Nuclear Instruments and Methods in Physics Research A 670 (2012) 18–24

Contents lists available at SciVerse ScienceDirect

Nuclear Instruments and Methods in Physics Research A

journal homepage: www.elsevier.com/locate/nima

Exponential signal synthesis in digital pulse processing

Valentin T. Jordanov

Yantel, LLC, Los Alamos, New Mexico, USA

ARTICLE INFO

Article history:
Received 25 July 2011
Received in revised form
30 November 2011
Accepted 5 December 2011
Available online 23 December 2011

Keywords:
Signal synthesis
Exponential signal
Digital pulse processing
Digital signal processing
Pulse shaping
Cusp shape
Radiation

ABSTRACT

Digital pulse processing allows the synthesis of exponential signals that can be used in pulse shaping and baseline restoration. A recursive algorithm for the synthesis of high-pass filters is presented and discussed in view of its application as a baseline restorer. The high-pass filter can be arranged in a gated baseline restorer configuration similar to widely used analog implementations. Two techniques to synthesize time-invariant, finite impulse response (FIR) cusp shapers are presented. The first technique synthesizes a true cusp shape in the discrete-time domain. This algorithm may be sensitive to round-off errors and may require a large amount of computational resources. The second method for synthesis of cusp shapes is suitable for implementation using integer arithmetic, particularly in hardware. This algorithm uses linear interpolation to synthesize close approximations of true cusp shapes. The algorithm does not introduce round-off errors and has been tested in hardware.

1. Introduction

Exponential signals are essential for the formation of detector signals and pulse shaping in radiation measurements. Exponential signals can be found in noise analysis and optimal pulse shaping [1,2]. Historically RC–CR circuits have been used to shape detector pulses. The R–C low pass filter has an impulse response, which is a decaying exponential signal. The step response of the C–R high pass filter is also a decaying exponential signal. With the development of digital techniques to process detector pulses various pulse shape synthesis algorithms were introduced that resemble cusp shapes [3,4]. This paper describes efficient digital techniques to synthesize exponential signals including the true cusp shape. The algorithms and the system descriptions use discrete-time signal notations, basic signal definitions and graphical system blocks as defined in reference [5]. It should be noted that the functional block diagrams are graphical representations of digital signal processing operations and not of actual hardware blocks or software routines. These operations accept and generate a sequence of signal samples by performing instantaneous operations. Synchronous hardware designs will require accurate data synchronization, which can be achieved by accounting for any delays associated with the hardware-implemented arithmetic and logic operations.

2. Digital synthesis of exponential signals

Mathematically the exponential signals in the discrete time domain are defined by

$$y(n)=\begin{cases} a^n & \text{for } n\geq 0 \\ 0 & \text{elsewhere} \end{cases} \quad (1)$$

where a is called an exponential base [5]. In digital signal processing n is the index of the consecutive values (samples) of the discrete-time signal. If $a=0$ or $a=1$ then all samples $y(n)$ for $n\geq 0$ are constant (equal to 0 or 1, respectively). If the exponential base a is greater than 0 but less than 1, $y(n)$ is a decaying exponential signal. If a is greater than 1 then $y(n)$ is a growing exponential signal. If a is negative then $y(n)$ alternates between positive and negative numbers. In this paper we consider the exponential base a to be greater than zero.

From Eq. (1) the ratio of two consecutive values of an exponential signal can be expressed as

$$\frac{y(n)}{y(n-1)}=\frac{a^n}{a^{n-1}}=a \quad (2)$$

for $n>0$ and $y(0)=1$.

Using Eq. (2), a growing or decaying exponential signal can be expressed in the following recursive form

$$y(n)=ay(n-1) \quad (3)$$

for every n, given the initial conditions $y(0)=1$ and $y(n)=0$ for $n<0$.

E-mail address: jordanov@ieee.org

0168-9002/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.nima.2011.12.042

The goal of exponential signal synthesis is to define a linear time-invariant (LTI) recursive system that produces an exponential signal in response to an input signal $x(n)$. This system can be described using a first order difference equation [5]

$$y(n) = x(n) + ay(n-1) \tag{4}$$

The output signal $y(n)$ represents an exponential signal only when the conditions $y(0)=1$ and $y(n)=0$ for $n<0$ are fulfilled. These conditions are met when the system is relaxed [5] by forcing delayed term $y(n\text{-}1)=0$ for $n<0$, and when the input signal $x(n)$ is the unit impulse $\delta(n)$ ($x(n)=\delta(n)$). In this case the output of the system $y(n)$ is the system's infinity impulse response (IIR) $h(n)$

$$h(n) = \delta(n) + ah(n-1) \tag{5}$$

for $n \geq 0$ and $h(n)=0$ for $n<0$. It is clear that this impulse response is an exponential signal, which grows or decays infinitely in time.

The functional block diagram of a system with an exponential impulse response, as defined by Eq. (5), is illustrated in Fig. 1b. The recursive algorithm requires three functional blocks: an adder, a unit delay and a constant multiplayer. This system, when excited by a unit impulse $\delta(n)$, will generate exponential signals that are either growing (Fig. 1c) or decaying (Fig. 1d). The growth/decay rate is determined by the magnitude of the multiplication coefficient a which is the exponential base of the output exponential signal. The system depicted in Fig. 1b is a basic functional block for the synthesis of cusp shapers that will be discussed later in this paper.

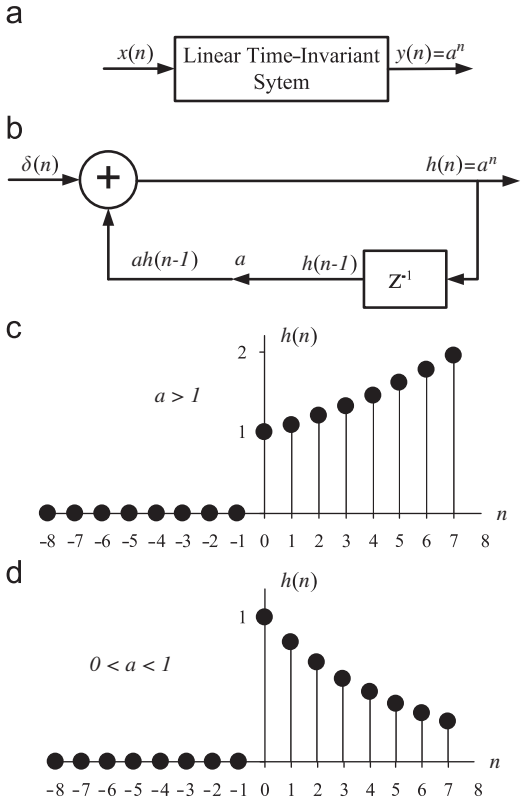

Fig. 1. Block diagrams: (a) of a system synthesizing exponential signals and (b) exponential unit impulse response. Examples are shown of: (c) a growing exponential impulse response and (d) a decaying exponential impulse response.

3. High-pass filter

High pass filters are used in pulse shaping as well as in baseline restorers of radiation spectrometers [2,6,7]. The C–R differentiation is widely used as a pulse conditioning circuit in spectroscopy amplifiers. In some cases pole-zero cancelation circuits are used to achieve a single real pole exponential response. The step response of a high-pass C–R filter is a decaying exponential signal with decay time constant equal to CR.

In the discrete-time domain a decaying exponential signal can be synthesized as a response to a unit step $u(n)$ ($x(n)=u(n)$). In Fig. 2a functional block diagram of such a system is shown. The system implements a recursive algorithm using a constant multiplier, an accumulator (ACC) and a subtractor. It is important that an initial condition of the system is established. Particularly, the accumulator needs to be reset to zero before any non-zero digital signal values are applied to the system input, e.g $w(n)=0$ for $n<0$.

At any time the accumulator output $w(n)$ contains the sum of all weighted output values preceding $y(n)$.

To derive the unit impulse response of the system the accumulator response is expressed as a function of the weighted output signal. The accumulator response is given by the sum of all previous $cy(i)$ values, and $w(n)$ is the same, shifted by a unit delay

$$w(n) = \sum_{i=-\infty}^{n-1} cy(i) \tag{6}$$

Let $x(n)=u(n)$ and $w(n)=0$ for $n<0$. These conditions define $y(n)=0$ for $n<0$. The unit step response of the system $y(n)$ is then given by the following equation:

$$y(n) = x(n) - w(n) = u(n) - \sum_{i=0}^{n-1} cy(i) \tag{7}$$

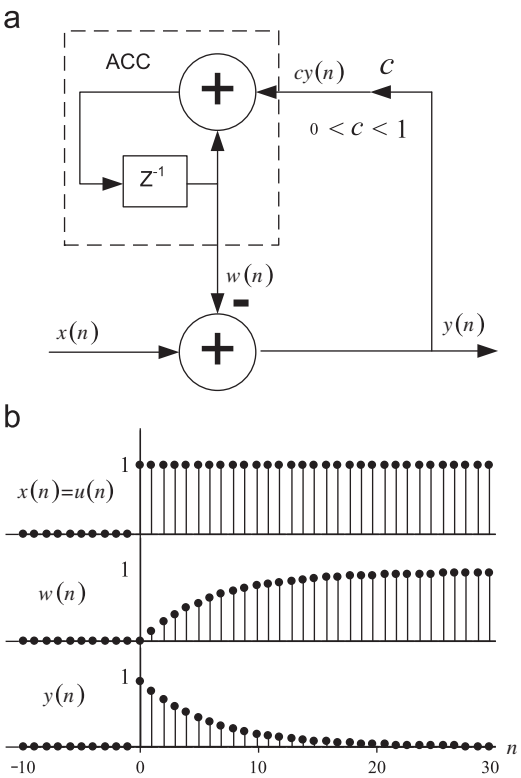

Fig. 2. A digital high-pass filter: (a) functional block diagram and (b) its response to a unit step signal.

By definition $u(n)=0$ for $n<0$ and $u(n)=1$ elsewhere. Thus, Eq. (7) can be expressed in a recursive form for $n>0$

$$y(n)=(1-c)y(n-1) \tag{8}$$

If $a=1-c$ then Eq. (8) becomes equivalent to Eq. (3), which defines an exponential signal. Therefore, the unit step response of the discrete-time system in Fig. 2a is an exponential signal. The exponential base of this signal is determined by the multiplication constant c. If $0<c<1$ the system is a discrete-time equivalent to the C–R differentiation network in the continuous time domain. Fig. 2b shows the digital signals synthesized by a digital high-pass filter.

This filter may be used as a gated high pass filter for baseline restoration and DC stabilization similar to analog implementations [6,7]. Base line restorers affect to some degree the noise performance of pulse shapers [8,9,10]. Historically, the baseline restorers were introduced to remove the baseline fluctuations caused by AC coupling networks [8]. Digital pulse processing algorithms allow removal of analog responses of CR and RC networks, which virtually eliminates the need of baseline restorers. The only function of the baseline restorers in such systems is to compensate for DC offsets and drifts. If the high-pass filter described here is used as a gated baseline restorer and its time constant is very large then the baseline restorer impact on the shaper noise-suppression function will be minimal [10]. If, however, a finite impulse response of the entire pulse processing system cannot be achieved, the choice of baseline restorer should be made by considering optimal baseline restoration techniques [8,9,10].

4. Digital cusp shaper

Cusp shapers have been recognized as the optimal filters for high resolution spectroscopy in the presence of white noise when the signal duration is much longer than the noise corner time constant [1,2]. In order to determine the noise corner time constant a detailed determination of the noise properties of the spectroscopy system must be performed. In addition, the pulse pile-up and the counting throughput requirements impose constraints that not only limit the total filter duration, but also define the optimal pulse shape. For example, a triangular shape is optimal for a duration much shorter than the noise time corner constant in the case of a series and parallel only noise model. The presence of $1/f$ noise will also have an effect on the optimal pulse shape [11].

Various publications and text books provide detailed noise and optimal pulse shaping analysis [1,2,6,11–13]. The selection of optimal pulse shapers is beyond the scope of this paper. This paper focuses on the synthesis of cusp shapers that may or may not be optimal for a given spectroscopy setup. A practical cusp shaper would have a finite exponentially rising edge, a flat top, if needed and an exponentially decaying edge with finite duration. Although of little use, a decaying edge with infinite duration can also be synthesized and is presented in this paper only as a general consideration.

Analog detector signals are normally conditioned and applied to a digitizing ADC to be converted into discrete-time signals. The process of digital pulse shaping can be easily performed by first unfolding (deconvoluting) the digitized analog signal, and then synthesizing the impulse response of the desired pulse shape. A system based on the unfolding-synthesis technique is depicted in Fig. 3.

Algorithms are readily available to unfold single and multiple real pole signals (exponential signals) [14]. Digitized step signals from a reset type preamplifier can be unfolded using digital differentiation. Thus, the goal of synthesizing a cusp shaper is to define a linear digital system with an impulse response identical to the desired cusp shape. This synthesis includes both growing and decaying exponential signals. Depending on the type of exponential decay of the cusp shape, two digital shapers can be synthesized. A finite impulse response (FIR) cusp shaper corresponds to a cusp shape with finite decay duration. When the decay is infinite the shaper has an infinite impulse response (IIR).

The impulse response of a cusp shaper is shown in Fig. 4. The cusp shape has three clearly distinguishable regions: RT represents the rising edge of the cusp shape, FT is the flat top portion, and DT is the decaying part of the cusp shape. In the digital domain the duration of each of these regions is expressed by the respective number of digital samples. The durations of these regions are related to the continuous time durations, which are obtained by multiplying the number of samples in each region by the ADC sampling period ΔT.

The synthesis of the cusp shaper can be carried out by synthesizing each region separately and then combining the impulse responses shifted appropriately. Fig. 5 shows the impulse responses $r(n)$, $q(n)$ and $p(n)$ to be synthesized. In this synthesis the number of non-zero digital samples per region is as follows: RT has k samples, FT has m samples and DT has either $k+1$ samples (FIR) or an infinite number of samples (IIR). The number of digital samples

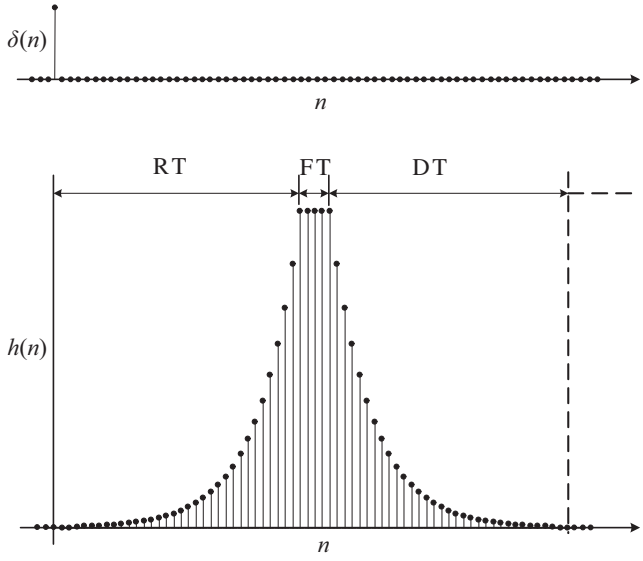

Fig. 4. Impulse response of a cusp shaper.

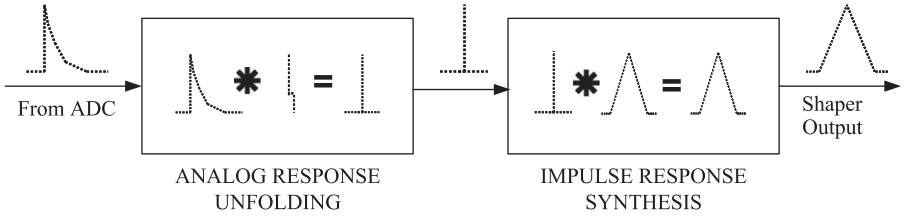

Fig. 3. Pulse shape synthesis using unfolding-synthesis techniques.

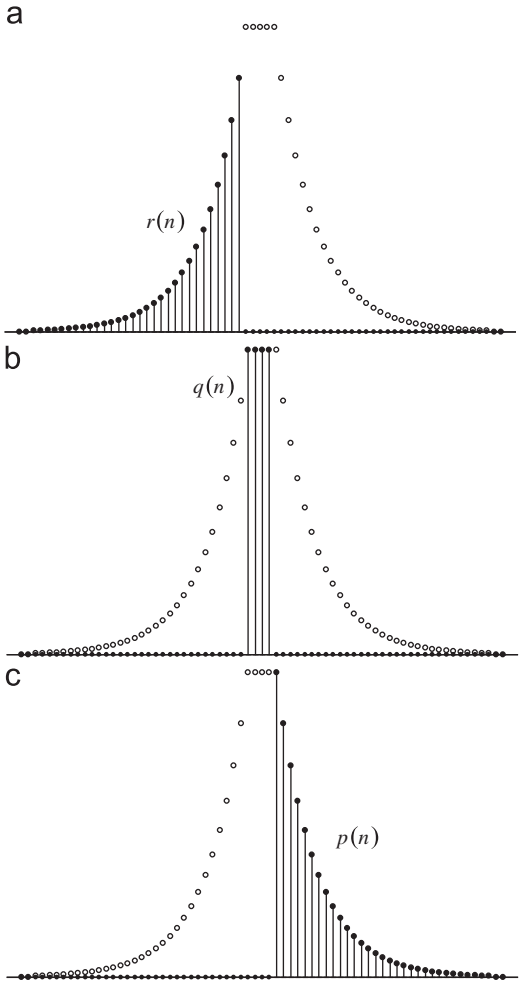

Fig. 5. Partition of the cusp impulse response into three separate impulse responses: (a) exponential growth; (b) flat top and (c) exponential decay.

corresponding to the discrete peaking time of the cusp pulse is $k+m+1$ and its continuous time equivalent is $(k+m+1)\Delta T$. The cusp shape has a maximum equal to a^k. As symmetrical pulse shapes normally offer the optimal noise suppression, here it is considered that the exponential growth and decay rates are the same. It is straightforward, however, to synthesize asymmetrical pulse shapes by making the exponential growth rate different from the exponential decay rate.

The impulse response $r(n)$ has a growing exponential part followed by an abrupt return to zero. The recursive expression of this impulse response is

$$r(n)=\delta(n)+ar(n-1)-a^k\delta(n-k) \tag{9}$$

The recursive term $ar(n-1)$ causes $r(n)$ to grow exponentially reaching a value of a^k after k recursive iterations. At the moment when $r(n)$ becomes equal to a^k the weighted and shifted unit impulse term $a^k\delta(n-k)$ is subtracted from $r(n)$ causing $r(k)$ to become zero. This terminates the exponential growth. Note that the return of $r(n)$ to zero must be exact as any non-zero remnant will grow exponentially due to the natural response of the system, even when the input signal is equal to zero [5].

The cusp shape FIR of the decay part has a sharp rise from zero to the maximum value a^k of the cusp shape followed by a finite exponential decay

$$p(n)=a^k\delta(n-k-m)+\frac{p(n-1)}{a}-\frac{\delta(n-2k-m-1)}{a} \tag{10}$$

The recursive term $(p(n-1)/a)$ causes $p(n)$ to decay exponentially. After k consecutive iterations from $n=k+m$ to $n=2k+m$, the exponentially decaying signal will decay from $p(k+m)=a^k$ to $p(k+m)=a^0$. At the next sample $n=k+m+1$ the term $(p(n-1)/a)$ is equal to $1/a$. At this point the exponential decay of $p(n)$ is terminated ($p(n)$ returns to zero) by a subtraction of the term $(\delta(n-2k-m-1)/a)$.

The IIR of the decay part of the cusp shape is similar, without the term that terminates the exponential decay

$$p(n)=a^k\delta(n-k-m)+\frac{p(n-1)}{a} \tag{11}$$

The flat top is synthesized by convolution of the shifted input samples and a rectangular function

$$q(n)=q(n-1)+a^k[\delta(n-k)-\delta(n-k-m)] \tag{12}$$

Eq. (12) represents the well known box car averaging linear system.

The impulse response of a cusp shaping system is achieved by combining the impulse responses defined by Eqs. (9) and (12) and either Eq. (10)(FIR) or Eq. (11)(IIR). The functional block diagram of the shaper is shown in Fig. 6.

Caution should be exercised when these algorithms are implemented using floating point arithmetic. The return of $r(n)$ to zero after reaching its peak is absolutely necessary. In the process of $r(n)$ synthesis small and very large numbers are added together, which may cause round-off errors. These errors will exponentially propagate and accumulate leading to a numerical overflow.

As the native data from analog to digital converters are integers, the use of integer arithmetic is preferable, especially in the case of real time pulse shape synthesis. However, integer arithmetic imposes additional constraints that may limit the flexibility of choosing cusp shape parameters. These limitations are primarily caused by the restricted fractional multiplication in integer arithmetic. In addition, exponential signals may require digital words with a large number of bits, which may reduce the speed of the algorithms synthesized in hardware, e.g. in FPGAs.

5. Cusp shaper with linear interpolation

One practical solution to synthesize cusp shapes using integer arithmetic is to use an integer exponential base $a=2,3,4\ldots$. The

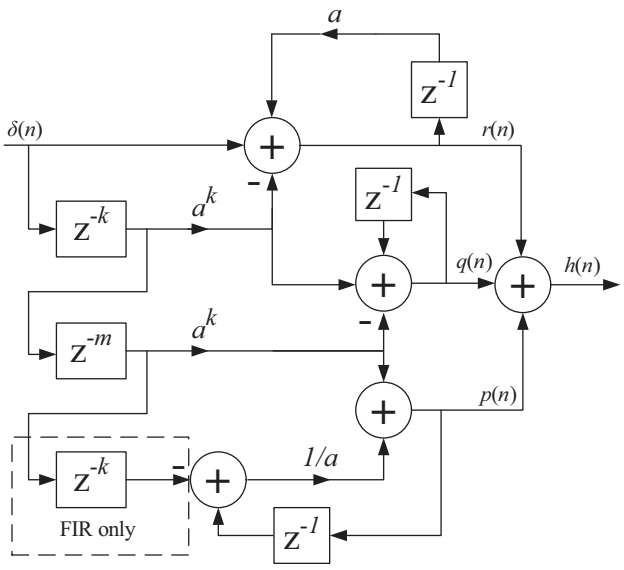

Fig. 6. Functional block diagram of the true cusp shaper.

process of incrementing each consecutive value of the growing exponential $y(n)$ requires a large amount of arithmetic resources which increase the complexity of the cusp shape synthesis. For example, a cusp shape with a peaking time of 100 samples will require a digital word capable of storing a value of a^{100}. If $a=2$ the digital word needs to have length of at least 100 bits to avoid the introduction of round-off errors. Practically, only cusp shapes with a very limited rising edge duration can be synthesized using integer arithmetic operations. To synthesize cusp shapes with longer rising edge durations a close approximation of the cusp shape can be used. Let $K \geq 0$ and $L > 0$ be integer numbers. Let us consider the following equation:

$$y(n) = \begin{cases} \sum_{K=0}^{\infty} a^K \, \delta(n-KL) & \text{for} \quad n \geq 0 \\ 0 & \text{for} \quad n < 0 \end{cases} \tag{13}$$

Eq. (1) is a special case of Eq. (13). Indeed, if $L=1$ Eq. (13) is identical to Eq. (1). If $L>1$ then the non-zero values of the exponential signal are spread out by introducing gaps between them. The non-zero values of $y(n)$ are $y(KL)=a^K$. This effectively scales down the exponential base relative to the case when $L=1$. Let $y_c(n)$ be an exponential signal defined as

$$y_c(n) = b^n \tag{14}$$

If $y_c\ (KL)=y(KL)$ for every $K \geq 0$, then the non-zero points of $y(n)$ lie on the curve defined by $y_c(n)$ as shown in Fig. 7. Thus, the effective exponential base of Eq. (13) is reduced to

$$b = a^{\frac{1}{L}} = \sqrt[L]{a} \tag{15}$$

The reduction of the exponential base relaxes the requirements for the integer arithmetic operations to use a large number of bits for the synthesis of cusp shapes. However, this exponential base reduction introduces gaps in the exponential signal. These gaps cannot be filled with exact exponential values due to limitations of integer arithmetic, but can be approximated using linear interpolation. The linear interpolation is carried out using integer numerical integration (accumulation) of the exponential signal $y(n)$. The first accumulation is given by the following expression

$$y_I(n) = \begin{cases} \sum_{i=0}^{n} y(i) & \text{for} \quad n \geq 0 \\ 0 & \text{for} \quad n < 0 \end{cases} \tag{16}$$

where $y(i)$ is the signal as defined by Eq. (13).

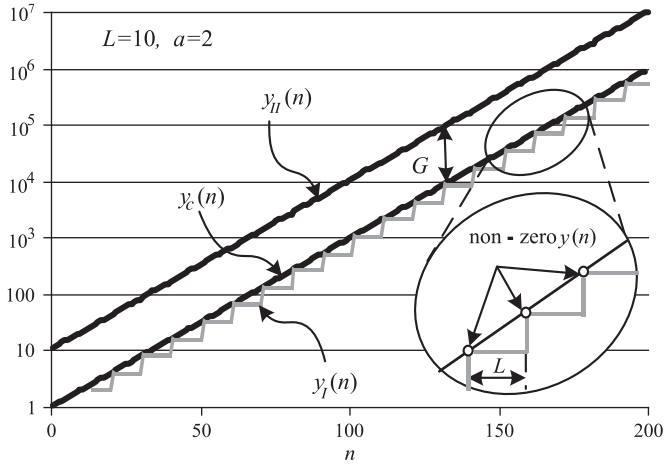

Fig. 7. Exponential signal $y_{II}(n)$ using linear interpolation in comparison to exponential signals $y(n)$, $y_c\ (n)$ and $y_I(n)$.

The resulting signal is stair-case like as shown in Fig. 7. The area enclosed by $y_I(n)$ is found by a second accumulation, which completes the process of linear interpolation

$$y_{II}(n) = \begin{cases} \sum_{i=0}^{n} y_I(i-1) & \text{for} \quad n \geq 0 \\ 0 & \text{for} \quad n < 0 \end{cases} \tag{17}$$

The linearly interpolated signal $y_{II}(n)$ closely approximates the exponential signal of Eq. (14), but has a term that causes non-exponential behavior for small values of $y(n)$. This term is a constant term equal to $L/(a-1)$, which can be obtained from Eq. (17) taking into account that the area under $y_I(n)$ is the sum of tightly stacked rectangles with height a^i and length equal to L

$$y_{II}(KL) = \sum_{i=0}^{K-1} La^i = \frac{L}{a-1}a^K - \frac{L}{a-1} \tag{18}$$

The cascaded accumulations of $y(n)$ introduces a gain $G=L/(a-1)$. The goal is to make $y_{II}(n)$ a true exponential signal at the $n=KL$ points. That is $y_{II}(KL)/y_{II}(KL-L)=a$. To achieve this goal a new term is added to Eq. (17), which cancels out the constant $L/(a-1)$, redefining $y_{II}(n)$ as

$$y_{II}(n) = \sum_{i=0}^{n} \left[y_I(i-1) + \frac{L}{a-1}\delta(i) \right] \quad \text{for } n \geq 0 \tag{19}$$

Fig. 7 depicts the synthesized exponential signal using linear interpolation with $a=2$ and $L=10$. The exponential functions defined by Eqs. (13), (14) and (16) are also shown for reference.

Cusp shapes based on linearly interpolated exponential signals are synthesized using Eqs. (13), (16) and (19). For the purpose of real time synthesis all equations need to be used in their recursive form. The recursive form of Eq. (13) is given by the following expression

$$y(n) = \sum_{K=0}^{\infty} a y(n-L)\delta(n-KL) \tag{20}$$

Fig. 8 shows the functional block diagram of an FIR system that synthesizes linearly interpolated cusp shapes. Note that this circuit requires only two more delays and two more adders than the cusp filter in Fig. 6. To synthesize the impulse response let $x(n)=\delta(n)$. The following equations describe the recursive algorithm of this system:

$$r(n) = x(n) + ar(n-L) - a^K x(n-k) \tag{21}$$

$$p(n) = \frac{a^K x(n-k-m) + p(n-L) - x(n-2k-m)}{a} \tag{22}$$

$$q(n) = q(n-1) + [r(n-1) - r(n-L-1)] - [p(n-1) - p(n-L-1)] \tag{23}$$

$$y(n) = y(n-1) + q(n) + \frac{L}{a-1}\left[x(n) - x(n-2k-m-1)\right] \tag{24}$$

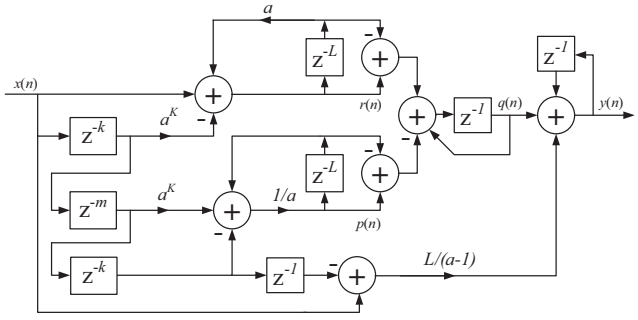

Fig. 8. Functional block diagram of the digital cusp shaper using linear interpolation.

where $r(n)=0$ and $p(n)=0$ for $-L \le n < 0$, $q(n-1)=0$ and $y(n-1)=0$ are the conditions to make the system relaxed [5].

The duration of the rising and the falling edges of the cusp shape is equal to k digital samples. The duration of the flat top is equal to m digital samples. The flat top duration m is an integer equal to or greater than zero. When $m=0$ the maximum of the cusp shape is reached by only a single digital sample. The duration k of the rising and the falling edges is also a positive integer which is, however, restricted to $k=KL$. The total width of the FIR cusp shape is $2k+m+1$. When $K=1$ the linearly interpolated cusp shape transforms to a triangular/trapezoidal shape. The peak value of the linearly interpolated cusp shape is

$$y_{\mathrm{PEAK}} = \frac{L}{a-1} a^K \tag{25}$$

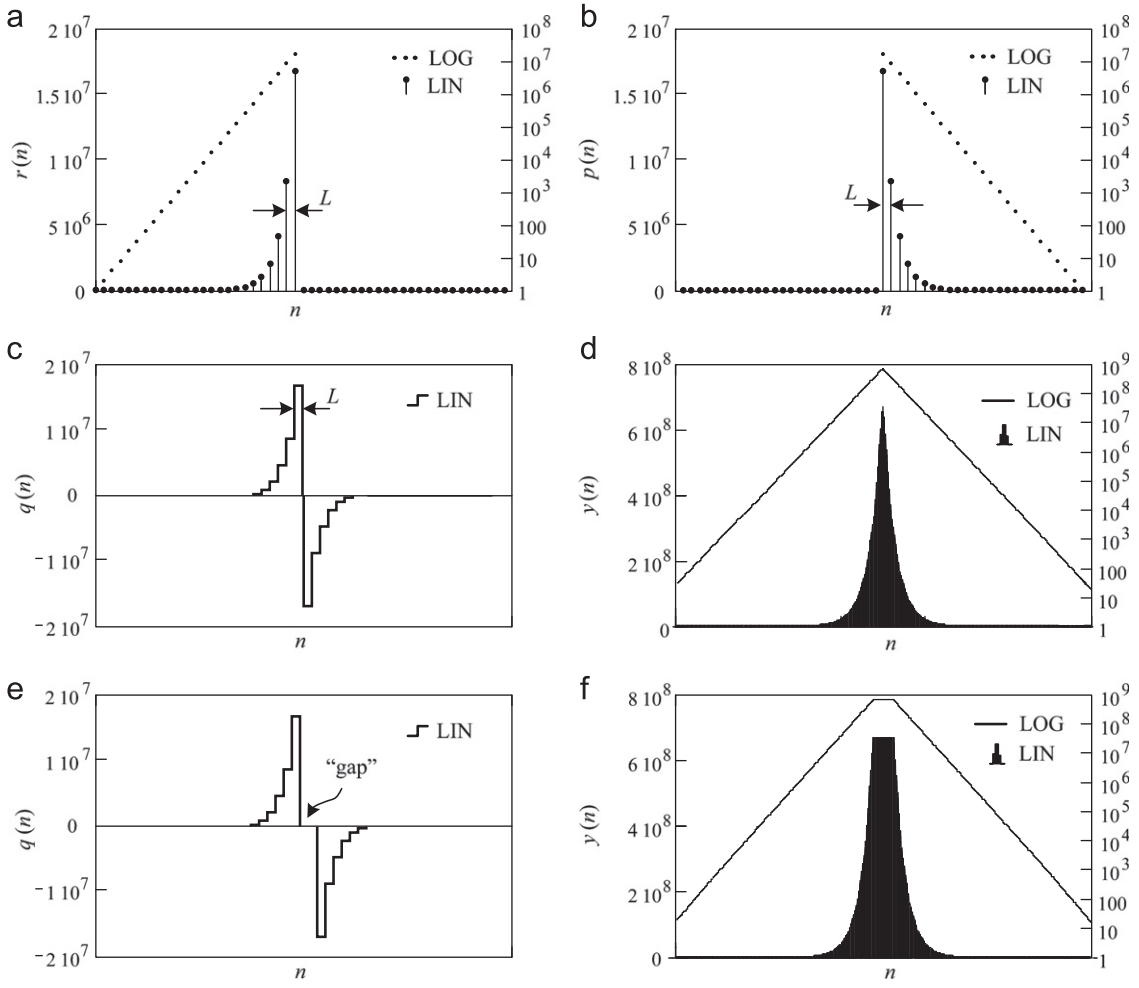

Fig. 9. FIR examples of linearly interpolated cusp shape synthesis with $K=24$, $L=40$ without a flat top (c, d) and with a flat top (e, f). Note that the vertical logarithmic scale is not aligned and scaled as the linear scale.

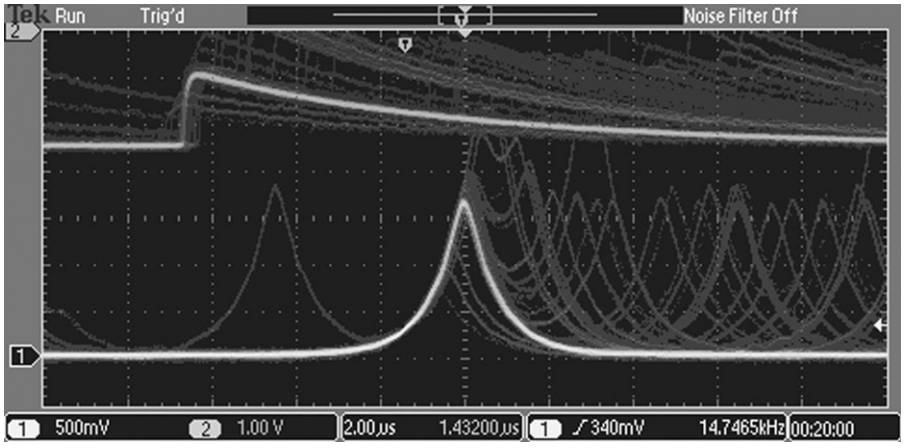

Fig. 10. Oscilloscope traces of real time cusp shape (trace 1) in response to a preamplifier tail pulse (trace 2).

When $L=1$ the system in Fig. 8 will synthesize true cusp shapes. Once the impulse response of the cusp pulse shaper is synthesized it is straightforward to synthesize cusp shapes from other digitized analog signals using the method of unfolding-synthesis as shown in Fig. 3.

Examples of cusp shape synthesis using the linear interpolation technique are shown in Fig. 9. The shapes are depicted in linear and logarithmic scales, Fig. 9a shows the exponentially growing part $r(n)$. Note the abrupt return to zero at the end of the exponential growth. Fig. 9b shows the exponentially decaying part. The non-zero samples of both $r(n)$ and $p(n)$ occur at intervals of L samples. Fig. 9c and d show the result of the first accumulation $q(n)$, which is a staircase like signal with step width of L samples. Fig. 9c represents a case of a cusp shape without a flat top ($m=0$). Fig. 9d shows a case of cusp shape with a flat top. Note that the positive (exponential growth) and the negative (exponential decay) parts of $q(n)$ are separated by a "gap" of samples with zero values. The length of this "gap" is equal to m samples, which is the duration of the flat top in the synthesized cusp shape. Fig. 9e and f depict the synthesized cusp shapes without and with a flat top, respectively.

A time invariant cusp shaper defined by Eqs. (21)–(24) was implemented in an FPGA. The signal from a silicon drift detector is digitized at 80 MHz by a 14-bit ADC. The digitized signal is continuously processed by the FPGA at the same rate as the ADC sampling rate. A digital-to-analog converter (DAC) is used to convert the digital signal to a physical signal (voltage) in the real time domain. Fig. 10 shows oscilloscope traces of the DAC reconstructed response of the cusp shaper (trace 1) to the analog tail signal (trace 2) from the silicon detector exposed to a Fe-55 source. To illustrate the real time operation the oscilloscope display persistence is turned on.

6. Conclusion

Efficient algorithms for synthesis of exponential pulse signals have been developed. These algorithms are suitable for implementation in either time-invariant or time-variant linear digital systems. The cusp shape algorithm based on linear interpolation is especially suitable for real-time implementation in hardware. It is also possible to implement higher order cusp interpolations. The discussion of such implementations, however, is beyond the scope and the size of this paper. All algorithms that use exponential growth functions require special attention to avoid signal distortion and numerical overflow due to round-off error amplification.

Acknowledgment

For more than twenty years I have created hundreds of digital pulse processing algorithms, the majority of which were never published. I would like to thank Dr. Mladen Bogovac from IAEA for his recent inquiry about cusp pulse shapers. This inquiry inspired me to look back, revise and compile the algorithms described in this paper.

References

[1] F.T. Arecchi, et al., Energia Nucleare 7 (10) (1960) 691.
[2] M. Konrad, Detector Pulse Shaping for High Resolution Spectroscopy, IEEE Transactions on Nuclear Science NS-15 (1) (1968) 268.
[3] V.T. Jordanov, et al., Nuclear Instruments and Methods A 353 (1994) 261.
[4] V.T. Jordanov, Nuclear Instruments and Methods A 505 (2003) 347.
[5] J.G. Proakis, D.K. Manolakis, Digital Signal Processing: Principles, Algorithms and Applications, third ed., Prentice Hall, 1995.
[6] V. Radeka, IEEE Transactions on Nuclear Science NS-19 (1) (1972) 412.
[7] E. Fairstein, IEEE Transactions on Nuclear Science NS-22 (1) (1975) 463.
[8] M.O. Deighton, IEEE Transactions on Nuclear Science NS-16 (5) (1969) 68.
[9] A. Pullia, E. Gatti, G. Ripamonti, IEEE Transactions on Nuclear Science NS-44 (1997) 331.
[10] W. Xiangyang, W. Yixiang, IEEE Transactions on Nuclear Science NS-53 (6) (2006) 3865.
[11] E. Gatti, M. Sampietro, P.F. Manfredi, Nuclear Instruments and Methods A 287 (1990) 513.
[12] P.W. Nicholson, Nuclear Electronics, J. Wiley and Sons, New York, 1974.
[13] E. Gatti, A. Geraci, G. Ripamonti, Nuclear Instruments and Methods A 395 (1997) 226.
[14] V.T. Jordanov, Nuclear Instruments and Methods A 351 (1994) 592.