New Algorithms for Improved Digital Pulse Arrival Timing With Sub-GSps ADCs

William K. Warburton, *Member, IEEE*, and Wolfgang Hennig

***Abstract*— The ability to measure pulse times of arrival with resolutions at or below 100 ps is becoming increasingly desirable in various fields, typically for signals originating from photon detectors such as photomultiplier tubes (PMTs) or silicon photomultipliers. Achieving the best results has typically required digitizing signals at rates between 4 and 20 giga-samples/second (GSps), followed by off-line processing, since such in-line processing technologies as digital signal processors and field-programmable gate arrays (FPGAs) cannot handle the data rates. For multichannel applications, the cost of GSps digitizers also becomes an issue. In this paper, we present two new methods for achieving similar time resolutions that are designed for FPGA in-line implementations operating with digitizers running at 250–500 mega-samples/second (MSps), approximately a factor of ten times slower. The first method uses a modified sinc function to interpolate the arriving pulse twice, once to precisely estimate its maximum M and once on its leading edge to locate the constant fraction point $f \cdot M$ as the pulse's arrival time. The second method takes the ratio of two points captured from a rapidly changing region of the pulse, either on its leading edge or near its peak, and uses this ratio with a preconstructed lookup table to generate the arrival time. We examine the algorithms' improved timing capability by comparing their coincidence timing resolutions to those from three standard algorithms, all applied to three data sets of pulses with different characteristics. At 500 MSps, the interpolation technique achieves about 7-ps full-width at half-maximum (FWHM) for an analog trigger pulse split between two ADC channels; for a fast laser pulse illuminating two ADIT L25D19 PMTs, better than 50-ps FWHM, and at 250 MSps, for two ϕ 25 mm × 25 mm $LaBr_3$ crystals exposed to ^{60}Co, 137-ps FWHM for gamma rays in a 0.95- and 1.33-MeV window.**

***Index Terms*— Digital signal processing, digital timing, field-programmable gate array (FPGA), pulse measurement, time-of-arrival estimation, timing, timing jitter.**

I. Introduction

THE problem of precisely measuring pulse arrival time arises in many fields and contexts. A few examples include: nuclear medicine, where photon–photon coincidence is the core component of positron emission tomography (PET) scanning and time of flight (TOF) can be further used to localize the point of emission; nuclear physics, where neutron TOF is used to determine their energies; light detection and ranging, where target range is found from photon round trip travel time [1]; and medical research, where fluorescence lifetime spectroscopy and imaging provide sensitive ways of detecting natural or tagged proteins [2]. Although these specific examples all involve pulses generated by a photon detector such as a photomultiplier tube (PMT) or silicon photomultiplier (SiPM), the methods we present are not so limited.

The structure of this paper is as follows. In Section II, we will provide a brief review of current analog (Section II-A) and digital (Section II-B) timing methods and consider the issues involved in going to larger, multichannel systems operating in real time (Section II-C). In Section III, we will analyze the constant fraction discrimination (CFD) timing method to understand in detail why it works better at higher sampling frequencies. In Section IV, we will present our first new timing algorithm, using interpolation. In Section V, we will present the second algorithm, using ratios of sample values. In Section VI, we will apply both these algorithms and three existing algorithms (CFD with linear interpolation, CFD with cubic spline interpolation, and weighted averaging [3]) to three data sets representing different types of timing pulses and compare achievable coincidence timing resolutions (CTRs). Finally, in Section VII, we will discuss our results and compare our $LaBr_3$ CTR results to a range of values found in the literature, primarily to benchmark the efficacy of our algorithms.

II. Existing Timing Methods

The history of timing methods is long, going back over 70 years. As a full review of their strengths and weaknesses is beyond the scope of this paper, we will restrict the following to a brief description of the leading methods, with a few references to papers with more extensive reviews for the interested reader.

A. Analog Timing Methods

Knoll [4] provides a good introduction to the issues involved in measuring pulse arrival times using analog techniques, and Moszyński and Mach [5] provide a discussion of the issues involved in achieving very high precision. The earliest approaches simply used a discriminator with a set threshold voltage, which worked (and still works) well for pulses whose shape and amplitude are relatively constant. However, when the pulse amplitude A is not approximately constant, then such leading edge discrimination (LED) timing introduces an amplitude dependent time walk. This issue was addressed by

Manuscript received July 25, 2017; revised September 18, 2017; accepted September 29, 2017. Date of publication October 24, 2017; date of current version December 14, 2017. Patent protection has been applied for on the methods described herein.

The authors are with XIA LLC, Hayward, CA 94544 USA (e-mail: bill@xia.com; whennig@xia.com).

Digital Object Identifier 10.1109/TNS.2017.2766074

CFD [6], which triggers at a constant fraction of A. Using analog electronics the *CFD(t)* signal is generated from the input signal $S(t)$ according to

$$\mathrm{CFD}(t) = fS(t) - S(t - \Delta) \tag{1}$$

where the fraction f lies between 0 and 1 and Δ is a time delay set approximately to the pulse's peaking height. The zero crossing of *CFD(t)* then defines the pulse arrival time. Further analog methods have been developed to cope with cases where pulse shape also varies, as is the case from large HPGe gamma-ray detectors, and these are described in [4]. However, since the vast majority of timing can be handled by the LED and CFD methods, we will not discuss them further here. Relatively recent reports of analog time resolutions obtained using $LaBr_3$ scintillators with both PMT and SiPM photo-detectors have been presented by Vedia *et al.* [7], Fraile *et al.* [8], Schaart *et al.* [9], and Seifert *et al.* [10], all showing the care required to obtain the very best CTR between two 511-keV gamma rays from ^{22}Na or between the 1.17- and 1.33-MeV gamma rays from ^{60}Co. Both cases are widely reported. We will reference the values they obtained in our discussion, Section VII-G.

B. Digital Timing Methods

The popularity of digital timing has been growing steadily in recent years due to benefits that include reduced drift, flexibility in analysis parameters and method without hardware changes [11], the ability to simultaneous perform particle identification through pulse shape analysis (PSA) [12], ease of interfacing with other experimental equipment and/or detectors [13], reduction in cost per channel in large channel count experiments [3], simple implementation of computer control over the experiment [14], reduction in preamplifier and shaping electronics [15], and the ability to preserve precise timing information between different filtering operations [14].

Given the flexibility inherent in digital processing, a large number of timing algorithms have been developed and reported. These range from the relatively simple, such as digital leading edge discrimination (DLED), digital CFD (DCFD), and weighted average [3] or moment analysis [11], to more complex methods including fitting of either average waveforms or analytic functions [3], or applying timing filters whose weighting functions are optimized for nonstationary noise [16]. However, given both our focus on algorithms capable on running in-line in multichannel systems and the observation that the more complex methods seem not to deliver significantly better results than the simple ones [3], [16], we will primarily consider only the DLED and DCFD algorithms.

DLED: The best reported time resolutions using digital leading edge discrimination commonly come from timing investigations of scintillators and photo-detectors intended for PET TOF applications ($\mathrm{PET_{TOF}}$). Typical recent studies include those by Seifert *et al.* [10], Ahmed *et al.* [17], Pro *et al.* [18], and Yeom *et al.* [19]. A typical implementation splits the preamplified output from each photo-detector, passing half to an "energy" digitizer input while further amplifying the other half for input to a "timing" digitizer. The energy side's gain is adjusted so that the largest energy pulses fit well within the digitizer's full scale range. The timing side gain is typically 4–5 times larger, making the leading edge signal large compared to both the digitizer's noise and its least significant bit in order to minimize digitization noise. Employed digitizers are either digital oscilloscopes (e.g., Agilent DSO90254A [19]) or discrete digitizers (e.g., Acqiris DC282 [10]) which can capture data at rates between 4 and 40 giga-samples/second (GSps) at 8–10 b. Event capture is triggered by an external coincidence unit fed by the two energy signals.

The actual timing calculations, however, do not operate in real time. The captured data are offloaded to a computer where the data are baseline corrected, energies determined from the amplitudes of the signals in the energy channels, and cuts applied to restrict further analysis to those events where both energies lie within a narrow energy band (such as the 511-keV peak). DLED is carried out only for these selected events and consists of locating the times where each trace crosses a preset threshold, which is adjusted to achieve best results. The actual crossing point normally lies between two digital samples, so the trace is locally fit using either a linear interpolation or a cubic spline to refine the time mark. The difference between the two detectors' time marks is then recorded and added to a histogram to finally determine time resolution.

DCFD: Data for DCFD are typically collected using setups similar to DLED, except that (again, typically) only a single trace is captured from each photo-detector [3], [20]–[23]. Then, in the data analysis, after energy selection has been applied, one of two different methods is used to extract timing marks from the traces. The most common method is to replicate the analog method, according to

$$\mathrm{DCFD}_i = fS_i - S_{i-\Delta} \tag{2}$$

which is the digital version of (1) with i indexing the digital samples and Δ (with occasional exceptions [24]) only being allowed to be a multiple of the digitizing interval. As per (1), f and Δ are adjusted for best results. The sought after zero crossing usually lies between two values of i, and so, just as in DLED, interpolation is used to refine its value. The second DCFD timing method, which we use at XIA LLC (XIA) and has also been reported in [11], [12], [25], and [26] is to implement the DCFD concept literally, using a digital delay line. That is, we make a delayed copy $S_{i+\Delta}$ of the data, where Δ is larger than the pulse's peaking time, search the original signal S_i for its maximum value A, and then seek the location in $S_{i+\Delta}$ where it crossed the value fA, using interpolation to refine the location of the timing mark. To our knowledge, this variant has also only been implemented offline. Further, whereas time resolutions reported using DCFD are nearly always poorer than those reported using DLED, the details of the reported experiments make it difficult to understand how much of this is due to the method itself and how much to experimental issues such as scintillator crystal size and number of digitization bits.

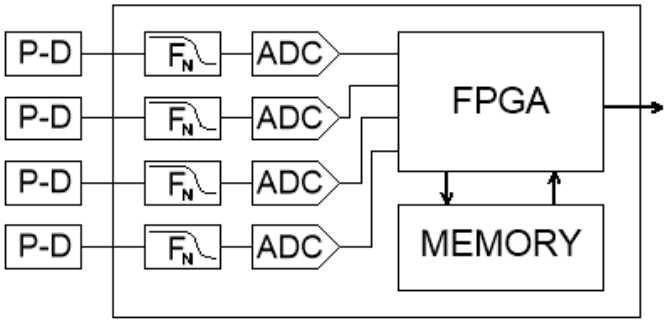

Fig. 1. Conceptual design of a digital multichannel timing system using minimal circuitry.

C. Practical Issues and Constraints

While the methods reviewed above are appropriate for advanced research, there are several issues that prevent their use in many applications, ranging from large multichannel detector arrays to small portable or hand-held devices, particularly those which require real time operation. First, there is the cost of digitizers running at multi-GSps rates. Second, there is the requirement for added circuitry to trigger data acquisition, and for DLED, the use of two digital channels per photo-detector.

For these applications, a more desirable goal might be a system whose design is sketched out in Fig. 1. Here, P-D is a photo-detector, F_N is a Nyquist filter, which may include amplification, ADC is an analog-to-digital converter, and FPGA is a field-programmable gate array. The output, to a local computer, is list mode data containing event energies and time marks, checked for both energy value and coincidence, as appropriate. Such a design could be employed in a neutron scattering experiment or a PET-TOF detector, for example.

In such a design, the use of the FPGA for real time data analysis effectively restricts the ADC clock speed to the maximum clock speed at which the FPGA fabric can run. Whereas it is true that FPGA interface logic can import data at much higher rates, e.g., serialized ADC data, the steps used to compute the fast pulse detection and slow energy extraction filters run more slowly. Thus, a typical value of a fast triangular filter FF_i is computed as

$$FF_i = S_i - S_{i-N} - S_{i-N-1} + S_{i-2N-1} \tag{3}$$

where the delayed terms are recovered from FIFO memory. By using a pipelined design, FF_i can be updated at the speed of the fabric clock. If new data samples S_i arrive faster than that, however, the filter cannot keep up. Given that modern FPGA clock speeds are approaching only 500 MHz, this sets a significant limit on allowed ADCs. On the other hand, because these ADCs can have 14 b of accuracy, as opposed to the 8–10 b found in very fast digitizers, the opportunities for fast timing with slower ADCs may not be as restricted as it at first might appear. Further, the data rates in this type of system are limited by pulse pileup, not be processing speed. Thus, for a $LaBr_3$ system with 50-ns wide pulses, the dead time from pileup inspection is 100 ns and the maximum throughput is or 3.7 MHz [4]. Finally, in cases where an external trigger can be used to trigger trace capture, then the FPGA need only process pulses fast enough to keep up with the experimental data rate and the above arguments do not apply. However, throughput rates may be much slower. Jäger and Butz [24] have considered event rate limits in one such case, with an estimated throughput of approximately 50 KHz.

III. Analysis of CFD Timing

Paulauskas *et al.* [3] investigated various issues involved with digital timing and, in particular, with DCFD timing of the type shown in (2). They made two important points. First, as also noted by Bardelli *et al.* [21] if the captured trace is not linear within the interpolation region, then simple linear interpolation will not produce accurate time marks. Second, they observe that the time extracted by linear interpolation varies with the phase of the sampled points with respect to the CFD point. Finally, they find that having more bits in the ADC leads to more accurate results for small pulses, showing the influence of digitization noise on timing accuracy.

We now examine the application of these concepts to DCFD of the second type, where we capture the pulse amplitude M and then look for the fM crossing in a delayed copy. Fig. 2 shows an example pulse digitized at approximately the Nyquist frequency (three points on the leading edge [27]) as well as 20 times faster. For the example pulse those frequencies are 100 mega-samples/second (MSps) and 2 GSps, but, for a faster pulse, the same arguments would similarly apply for frequencies of 500 MSps and 10 GSps.

The first point to notice is that, because of the sampling phase with respect to the pulse's arrival time, the maximum captured value M' does not equal the pulse's true maximum M. Thus, if the constant fraction threshold is set at fM', then a timing error will clearly result. Because the sampling phase will normally be random with respect to pulse arrival time, this leads to a distribution of timing errors. Note that, for the example pulse, M' can deviate from M by up to 5%. Therefore, to obtain accurate timing results, it is necessary to interpolate for M as well as for fM. As the pulse's peak region is strongly curved, some kind of higher order fitting function will be required. With the higher sampling rate, on the other hand, because the slope at M is zero, the maximum captured value M' in this case will be essentially equal to M.

The second point concerns the recovery of the time mark where the original pulse would have crossed the CFD value fM. As shown in Fig. 2, simple linear interpolation between points A and B is inadequate due to the pulse's local curvature. But what higher order fitting function should then be used? We experimented with using cubic splines to fit four points, two on each side of the crossing point (i.e., points A, B, and their two nearest neighbors) and found that, depending upon the phase of the sampling points, errors of up to 390 ps could easily occur (78 ps for 500-MSps sampling). This is not particularly surprising, since splines are curves bound by constraints on passage through specific points while maintaining continuity of various derivatives, as opposed to constraints related to the actual physical processes involved.

Timing algorithms using global fitting functions have the same weakness. If the selected function is not somehow related

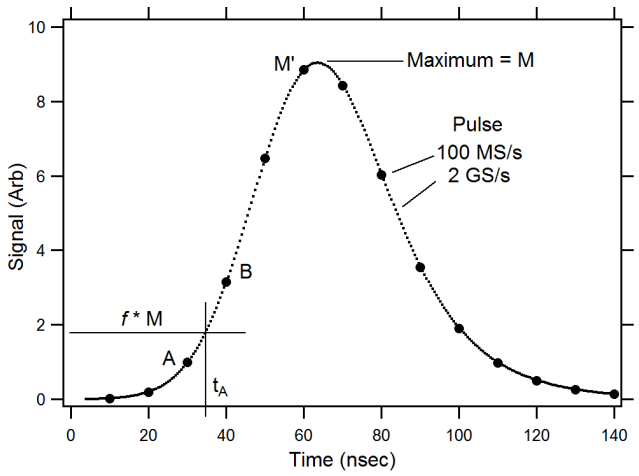

Fig. 2. Fast pulse digitized at two sampling frequencies, showing how the application of DCFD produces errors at lower sampling rates.

to the true shape of the sampled pulse, including per pulse variations, then there is no reason to expect interpolated values to accurately approach the original curve's values. As an additional practical constraint, since we are seeking algorithms that are readily implemented in an FPGA, we would prefer an interpolation algorithm that does not require very many operations and, in particular, requires few or no divisions.

Fig. 2 also clarifies the benefit of higher speed digitization. At the higher rate the density of sample points becomes sufficiently large so that the difference between the original curve and a linear interpolation between any two points becomes negligible. Our research question then becomes: can we find a sufficiently accurate interpolation method so that we can obtain comparable timing accuracy by replacing 10-b high rate data with interpolated 14-b data (8 times as accurate) that is approximately 10 times as slow (250–500 MSps versus 2–5 GSps)?

IV. Interpolated Timing

A. Introduction

As introduced earlier, our approach will require two interpolations: first of the peak and second of the constant fraction threshold crossing.

Generally speaking, there are two approaches to interpolation—fitting and convolution. Within fitting, we might consider both global fitting and local fitting. Global fitting has been tried by Paulauskas *et al.* [3], Nelson *et al.* [11], Haselman *et al.* [28], and others. The fit maximum M_F is then a supposedly better approximation to M and can also be used to interpolate the constant fraction value $f \cdot M_F$. The reported results, however, are not as good as either the pure analog results or the GSps digital results, probably because pulse fitting globally minimizes differences between the data and the fitting function, without particular emphasis on the peak, which is where we require the most accurate value. Further, nonlinear least-squares fitting is a computationally intensive operation and does not meet our requirement for implementation within a modestly sized FPGA [28]. Local peak fitting might provide superior results, provided that a simple function can be found that accurately captures the pulse's shape and has a small number of parameters so that it can be simply adjusted without using excess FPGA resources. However, considering that pulses from different sensors would probably have different pulse shapes and that this approach might entail finding a new fitting function for each experiment, we focused on convolution methods.

B. Interpolation by Convolution

The basis for interpolation by convolution lies in Fourier transform theory, where it is well known [29] that one can perfectly reconstruct the continuous function $g(x)$ from its sampled representation $g_s(x)$ via

$$g(x) = \mathrm{sinc}(x) * g_s(x) = \int_{-\infty}^{\infty} \mathrm{sinc}(\lambda) g_s(x-\lambda) d\lambda \quad (4)$$

where $\mathrm{sinc}(\lambda)$ is the ideal convolution kernel given by

$$\mathrm{sinc}(x) = \sin(x)/x \quad (5)$$

provided that the sampled function $g(x)$ has been properly bandwidth limited to satisfy the Nyquist criterion. However, it is also well known that this form cannot be used in practice because of the infinite convolution interval. It is further known that truncating $\mathrm{sinc}(\lambda)$, e.g., multiplying it by the *rect* function that is unity for $|x| < a$ and zero elsewhere), typically leads to oscillations in the interpolated function $g(x)$. As a result, researchers have developed a great variety of finite width interpolation kernels, either by truncating *sinc* more gently, using functions such as the Hann or Hamming, or using other functions entirely, such as the triangle filter or cubic approximation to the *sinc* function [29]. In this paper, expecting that non-*sinc* functions could distort the accuracy of the interpolated estimate of M, we elected to work with terminated *sinc* functions and explore methods to minimize the amount of computation involved.

As noted above, terminating *sinc* with a function of finite extent results in oscillations because the Fourier transform of a function of finite extent oscillates. However, it is also well known that the Fourier transform of a Gaussian is also a Gaussian, there being no oscillations because the Gaussian is nominally of infinite extent. However, the extent over which its value is actually significant (i.e., greater than 10^{-6}, for example) is only about 4.5 times its full-width half-max (FWHM) and this can be made small by choosing a small FWHM. Combining these two ideas, we developed the following convolution kernel, the discrete tapered sinc (*tsinc*) function defined at point i as :

$$\mathrm{tsinc}(i) = \mathrm{sinc}(i\pi/N)\exp(-(i/T)^2) \quad (6)$$

where N is the number of interpolated points per data point and T is a tapering constant, the Gaussian falling to zero so rapidly after a few multiples of T that larger values of i can be ignored without significant consequence. In a typical implementation, N is 8, T is 30 and i is computed for values between ± 48, which corresponds to L equals 6 lobes of the *sinc* function. That is, the number of points on each side of

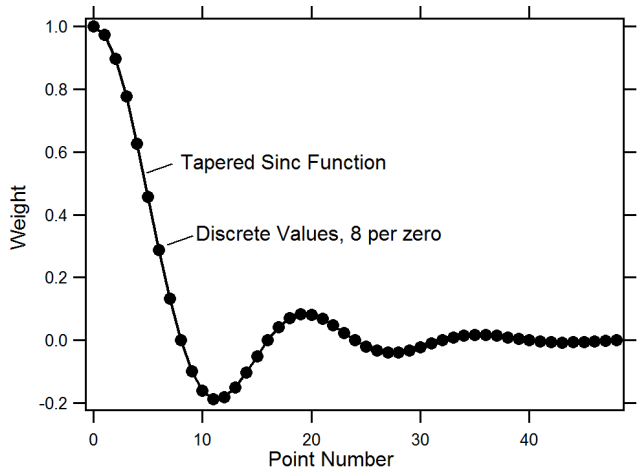

Fig. 3. Convolution kernel *tsinc* for $N = 8$, $T = 30$, and $L = 6$.

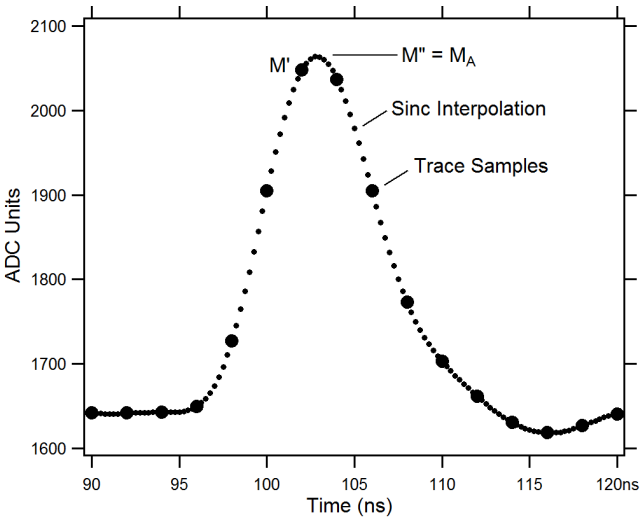

Fig. 4. Example fast pulse interpolated using *tsinc*, showing an improved ability to estimate the pulse's maximum value.

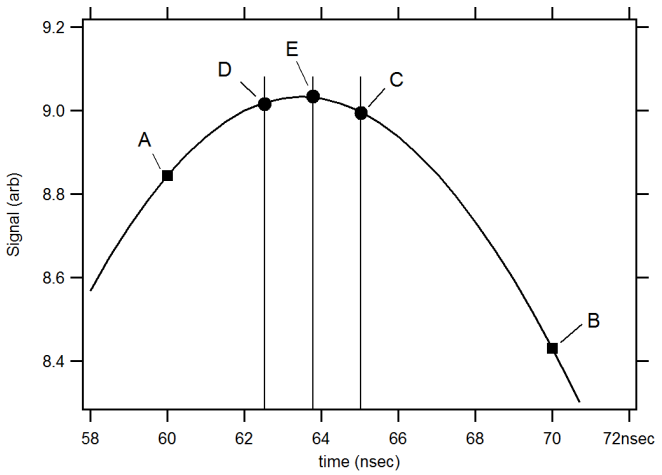

Fig. 5. Expanded view of the peak region of Fig. 4, showing the points found in a binary search for the pulse's maximum.

zero equals $L \times N$ although, because the function is even, we will only use positive i values in our operations. We note that this interpolation function, like *sinc* itself, preserves the values of the data points it interpolates between. As a concrete example, if N is 8 and our data are collected at 4-ns intervals (250-MSps sampling), then the interpolated points will be at 0.5-ns intervals (as if from 2-GSps sampling). Fig. 3 shows this example of *tsinc*, which can be seen to have little amplitude beyond i equal to 48. Both the continuous function and its discrete values are shown. Note that there are $N-1 = 7$ points between each of *tsinc's* zeros. Obviously, N can be adjusted for coarser or finer interpolation as desired.

Fig. 4 shows an actual fast trace that was digitized at 500 MSps from a fast timing PMT illuminated by a 60-ps laser pulse, as will be discussed further in Section VI. The tapered *sinc* interpolation is seen to be smooth between the data points, without any obvious oscillations near the peak, which was our goal. We notice that, because of the speed of this pulse, with a FWHM of only 7 ns, even 500 MS/s is not nearly fast enough to capture an accurate peak value M. The interpolation, however, at an effective sampling speed of 4 GS/s, nicely fills in the gap and provides an acceptably accurate estimate of M'' to work with, per the discussion regarding Fig. 2. Further, due to the Fourier transform properties of the underlying *sinc* function, we can have some confidence that the interpolated trace closely approximates the analog original and that we will thereby minimize or eliminate walk in our time mark estimates with varying sampling phase.

C. Finding the Pulse Maximum M''

Converting (4) into its discrete counterpart, the formula for an interpolated value $g(j,k)$, where j indexes the sampled data points and $1 \le k < N$ is the number of the interpolated point between j and $j+1$, is

$$g(j,k) = \sum_{i=0}^{L-1} (y(j-i)\text{tsinc}(iN+k) + y(j+1+i)\text{tsinc}((i+1)N-k)). \quad (7)$$

The *tsinc* values will be stored in a lookup table. Continuing with $L = 6$ (6 *tsinc* lobes), (7) shows that reconstructing each interpolated point $g(j,k)$ requires 12 multiply and accumulate operations (MACs). Therefore, making a full reconstruction of the trace, as per Fig. 4, requires thousands of MACs, which is unnecessary, since we actually only require the value of the highest point M'' and would like to find this point in as few computations of $g(j,k)$ as possible. Consider Fig. 5, which shows an expanded view of the peak region of the earlier pulse shown in Fig. 2. Continuing with $N = 8$, we see that there are only seven possible interpolated points between A and B to be tested. To find M'', we therefore implement a binary search routine for M'' that computes an interpolated point halfway between the current two largest points and repeats $K = ln_2(N)$ times. In this example, we therefore compute $g(A,x)$ for three values of x, here 4, 2, and 3, to produce the points C, D, and E, giving $g(A,3) = E = M''$ as the maximum we seek. This process is very efficient, using only 36 MACs to locate M''. If we were to double the density of interpolated points, to $N = 16$, only a single additional search point would be required.

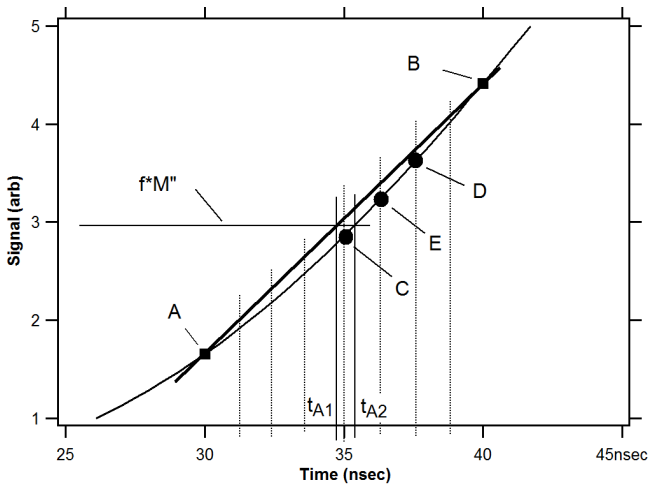

Fig. 6. Expanded view of the Fig. 2 pulse's leading edge, showing the points found in a search for the constant fraction point closest to the value $f * M''$.

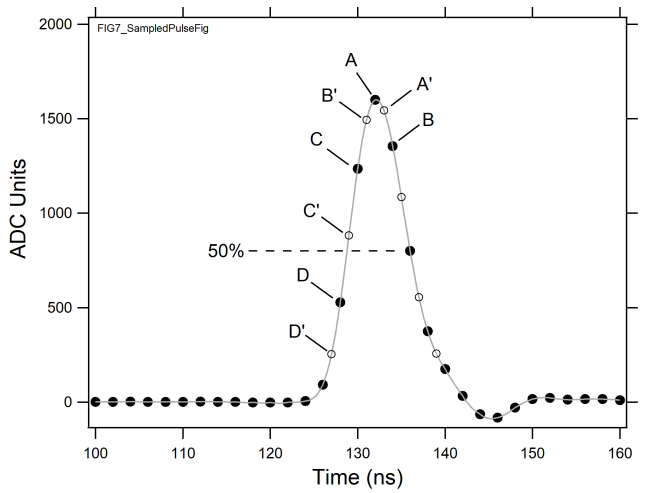

Fig. 7. Digitized pulse showing changes in sample values with changes in phase between pulse and sampling clock.

D. Finding the Pulse's Time Mark

Fig. 6 shows an expanded region of the Fig. 2 pulse where it crosses the constant fraction value $f \cdot M''$. Here, the sampled points A and B bracket the value $f \cdot M''$, so that, if we define the mark (arrival) time t_A by

$$t_A = t_L + \varphi \tag{8}$$

where t_L is the point A's clock time, we need to estimate the arrival time offset φ, the pulse's phase, to find the time mark t_A. Whereas a linear interpolation between points A and B, separated by 2 ns, would yield time mark t_{A1}, we will again use *tsinc* interpolation to produce two new points A' and B' that bracket $f \cdot M''$ but are only separated by 250 ps and then linearly interpolate between them to find a more accurate time mark estimate t_{A2}.

As in finding M'', we again employ a binary search routine to locate the bracketing points A' and B', but this time comparing each computed value of $g(A, x)$ to $f \cdot M''$ to determine which point to compute next. Referring to Fig. 6, we first compute $g(A, 4)$ (point C) halfway between A and B. Then, because it is less than $f \cdot M''$, next compute $g(A, 6)$ (point D). Because this value exceeds $f \cdot M''$, we finally compute $g(A, 5)$ (point E) to complete the search, with points C and E being A' and B', respectively. This process always takes exactly three computations of $g(A, x)$ for 36 MACs.

Once A' and B' are located, a linear interpolation is used to find t_{A2}. The time mark is then found from

$$t_{A2} = t_A + (x_{A'} + (f \cdot M'' - A')/(B' - A'))\Delta/N \tag{9}$$

where Δ is the time between ADC samples and N is the number of interpolated points per ADC sample. In the example of Fig. 6, $A' = g(A, 4)$ and $B' = g(A, 5)$. The found time difference between points t_{A1} and t_{A2} is about 500 ps, which underscores the importance of using an accurate interpolation procedure.

V. Ratio Timing

As with the previous algorithm, our goal is to develop a method that achieves very good timing resolution using lower speed ADC data (500 MSps and below) while being implementable using the resources available in a typical FPGA. The principle of the method may be understood by reference to Fig. 7, where an analog pulse (gray solid line) has been digitized twice, once at 2 ns/point (500 MSps) with some arbitrary phase relationship between the pulse and the digitizer clock (solid dots) and once where the pulse arrived 1-ns later (open circles). The 1-ns phase shift is clearly reflected in the locations of the samples on the curve. In the first case, for example, we have captured a point A that is very close to the pulse's peak, while the next largest value B is over 10% lower. In the second case, the first and second largest values, A' and B' are nearly equal in value, with the peak location lying between them. By calibrating this behavior it becomes possible to extract the pulse's phase with respect to one of the digital samples and so obtain the pulse's absolute arrival time.

We develop the calibration procedure under the following assumptions. First, that except for amplitude, the pulses' shapes are invariant. Second, that each digitized pulse has a time stamp associated with one of its samples. Third, that any dc offset has been removed, giving a zero baseline. Fourth, and critically, that we have some accurate independent means for determining the pulse's arrival, at least for enough pulses to construct a calibration curve.

We then proceed as follows: given a data set of 10 000 captured traces, we analyze each trace twice. First, we use the algorithm described in Section IV to find its arrival time. Then, we locate the two points in a region where the pulse is changing rapidly. These might be on the pulse's leading edge, for example, the two points that bracket the pulse's 50% height, or in the vicinity of its peak. The preferred method will depend upon the particular nature of the pulses, as we will discuss in more detail later.

A. Using a Ratio of Peak Values

In this section, for timing traces captured from a timing PMT illuminated with a fast pulse of laser light, we employ the case where we select the pulse's two highest points. According to Fig. 7 that would be either A and B or A' and B', depending upon the sampling phase. We then take their ratio B/A and

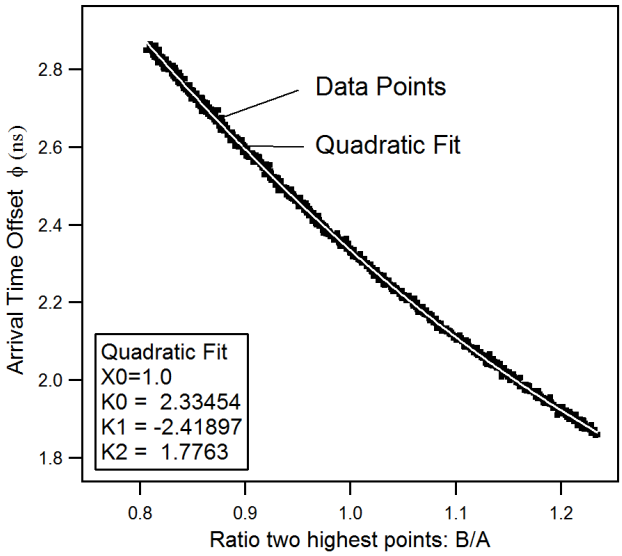

Fig. 8. Calibration curve for PMT A + laser pulse, showing pulse arrival times versus B/A ratio values and a fit thereto.

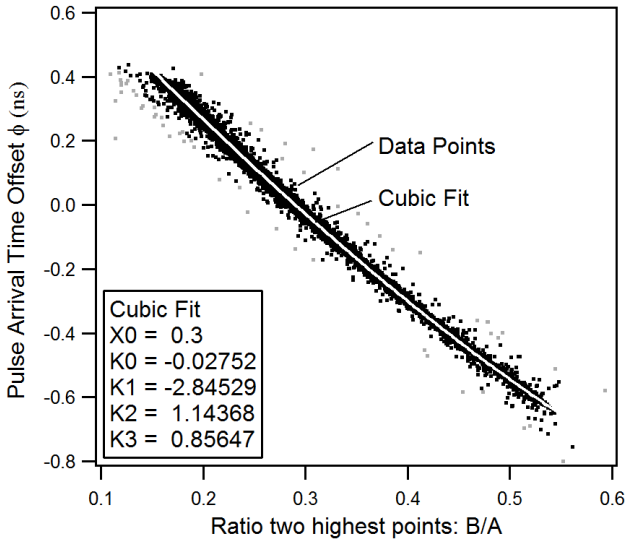

Fig. 9. Calibration curve for PMT A + $LaBr_3$, showing pulse arrival times versus C/D ratios and a fit thereto.

plot it versus the pulse's arrival time measured from the highest point A. The results from all the traces are then plotted and a curve fit to the data, as shown in Fig. 8. As may be seen, the data are fit well using a simple quadratic, with little scatter. Given that the plot's time scale is 200 ps/division, the vertical scatter suggests that the resulting time resolution will be very good.

Having produced a calibration curve for each PMT in one's timing setup, the collected data are analyzed as follows. For each trace, the points A and B are located, the ratio B/A formed, and its arrival time offset from point A computed using the coefficients of the fit to its associated calibration curve:

$$\Delta t_A = K_0 + K_1 R + K_2 R^2 \quad \text{where } R = B/A - X_0. \tag{10}$$

Then, given the easily computed offset t_{S-A} between the pulse's time stamp and point A, the pulse's arrival time t_A is

$$t_A = t_{ts} + t_{S-A} + \Delta t_A. \tag{11}$$

TOF or coincidence time differences are then found by taking differences between the arrival times of appropriate pairs of pulses.

B. Using a Ratio of Leading Edge Values

In this section, we process traces captured from the same timing PMT as in Section A, but now connected to a $LaBr_3$ crystal excited by a ^{60}Co source. For such pulses, we have found it better to employ the analysis where we select two points on the traces' leading edges that bracket some constant fraction of the pulses' heights. Our explanation of the difference between the two examples is that pulses from scintillators show more variation in their peak shapes due to photon emission statistics.

As shown in Fig. 7, for a 50% fraction, the selected points would be either C and D or C' and D', depending upon the sampling phase. We then take their ratio C/D and plot it versus the pulses arrival time, now measured from the earlier point C. The results from all the traces are then plotted and a curve fit to the data, as shown in Fig. 9. In this case, the data require a cubit fit and also show significantly more vertical scatter, from which we project that we will obtain poorer timing resolution than in the laser illuminated case. Data processing proceeds as before, using equations similar to (10) and (11).

In the preceding description of the ratio timing method we used the algorithm described in Section IV as the independent timing means to generate a calibration curve. Other techniques can be employed as well. For example, one could capture a few thousand pulses using a fast digitizer, determine their arrival times using one of the leading edge timing methods found in the literature, and also compute their ratios to generate the calibration curve. Alternately, one could capture a single trace at high sampling rate (i.e., 20 GSps or 50 ps/point) and decimate it 40 times at 2 ns/point to generate 40 traces of 500 MSps data, each with a unique sampling phase offset from the next by 50 ps. These decimated traces could then be used as described above to produce a calibration curve. Multiple 20-GSps traces could be processed to improve fitting statistics.

VI. Comparisons Between Methods

In this section, we will compare CTRs obtained by five digital algorithms—classical CFD with either linear interpolation or cubic spline interpolation, weighted averaging as described by Paulauskas *et al.* [3] interpolated timing per Section IV, and ratio timing per Section V—when they analyze coincident pulse pairs captured from three timing setups whose pulses have significantly different types of statistical fluctuations. These are: 1) a fast analog trigger pulse; 2) a fast laser pulse simultaneously illuminating a pair of fast timing PMTs; and 3) the same PMTs coupled to $LaBr_3$ crystals excited by coincident gamma rays from ^{60}Co decay. The three setups are shown schematically in Fig. 10.

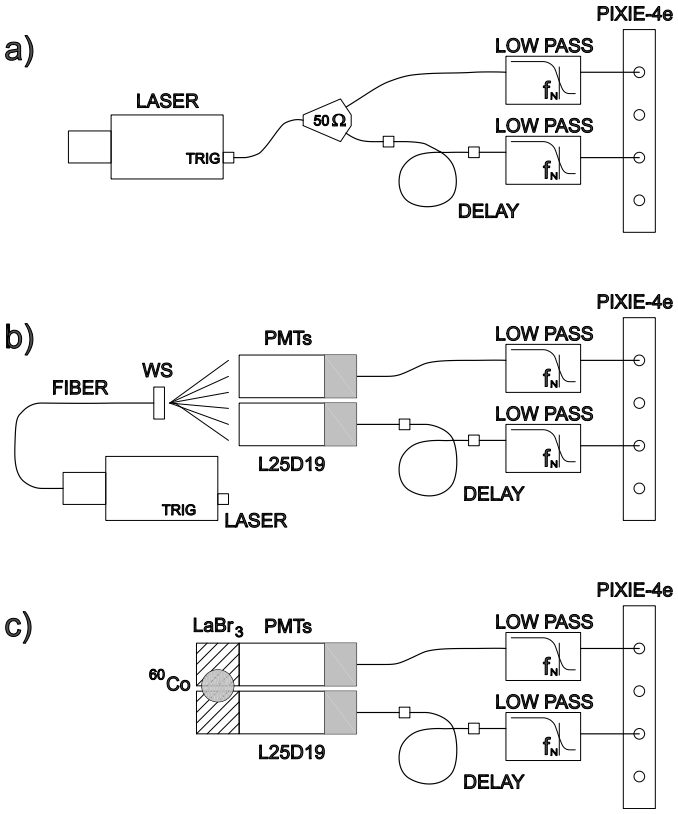

Fig. 10. Schematic drawings of the three experimental setups used for data collection. (a) Split fast electrical pulses. (b) Simultaneous fast optical pulses. (c) Coincident scintillator pulses.

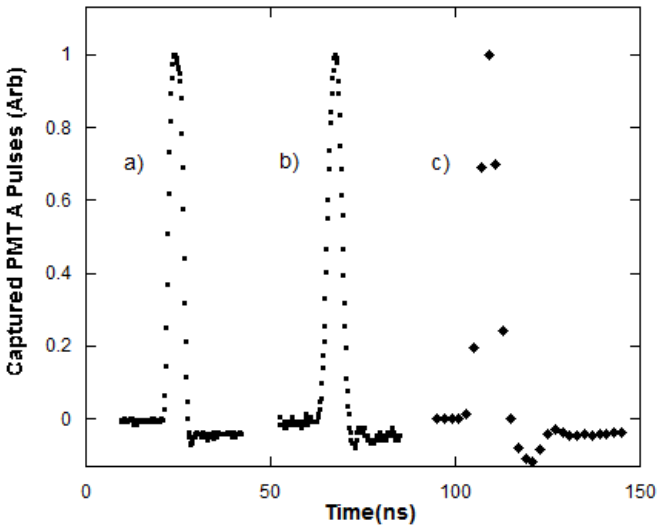

Fig. 11. Typical fast analog pulse as (a) generated, (b) filtered, and (c) captured at 500 MSps.

A. Experimental Procedures

1) Fast Analog Pulses—Fig. 10(a): For fast analog pulses, we used the trigger output of our Edinburgh Instruments EPL-405 diode laser operating at 20-K pulses/s. As generated, it is a logic pulse of 6.4-ns duration, 0.5-V amplitude, and 1.5-ns 10–90 rise and fall times. This signal is connected to a 50-Ω splitter and thence to a pair of custom built low pass filter boxes that have 477-MHz amplifiers (AD8000) at gain 2X, followed by a selection of passive seven element Bessel filters with 3 dB points of 31, 45, 54, 63, 88, or 125 MHz. Unless otherwise noted, the filter boxes were then connected to an XIA Pixie-500e, which captured pulses in coincidence using a 500 MS/s, 14-b ADC and stored them to disk for later analysis in an IGOR Pro [30] program that implemented the five timing algorithms. The 50-Ω cables were used throughout, with 50-Ω terminations at all interfaces to minimize any reflections that might distort the pulses. Typically between 40- and 80-K pulses were captured for each set of parameters investigated.

The selection of Bessel filters was included for three reasons. First, commercial digitizers may or may not include Nyquist filtering, depending upon their intended application. The Pixie 500e modules do, but they are only second-order Sallen-Key filters intended to suppress aliasing. As such, their pulse response to a delta function input is not symmetric, producing a much faster leading than trailing edge that does not have sufficient points [27] for accurate interpolation. Further, since lowering the cutoff frequency reduces high frequency noise, we wanted to determine if there were any timing benefits to be gained by intentionally operating with sub-Nyquist filtering.

Critically, a cable delay was inserted in one leg, as shown in Fig. 10(a), that could be adjusted in 0.5-ns steps between 0 and 1.5 ns. In all cases, all four delays were tested and the worst timing resolution is the one reported since it is the most accurate measure of the technique. This is because, as follows from the work of Paulauskas *et al.* [3] if the sampling phase is close to or equal in both legs, particularly for identical pulses, then any time estimation error in one leg is precisely replicated in the other leg and the difference cancels. Thus, the measured CTR is indicative only of issues like signal noise and time jitter in the ADC sampling stages and is not representative of the processing algorithm. Only when sampling occurs out of phase between the two pulses can the algorithm's intrinsic deficiencies be observed.

Data for this fast analog pulse demonstration were collected using the 125-MHz filter, with the gain adjusted so that the pulses were approximately 300 mV high, spanning about 15% of the ADCs input range. Fig. 11 shows the generated [Fig. 11(a)] and filtered trigger pulses [Fig. 11(b)], as captured at 10 GSps by a Tektronix DPO7254 digital oscilloscope, and the pulse [Fig. 11(c)] as captured for analysis at 500 MSps by the Pixie-500e. The additional pulse broadening is due to the Pixie-500e's internal Nyquist filtering. We observe that the captured pulse has 3–4 samples on its rising edge, which shows that we have satisfied the Nyquist criterion [27]. Further, as we shall see, slowing the pulse's rise time from 1.5 ns to over 6 ns by the two stages of Nyquist filtering does not appear to interfere with our ability to obtain excellent CTR values.

2) Fast Optical Pulses—Fig. 10(b): In this setup, we used the same Pixie-500e, low-pass filters and cables, including the delay, as in VI.A.1, but attached them to a pair of Electron Tubes L25D19 fast timing PMTs [31]. This is an eight-dynode, linearly focused PMT with a blue–green sensitive bialkali

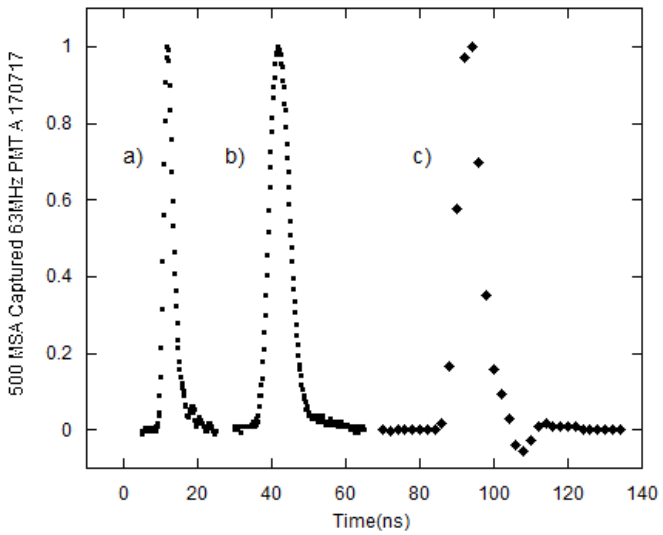

Fig. 12. Typical fast optical pulses as (a) generated, (b) filtered, and (c) captured at 500 MSps.

photocathode whose response peaks at 350 nm. Its timing specifications are: multielectron rise time = 1.0 ns, single electron jitter = 290 ps FWHM, and transit time = 12 ns. The tubes were placed side-by-side in a dark box and illuminated by light pulses from the EPL-405 laser whose duration is specified to be less than 60 ps. The light is transmitted into the dark box by fiber optic and then scattered from a thin sheet of white filter plastic (Lee Filters No. 216) [WS in Fig. 10(b)] to more uniformly illuminate the two tubes' faces. Neutral density gray filters (Lee Filters No. 209) inserted between the laser and fiber optic provided attenuation as required. The PMTs were operated at 1200 V and the filter gains were adjusted so that the pulses were typically 120 mV high, spanning only 6% of the ADCs input range. By measuring the standard deviation in the output pulse heights, we estimate that typical pulses contained 700–800 photoelectrons. Hoping to improve signal to noise with respect to electronic noise sources later in the amplification and digitization chain, we tried increasing the pulse's amplitudes by increasing the PMT voltage to 1400 V, but found that this actually degraded achievable CTR. Our selected operating point was in a region where PMT pulse amplitude was linear in number of photoelectrons at 1200 V, but showing signs of nonlinearity at 1400 V. The issue may be related to the use of the laser source. With only a 60-ps pulsewidth, instantaneous charge densities within the PMT can easily be 100's of time larger than would be observed from even a fast scintillator pulse producing the same number of photoelectrons. In any case, fully exploring and understanding the subtleties of the PMTs timing behavior will require further exploration that is beyond the scope of this paper. The Bessel filter cutoff frequency that produced the best CTR was found to be 63 MHz.

Fig. 12 repeats Fig. 11, showing the pulses (a) before and (b) after filtering (captured at 10 GSps) and (c) as digitized at 500 MSps. Comparing Figs.11 and 12, the analog and optical pulses appear to be fairly similar, although the latter has approximately one additional point on their rising edges, reflecting the lower Bessel filter bandpass.

3) Fast $LaBr_3$ Scintillator Pulses—Fig. 10(c): In this setup, we attached a $\phi 1'' \times 1''$ $LaBr_3$ crystal to each of the PMTs using silicone coupling compound (Hilger Crystals). The $LaBr_3$ crystals were used as provided by Saint Gobain, each Teflon wrapped in an aluminum can with one glass window. These are the same crystals reported in [22]. The PMTs and scintillators were placed side-by-side in the dark box and a few μCi ^{60}Co source set on top of the $LaBr_3$ crystals as shown in Fig. 10(c). The PMTs were operated at 1200 V with 31-MHz Bessel filters, whose gain was set so that the 1.17- and 1.33-MeV gamma rays produced 710- and 805-mV pulses, spanning 40% of the ADCs input range. These pulses were captured using an XIA Pixie-16, which has 14-b 250-MSps ADCs. These pulses typically have four leading edge samples (12 ns) due to the 31-MHz filter, and decay back to baseline in 12 samples (50 ns), which reflects $LaBr_3$'s approximately 16-ns decay time. Estimating a 90% photon collection efficiency and a 26% PMT conversion efficiency, our pulses would contain of order 16-17 000 photoelectrons per MeV.

We note that, in this geometry, single photons backscattering between the crystals can also produce "coincidence" events that can either precede or follow true prompt gamma–gamma coincidences by up to 100 ps, the transit time of the scattered phonon across the 3 cm between the two $LaBr_3$ crystals. The deposited backscattering energy is remarkably insensitive to either the gamma ray's initial energy or the exact backscattering angle, producing a spectral peak at 240 keV (145 mV peak). These events can therefore be eliminated by setting a threshold above about 200 mV. Since any events that escape this cut can only degrade our time resolution results, they should therefore be considered only as upper bounds on what is achievable.

B. Analyses

Five analysis algorithms were applied to all the data sets, as follows. In all cases, we adjusted the methods' parameters, where possible, to obtain the best result. In particular, the constant fraction values used were varied from 10% to 90% to locate the optimum value. All methods operated on baseline subtracted data, find the arrival times of the two coincident pulses independently, record the difference, and then histogram these differences to obtain the reported CTR. All reported CTRs are FWHM.

1) DCFD, Linear Interpolation ("CFD"): As discussed above, in this method, we set the CFD value as a fraction of the *captured* peak value and linearly interpolate between the two points bounding this value to obtain the pulse's arrival time.

2) DCFD, Cubic Spline Interpolation ("Spline"): Since many literature studies use cubic spline interpolation, we included this method. We interpolated each pulse between the start of its rising edge and halfway down its falling edge using a built-in IGOR function. We then set the CFD as a percentage of the maximum interpolated function and linearly interpolated between the two closest spline values to obtain the pulse's arrival time.

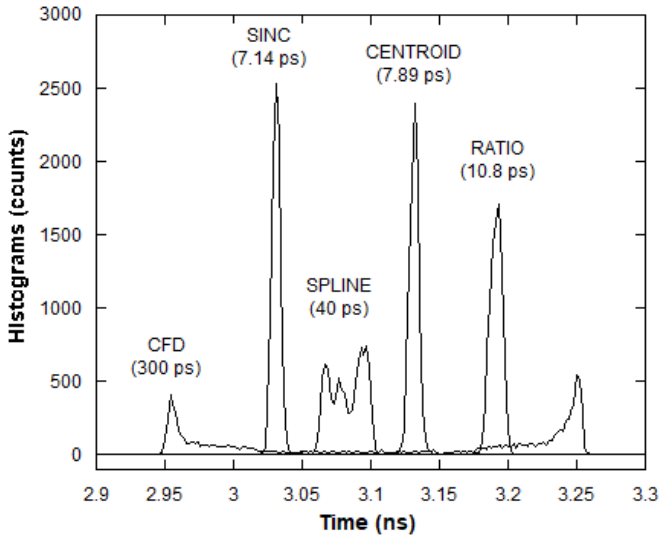

Fig. 13. Timing histograms produced by the five algorithms processing laser trigger pulses. Their FWHM time resolutions are indicated.

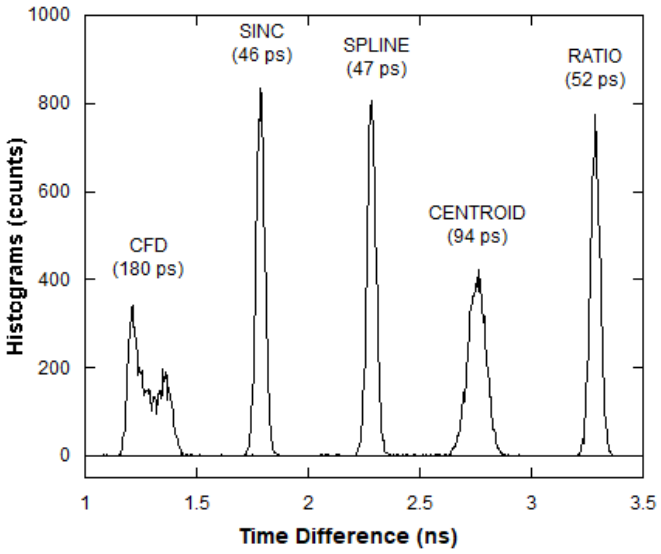

Fig. 14. Timing histograms produced by the five algorithms processing fast optical pulses. Their FWHM time resolutions are indicated.

3) Centroid ("Centroid"): Here, the pulse's arrival time is its centroid, computed from a common clock time, using the usual formula [3].

4) DCFD, tsinc Interpolation ("Sinc"): This method, described in Section IV, finds the pulse maximum by *tsinc* interpolation between the two maximum captured points, takes the CFD as a percentage of this value, and then uses *tsinc* interpolation a second time between the two nearest data points to find the pulse's arrival time.

5) Ratio Timing ("Ratio"): In this method, the ratio of two data points is taken and then a lookup table is used to determine the pulse's arrival time, as described in Section V. For the analog and optical pulses we used the two maximum points. For the $LaBr_3$ data, we used the two points surrounding a "CFD" value on the pulse's leading edge, per Section V. In all cases the calibration curve was generated using *tsinc* interpolation.

C. Experimental Results

1) Fast Analog Pulses: Our CTR results from analyzing the fast analog laser trigger pulses are presented in Fig. 13. The best FWHM time resolution, 7.14 ps, is found using *Sinc*, followed by *Centroid* (7.89 ps), *Ratio* (10.8 ps), *Spline* (40 ps) and *CFD* (300 ps). The *Sinc* value is impressive, considering that data samples were taken at 2-ns intervals on a pulse with a 6-ns rising edge. We reiterate that these are our *worst* values with regard to inter-pulse phase variations. The *Sinc* method was relatively immune to phase, as hoped, varying only by about 1 ps between the best and worst values. We will further discuss the behavior of the five methods in Section VII.

2) Fast Optical Pulses: CTR results from analyzing the fast optical pulses are presented in Fig. 14. All pulses were included in the analysis. The best FWHM time resolution, 46 ps, is again found using *Sinc*, followed by *Spline* (47 ps), *Ratio* (52 ps), *Centroid* (94 ps), and *CFD* (180 ps). Achievable CTR has degraded by a factor of about 7 compared to the fast analog case.

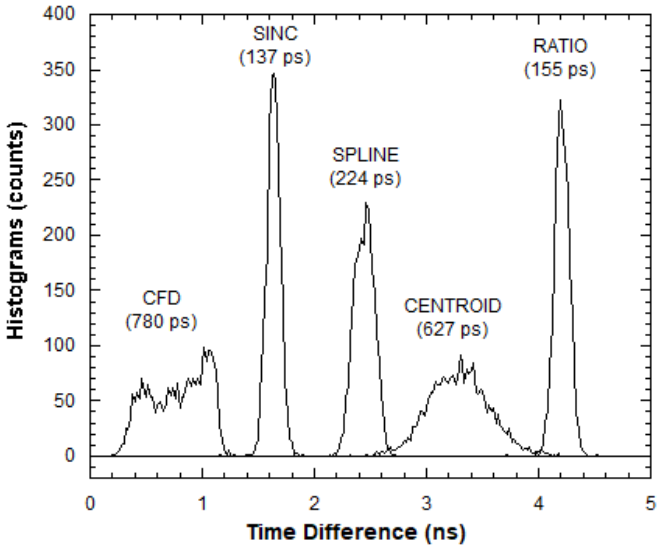

Fig. 15. Timing histograms produced by the five algorithms processing fast $LaBr_3$ pulses. Their FWHM time resolutions are indicated.

3) Fast $LaBr_3$ Scintillator Pulses: Our CTR results from analyzing fast $LaBr_3$ scintillator pulses are presented in Fig. 15. In order to span an energy range from 0.95 to 1.40 MeV, only the largest 30% of the pulses were included in the analysis. *Sinc* still provides the best FWHM time resolution (137 ps), followed by *Ratio* (155 ps), *Spline* (224 ps), *Centroid* (627 ps), and *CFD* (780 ps). Achievable CTR has degraded by another factor of three, compared to the fast optical pulse case.

VII. Discussion

A. Comparison Between the Pulses

Although it has the pulses of least duration, the fast analog pulses provide perhaps the easiest timing case for two reasons.

First, all of the pulses are essentially identical in height, so that any amplitude dependent processes in either the electronics or the algorithms are minimized. Second, the pulse shapes are highly reproducible, without any pulse-to-pulse variations arising from photon statistics, electron transit time jitter or the like. In this case, it does not matter if an algorithm like the *Spline* cannot reproduce the pulse shape precisely, so long as it does so reproducibly.

The optical pulses add two more statistical processes—the number of photoelectrons generated and their transit time variations through the PMTs. The former process is fast, within the 60-ps laser pulse, while the second is set by the PMTs timing characteristics. Both processes can lead to pulse shape variations at the sub-100 ps level, so interpolation accuracy becomes more important, although the slower Bessel filter smears out some of this.

The $LaBr_3$ pulses add a final source of fluctuation due to photon emission statistics within the scintillator, which can lead to strong variations in pulse shape depending upon exactly when the photons are emitted. This is the reason why scintillators with both fast rise and fall times are valued for timing applications since both processes concentrate photon emission into the pulse's leading edge, which then leads to less pulse-to-pulse variation. This is an important consideration. As we see from the fast analog case, we do not necessarily need a super-fast rising edge to obtain good timing resolution, provided we can obtain a super *consistent* leading edge shape, pulse-to-pulse. From this viewpoint, the more the light emission looks like a delta function the more its shape will be reproducible after Nyquist filtering.

Given these observations, we can now interpret our results on the three sets of data.

B. CFD Algorithm

This algorithm has a very difficult time when the sample points are so sparse that they do not track the underlying pulse shape well. As discussed earlier, *CFD* does not obtain either an accurate peak value or an interpolated CFD value. This is starkly revealed in the fast analog case. With a zero nanosecond delay between the two branches, the *CFD* algorithm almost matches the *Spline* time resolution, for reasons discussed earlier. But once the phases of the two pulses are no longer synchronized, the full range of its estimation errors is revealed. It does a bit better in the fast optical case because of the lower Bessel filter bandwidth but is again thrown off by photon statistics in the $LaBr_3$ data set.

C. Spline Algorithm

The *Spline* algorithm does poorly in the fast analog case because the mathematical shape produced by the restrictions of curve continuity is different from the actual pulse shape and so timing estimates vary with sampling phase. When the pulses have a lower bandwidth, compared to their internal fluctuations, as in the fast optical case it does quite well, but then degrades again when the pulse shapes vary due to photon generating statistics in the $LaBr_3$ data set.

D. Centroid Algorithm

The *Centroid* algorithm works well only when the pulse shape is extremely reproducible. As reproducibility degrades, in the fast optical and more so in the $LaBr_3$ cases, its performance degrades quite rapidly.

E. Sinc Algorithm

As hoped, with its ability to recover the pulse's underlying shapes when the Nyquist criterion is satisfied, the *Sinc* interpolation method works well in all three cases. While it cannot provide a correction for fluctuations in photon arrival times, as has been proposed [32], it achieves excellent results without the need for ultrahigh speed digitizers and is capable of in-line data processing. In particular, these results are achieved from a single ADC trace capture for each pulse and, as presented, are only using less that 25% of the ADC range. Even better performance should be achieved by better matching of the pulse amplitudes to the ADCs input ranges.

F. Ratio Algorithm

Like the *Sinc* algorithm used to generate its calibration table, the *Ratio* algorithm also achieves good results for all three data sets. In the shown worst results, it always obtains slightly worse performance than *Sinc*. However, in many of the data sets we processed it essentially matched *Sinc* and, on occasion, surpassed it. With its simpler in-line processing requirements, *Ratio* may be a good compromise in applications with large numbers of channels.

G. Comparison to Literature Values

As is well known in the field, it is very difficult to make direct comparisons between CTR values obtained from different experiments, since they depend upon so many experimental variables including scintillator size, shape and packaging, PMT model and operating voltage, the electronics chain and digitizer specifications including rate, number of bits, and both differential and integral nonlinearities, and the gamma-ray energies measured. Our three reference data sets were therefore collected primarily to provide a means to directly compare the performance of our new algorithms to literature standards. In particular, for our $LaBr_3$ measurements we simply took a pair of available crystals, attached them to a pair of available (but good) timing PMTs, set an available source on them, and collected a single data set for each of several Bessel filter bandwidth values using one of XIAs popular digital spectrometers. No optimization attempts were made.

These provisos aside, our results appear to be competitive with some of the best literature values. Table I provides a sampling of literature values, together with a few of the more important experimental parameters. Where the authors report a time resolution for a single detector, we have multiplied by sqrt(2) to estimate CTR values. As the table shows, the majority of reported measurements have been made at 511 keV, reflecting PET as a driving force in developing improved timing methods. The replacement of PMTs with SiPM detectors is also significant, since it removes transit

TABLE I
REPORTED VALUES FOR CTR VALUES USING $LaBr_3$, ORDERED BY YEAR

1st Author	Year	Size (mm)	Detector	Method	Details	Eγ (keV)	Δt (ps fwhm)
Moszyński [33]	2006	ϕ 25 x 25	PMT XP20D0	Analog	Ortec 935/456	1,330	151
Moszyński	2006	ϕ 25 x 25	PMT XP20D0	Analog	Ortec 935/456	511	218
Fallu-Labruyere [22]	2007	ϕ 25 x 25	PMT XP2020	DCFD (0 cross)	75 MSps 14 bits	511	576
Seifert [37]	2009	3 x 3 x 5	SiPM	DLED	8 GSps, 10 bit	511	100
Joly [16]	2010	ϕ 13 x 13	PMT XP20D0	DLED (23%)	5 GSps 8 bits	511	237
Schaart [9]	2010	3 x 3 x 5	SiPM	DLED (9 photon)	8 GSps, 10 bit	511	101
Ahmed [17]	2012	3 x 3 x 30	SiPM	DLED (20%)	40 GSps 8 bits	511	176
Régis [38]	2012	ϕ 38 x 38	PMT XP20D0	Analog	Ortec 935	511	353
Nakhostin [36]	2014	ϕ 38 x 51	PMT H10570	DCFD (7%)	4 GSps 10 bit	1,100 – 1,400	240
Nakhostin	2014	ϕ 38 x 51	PMT H10570	DCFD (7%)	500 MSps 10 bit	1,100 – 1,400	375
Nakhostin	2014	ϕ 38 x 51	PMT H10570	Analog	Ortec 583B/567	1,100 – 1,400	300
Du [35]	2015	ϕ 20 x 5	PMT XP20D0	DCFD (6%)	DRS4 5 GSps	511	195
Du	2015	ϕ 20 x 5	PMT XP20D0	Analog	Ortec CF8000	511	255
Vedia [34]	2017	ϕ 25 x 45	PMT R9779	Analog	Ortec 935/567	1,330	156
This work	2017	ϕ 25 x 25	PMT L25D19	DCFD (30%)	500 MSps, 14 bits	1,100 – 1,400	137

time jitter as a source of timing error. This suggests that similar measurements would be useful in evaluating our new algorithms as we seek to explore their resolution limits. For comparison to the ^{60}Co measurements at hand, between 1100, and 1400 keV, we see that the best results have been analog, achieving 151- [33] and 156 [34]-ps FWHM. Digital results have been somewhat poorer, achieving 195- [35] and 240 [36]-ps FWHM using 4–5 GSps digitizers. The only reported result using 500-MSps sampling is 375-ps FWHM [36], which is almost a factor of three worse than what *Sinc* achieved at 250-MSps sampling. It will be very interesting to see what timing resolutions can be achieved as we work to optimize both our data collection electronics and analysis algorithms.

VIII. CONCLUSION

We have presented two new algorithms for extracting times of arrival from digitized pulses that were developed with the threefold goal of achieving competitive results while using lower speed ADCs, operating on a single captured trace, and being capable of FPGA implementation to allow in-line operation in real time. Data sets of pulses were collected from three different signal sources and used to compare the performance of the new algorithms to three algorithms currently employed. The new algorithms provided superior performance in all cases. Further, CTR results from a ^{60}Co source using $LaBr_3$ scintillators coupled to L25D19 PMTs and sampled at 250 MHz produced a slight improvement over the best analog results and a threefold improvement over a digital result using a 500-MHz ADC. These results are very encouraging and suggest useful directions for future work, including studies with SiPM detectors, starting to develop FPGA implementations, and examining our algorithms and electronic setups to see if further gains in CTR may be possible.

REFERENCES

[1] D. E. Philippov, E. V. Popova, V. N. Belyaev, P. Z. Buzhan, A. A. Stifutkin, and S. L. Vinogradov, "Digital signal processing for SiPM timing resolution," *J. Phys. Conf. Ser.*, vol. 798, no. 1, p. 012220, Jan. 2017.

[2] L. Marcu, P. M. W. French, and D. S. Elson, *Fluorescence Lifetime Spectroscopy and Imaging: Principles and Applications in Biomedical Diagnostics*, 1st ed. New York, NY, USA: CRC Press, 2014.

[3] S. V. Paulauskas, M. Madurga, R. Grzywacz, D. Miller, S. Padgett, and H. Tan, "A digital data acquisition framework for the Versatile Array of Neutron Detectors at Low Energy (VANDLE)," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 737, pp. 22–28, Feb. 2014.

[4] G. F. Knoll, "Systems involving pulse timing," in *Radiation Detection and Measurement*, 3rd ed. New York, NY, USA: Wiley, 2000, ch. 17, sec. 9, pp. 659–679.

[5] M. Moszyński and H. Mach, "A method for picosecond lifetime measurements for neutron-rich nuclei: (2) Timing study with scintillation counters," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 277, nos. 2–3, pp. 407–417, May 1989.

[6] D. A. Gedcke and W. J. McDonald, "Design of the constant fraction of pulse height trigger for optimum time resolution," *Nucl. Instrum. Methods*, vol. 58, no. 2, pp. 253–260, 1968.

[7] V. Vedia, H. Mach, L. M. Fraile, J. M. Udías, and S. Lalkovski, "Enhanced time response of 1-in. $LaBr_3$(Ce) crystals by leading edge and constant fraction techniques," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 795, pp. 144–150, Sep. 2015.

[8] L. M. Fraile *et al.*, "Fast timing study of a $CeBr_3$ crystal: Time resolution below 120 ps at ^{60}Co energies," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 701, pp. 235–242, Feb. 2013.

[9] D. R. Schaart *et al.*, "$LaBr_3$:Ce and SiPMs for time-of-flight PET: Achieving 100 ps coincidence resolving time," *Phys. Med. Biol.*, vol. 55, no. 7, pp. N179–N189, Apr. 2010.

[10] S. Seifert *et al.*, "A comprehensive model to predict the timing resolution of SiPM-based scintillation detectors: Theory and experimental validation," *IEEE Trans. Nucl. Sci.*, vol. 59, no. 1, pp. 190–204, Feb. 2012.

[11] M. A. Nelson, B. D. Rooney, D. R. Dinwiddie, and G. S. Brunson, "Analysis of digital timing methods with BaF_2 scintillators," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 505, nos. 1–2, pp. 324–327, Jun. 2003.

[12] M. D. Aspinall *et al.*, "Sample-interpolation timing: An optimized technique for the digital measurement of time of flight for γ rays and neutrons at relatively low sampling rates," *Meas. Sci. Technol.*, vol. 20, no. 1, p. 015104, Jan. 2009.

[13] V. Modamio *et al.*, "Digital pulse-timing technique for the neutron detector array NEDA," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 775, pp. 71–76, Mar. 2015.

[14] W. K. Warburton, M. Momayezi, B. Hubbard-Nelson, and W. Skulski, "Digital pulse processing: New possibilities in nuclear spectroscopy," *Appl. Radiat. Isotopes*, vol. 53, nos. 4–5, pp. 913–920, 2000.

[15] Q. Xie *et al.*, "Potentials of digitally sampling scintillation pulses in timing determination in PET," *IEEE Trans. Nucl. Sci.*, vol. 56, no. 5, pp. 2607–2613, Oct. 2009.

[16] B. Joly *et al.*, "An optimal filter based algorithm for PET detectors with digital sampling front-end," *IEEE Trans. Nucl. Sci.*, vol. 57, no. 1, pp. 63–70, Feb. 2010.

[17] M. Ahmed, B. Camanzi, and J. Matheson, "Characterisation of silicon photomultipliers for time-of-flight PET," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 695, pp. 252–256, Dec. 2012.

[18] T. Pro *et al.*, "New developments of near-UV SiPMs at FBK," *IEEE Trans. Nucl. Sci.*, vol. 60, no. 3, pp. 2247–2253, Jun. 2013.

[19] J. Y. Yeom, R. Vinke, and C. S. Levin, "Optimizing timing performance of silicon photomultiplier-based scintillation detectors," *Phys. Med. Biol.*, vol. 58, no. 4, p. 1207, 2013.

[20] M. Aykac, I. Hong, and S. Cho, "Timing performance comparison of digital methods in positron emission tomography," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 623, no. 3, pp. 1070–1081, Nov. 2010.

[21] L. Bardelli, G. Poggi, M. Bini, G. Pasquali, and N. Taccetti, "Time measurements by means of digital sampling techniques: A study case of 100 ps FWHM time resolution with a 100 MSample/s, 12 bit digitizer," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 521, nos. 2–3, pp. 480–492, Apr. 2004.

[22] A. Fallu-Labruyere, H. Tan, W. Hennig, and W. K. Warburton, "Time resolution studies using digital constant fraction discrimination," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 579, no. 1, pp. 247–251, Aug. 2007.

[23] L. M. Fraile, J. M. Udías, A. M. Ortega, and V. Vedia, "Digital processing of scintillator signals for fast timing applications," in *Proc. IEEE NSS/MIC Conf.*, Oct./Nov. 2015, pp. 1–2.

[24] M. Jäger and T. Butz, "FPGA implementation of digital constant fraction algorithm with fractional delay for optimal time resolution," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 674, pp. 24–27, May 2012.

[25] H. Saito, Y. Nagashima, T. Kurihara, and T. Hyodo, "A new positron lifetime spectrometer using a fast digital oscilloscope and BaF_2 scintillators," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 487, no. 3, pp. 612–617, 2002.

[26] R. Vinke *et al.*, "Optimizing the timing resolution of SiPM sensors for use in TOF-PET detectors," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 610, no. 1, pp. 188–191, Oct. 2009.

[27] A. Bousselham and C. Bohm, "Sampling pulses for optimal timing," *IEEE Trans. Nucl. Sci.*, vol. 54, no. 2, pp. 320–326, Apr. 2007.

[28] M. D. Haselman, S. Hauck, T. K. Lewellen, and R. S. Miyaoka, "Simulation of algorithms for pulse timing in FPGAs," *Proc. IEEE NSS/MIC Conf.*, Oct./Nov. 2007, pp. 3161–3165.

[29] G. Wolberg, "Sampling, reconstructiion, and anitaliasing," in *Computer Science Handbook*, A. B. Tucker, Ed. 2nd ed. Boca Raton, FL, USA: Chapman & Hall, 2004, sec. 39, pp. 1–31.

[30] *Igor Pro—Technical Graphing and Data Analysis*. Accessed: Jul. 22, 2017. [Online]. Available: https://www.wavemetrics.com/products/igorpro/igorpro.htm

[31] ET Enterprises. (Jul. 1, 2017). *ET Enterprises Electron Tubes*. Accessed: Jul. 1, 2017. [Online]. Available: https://my.et-enterprises.com/search?search-source=&q=L25D19

[32] S. Gundacker, E. Auffray, P. Jarron, T. Meyer, and P. Lecoq, "On the comparison of analog and digital SiPM readout in terms of expected timing performance," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 787, pp. 6–11, Jul. 2015.

[33] M. Moszyński *et al.*, "New photonis XP20D0 photomultiplier for fast timing in nuclear medicine," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 567, no. 1, pp. 31–35, Nov. 2006.

[34] V. Vedia, M. Carmona-Gallardo, L. M. Fraile, H. Mach, and J. M. Udías, "Performance evaluation of novel $LaBr_3$ (Ce) scintillator geometries for fast-timing applications," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 857, pp. 98–105, Jun. 2017.

[35] C.-M. Du *et al.* (Oct. 2015). "Analysis of digital timing methods with DRS4 module." [Online]. Available: https://arxiv.org/abs/1506.01802

[36] M. Nakhostin, Z. Podolyak, and P. H. Regan, "Digital processing of signals from $LaBr_3$:Ce scintillation detectors," *J. Instrum.*, vol. 9, no. 12, p. C12049, 2014.

[37] S. Seifert *et al.*, "Ultra precise timing with SiPM-based TOF PET scintillation detectors," in *Proc. IEEE NSS/MIC Conf.*, Oct./ Nov. 2009, pp. 2329–2333.

[38] J.-M. Régis *et al.*, "The time-walk of analog constant fraction discriminators using very fast scintillator detectors with linear and non-linear energy response," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 684, pp. 36–45, Aug. 2012.