples, it is then called an infinite impulse response (IIR).

In reality, the impulse response $h(n)$ can only be non-zero in a limited interval. I.e. $h(n)$ settles to zero in a finite number of samples. It is then called a finite impulse response (FIR).
Unfolding-synthesis technique

By construction of appropriate digital filters, the output from a digital processor can be adapted to a shape optimal for further analysis. This technique is comprised of two steps:

1. Unfolding, or deconvolution, of the digitized analog signal.
2. Synthesis of the desired output pulse shape using digital filters.

This two step process will be demonstrated with the example of deconvolution of a digital exponential pulse, followed by synthesis of a triangular pulse shape.

1. Digital exponential pulse deconvolution

Let us assume that the ADC samples from an exponential pulse \( x(t) = A_0 e^{-t/\tau} \) at regular intervals \( \Delta T \). I.e., we get the sequence \( x(n) = A_0 e^{-n \Delta T/\tau} \) for \( n \geq 0 \) (\( x(n) = 0 \) for \( n < 0 \)). It we set \( a = e^{-\Delta T/\tau} \), we then have \( x(n) = A_0 a^n \) and for \( n > 0 \) we have

\[
x(n) = ax(n - 1)
\]

By subtracting a shifted and scaled version of the original sampled exponential pulse, we get an unfolded exponential signal being simply a unit impulse function with impulse strength \( A_0 \):

\[
y(n) = x(n) - ax(n - 1)
\]

\[
\begin{bmatrix}
y(0) \\
y(1) \\
y(2) \\
\vdots
\end{bmatrix} = \begin{bmatrix} x(0) \\
x(1) \\
x(2) \\
\vdots
\end{bmatrix} - \begin{bmatrix} ax(-1) \\
ax(0) \\
ax(1) \\
\vdots
\end{bmatrix} = \begin{bmatrix} A_0 \\
0 \\
0 \\
\vdots
\end{bmatrix}
\]

The decay time (\( \tau \)) of the exponential pulse can be tuned by changing the RC-circuitry within the pre-amplifier, i.e. it can be tuned so that the scaling factor \( a \) used above is an integer which facilitates integer arithmetic. We can write the resulting signal as a convolution:

\[
y(n) = x(n) * h(n)
\]

with \( h(n) \) being the impulse response:

\[
h(n) = \delta(n) - a\delta(n - 1)
\]

2. Triangular pulse shape synthesis and the flat top

By combining two moving sums as in figure 5 we can see that the resulting output shape is a triangular shape with a flat top.

The output \( h^T(n) \) is the convolution of the two moving sums:

\[
h^T(n) = h^k(n) * h^m(n)
\]
Figure 5. Digital synthesis of a triangular shaper with a flat top, i.e. a trapezoidal shaper.

The Cusp shape

Gatti have shown that for detector charge measurements in the presence of $1/f$ and white parallel and series noise, there is an optimal filter shape to be used for maximising the signal to noise ratio. This is a filter that produce a Cusp-like impulse response, illustrated in figure 6. See e.g. refs. [GATTI-1990,GATTI-1997]. However, there are practical problems to implement such a filter. Further, [JORDANOV] have shown that addition of a flat top reduces the effect of incomplete charge collection, in detectors where the current pulse depends on the interaction point within the detectors, e.g. very large HPGe detectors or CdTe detectors.

Figure 6. The Cusp shape.

Voluntary programming exercise

Sample one pulse shape sampled from a HPGe detector using a fast digitizer. Use the techniques from the references to this document to program a trapezoidal filter. Study the effect on the resulting pulse shape when parameters for the filter are varied. Study especially the effect of ballistic deficit on the shape of the output pulse.
Kalman filters

In Kalman smoothing (see ref. [KALMAN]), each individual measurement is used together with the information already measured to enhance an estimate of some unknown parameter. In an iterative approach, the uncertainty in the estimate is reduced by each subsequent measurement. The basic concept behind Kalman filtering is displayed in figure 7.

![Figure 7. The basic steps of Kalman filtering. Predictions and updates are iterated whenever a new measurement is performed. (Image from Wikipedia Commons, licenced under Creative Commons CC0 1.0 Universal Public Domain Dedication.)](image)

Although currently not to be found in the main stream digital techniques in use in gamma-ray spectroscopy, the ADONIS ("Atelier de Développement Numérique pour l'Instrumentation en Spectrométrie", see e.g. ref. [ADONIS]) system is using Kalman smoothing on the sampled pulse shapes. The purpose of this approach is to address count-rate limitations that exists both in classical analog and most digital systems on the market. It has been demonstrated in ref. [BARAT] that using Kalman smoothing leads to a dramatic improvement in e.g. energy resolution for very high count rates, compared to conventional techniques (both analog and digital).

**Post-processing**

After the pulse height of the detector pulse has been determined, one of two types of post-processing is usually performed, depending on the measurement requirements.

In list mode, also called event-bye-event mode, each pulse height value is stored in a list together with a time stamp of the pulse.

In histogram mode, or spectrum mode, the frequency distribution of pulse height values is stored. This pulse height spectrum loses the timing information of the individual pulses.
Acknowledgements

In 2013, the production of this material was supported by the pedagogical development fund at the Department of Physics and Astronomy, Uppsala University, Sweden.

References

[ADONIS]

[BARAT]

[BETA]
Lennart Råde, Bertil Westergren; Mathematics Handbook for Science and Engineering

[IEEE_2005]

[GATTI-1990]

[GATTI-1997]

[JORDANOV]

[KALMAN]

[KNOLL]