A fast and precise peak finder for the pulses generated by the future HEP detectors

Vasile Buzuloiu
Institutul Politehnic Bucuresti and CERN/LAA project

ABSTRACT

A new method for fast computing the peak value of a pulse and its position in time, from a few samples is described. It consists in developing a representation of the pulse in the vector space of its samples - a curve for 1-D signals - and in finding a set of optimal (hyper-)planes embedding the curve. The resulting algorithm needs a few simple arithmetic-logic operations and yet allows a very good precision. It is very well suited for implementation in VLSI chips and is intended to be used for the next generation of HEP detectors as a real time (i.e. pulse rate) feature extractor.

Introduction.

In a previous paper [1] we described a new approach to the following problem [2] of considerable concern in data processing for future HEP detectors : A detector element (including the related analog front-end electronics) when hit, produces a pulse of a known shape, with the peak and time position depending on the hit intensity and its instant (the present day devices produce pulses of tens of nanoseconds). By sampling one gets a few samples over the duration of the pulse, usually 3 - 5 samples equally spaced but randomly shifted relatively to the peak (and correspondingly, to the hit instant). This is all what one retains in a full digital version of the processing chain. Given these samples with an appropriate precision, the task is to find the *maximum value* (the *peak* or the *amplitude* of the pulse) *and its position in time* (relatively to the sampling clock) with a high *precision* (we imposed 8 bits for the amplitude in the concrete examples worked out so far) and to do it in *real time* (i.e. the *rate of computing* must equal the *pulse rate*). The actual difficulty consists of the two combined constraints : precision *and* processing speed. The references [4-8] prove that the problem has been approached over the years in different ways and it is still under investigation. The method described in [1] allows the recovery of the amplitude and of its time instant with a very good precision and very fast : it uses a very short algorithm, simply to implement. The solution is scalable and does not have basic restrictions to the linear shaping circuits.

The present paper is complementary to [1] in the sense that it explains the algorithm described there and presents new examples which support the assertion that the method is indeed efficient. Because of lack of space, here we shall not discuss neither the possible implementations (briefly analysed in [1]) nor particular implementations worked out for the DATAWAVE chip in [3]. Yet a few comments in this direction will be made.

Representation of the signal in the space of its samples.

The mathematical representation of the pulse-shaped signal (Fig.1) is a real valued function defined over a finite time slot and in this sense it is a point in an infinite dimensional space. When sampled, the signal is represented by a point in a finite dimensional space (3D if we take only 3 samples, 5D if we take 5 samples). The immediate remark in this case is that the representative point depends on the shift between the clock pulses and the signal. If we want to extract a feature of the signal from a representative point we have to look for an invariant of the whole set of representative points when the shift runs over all possible values. Let T be the sampling period. Then the shift runs over the interval [0,T]. Correspondingly the representative point in the sample space will describe a curve, the representative curve (Fig.2).

In principle any invariant of this curve (of its point set) can be used to describe a feature of the signal. In fact we need to develop a strategy of choice among all possible invariants. Firstly, what we need is an invariant which can be easily computed. On the other hand we have to deal not with one signal but merely with a family of signals corresponding to excitations of different intensities ; we would like an algorithm which uses the same invariant for any signal (any representative curve).

The piece-wise linear quasi-invariant.

Suppose the representative curve of a signal is a *plane* curve, i.e. all its points satisfy a relation of the form :

$$\sum a_i s_i = v \qquad (1)$$

where v and a_i are constant and s_i $(i = 1,\dots,p)$ are the samples in temporal order and also the coordinates in the sample space. Then v is an invariant of the curve and it is of extremely simple form, suited for the fastest computation.

If (1) holds for a signal of the family and the shaping circuit is a linear one then it holds for any signal in the family and the corresponding v's can represent the peak for them. In this case — of a linear shaping circuit — it is enough to analyse the representation of a standard pulse : that of a maximum amplitude.

Unfortunately, in general the pulses do not possess the property described by (1). Still, interesting enough, if we try to find a plane which best fits all the points of the representative curve, we realize that the errors in v are quite small (a few percent) for pulses of quite different shapes and as many as only 3-5 samples (Increasing the number of samples would facilitate coplanarity). This can be enough for many applications and it is an interesting result in itself. Nevertheless a way to drastically reduce the errors is needed while preserving a simple computation (the demand in [2] was for 8 bit precision). Obviously the solution can not be just a dot product since the best one gives errors of a few percent (even using samples unaffected by any noise). We shall say that the representative curve is in a *e*-neighbourhood of the plane if the relative error in v does not exceed *e*.

The solution is at hand in the sample space. Indeed, we only need to relax the demand that the whole representative curve must be in an *e*-neighbourhood of *one* plane : we shall try to fit *a few* planes (Fig.2), each on a segment of the curve, so that for each segment the curve remains in an *e*-neighbourhood of the corresponding plane ; and, concomitantly, we have to be able to characterize in a simple way each segment.

In our particular problem of peak calculation things are again simple. Starting with an initial p-tuple (triplet in the examples considered in this paper) for which s_1 takes the smallest value accepted as non-negligible (say 1/256 of the maximum possible value) and increasing the shift from 0 (corresponding to the initial triplet) to T, s_1 will be an increasing quantity and s_3 a decreasing one (see again Fig.1). So, the commuting points for the approximation planes (for changing the dot products) can be defined in terms of quotients of s_1 and s_3 or, equivalently, using comparisons:

If $s_1 < k_1 s_3$	then use the dot product number 1,
if $k_1 s_3 < s_1 < k_2 s_3$	then use the dot product number 2,
if $k_2 s_3 < s_1 < k_3 s_3$	then use the dot product number 3 etc. (2)

where k_1, k_2 etc. are defined by the corresponding segments of the representative curve. In this way the peak finder acts as a *(self-)adaptive piece-wise linear filter.*

The robustness problem.

Suppose we were able to cover the representative curve with a family of hyperplanes so that the error of the resulting piece-wise linear expression does not exceed a given value *e*. The question is how the error will vary when the signal samples themselves are affected by some errors. In fact, with the idea of a full digital version of the data acquisition system in mind, we are faced with digital samples affected by round-off errors in the process of A/D conversion. It is this effect we have analysed so far in our examples.

Designing the "filter".

Let us sketch the design steps for this ad-hoc filter or "feature extractor". Consider a given pulse-shaped signal of a maximum amplitude and a sampling procedure which covers the pulse with p equidistant samples. Let T be the sampling period and t_0 the "time shift" — the time between the first sample and the instant of the maximum (in principle a measure of relative position between a time referential tied with the pulse and another one tied with the sampling). Consider the initial time shift so that the first sample is at a minimum threshold value, say approximately 1/256 of the peak value v_M(in fact one can take this initial threshold much bigger) and vary it in small (and equal) steps until the first sample reaches the position of the initial second sample — i.e. over an interval T. The p-tuples will describe a discrete version of the representative curve (discrete points). Let s_{ik} be the i-th of the p samples corresponding to the k-th value of t_0 (k=1,...,N). Solve the linear system with unknowns a_i :

$$\sum a_i s_{ik} = v_M \qquad k=1,2,\ldots,N \tag{3}$$

This is an over-determined system because $N \gg p$; by its solution we understand the generalized solution obtained via singular value decomposition (SVD). Compute the errors not only for the time shifts corresponding to the N equations of (3) but also for intermediate values; usually, i.e. for different pulse shapes, the range for errors is an interval of a few pecent of v_M Consequently the whole interval T *cannot* be covered with a single hyper-plane for an error under 0.2% (and consequently no linear filter will be able of such a performance because the hyper-plane is optimally fitted). Take an N_1 , $N_1 < N$ (try first with $N_1 = 0.5N$), so that the new system :

$$\sum a_i s_{ik} = v_M \qquad k=1,2,\ldots,N_1 \tag{4}$$

has a generalized solution which fulfils the error condition on the subinterval for shifts corresponding to N_1. Compute again the errors this time using the round-off values for samples s_{ik} (including the more dense set for the shifts). In case of a big instability (compared with the admissible errors) renounce to the smallest singular value and resume the procedure ; a further reduction of N_1 may also be necessary. After obtaining the desired behaviour on the first interval $[1,N_1]$, repeat the search for N_2 , $N_1 < N_2 < N$, and so on until the whole interval [0,T] for all possible time shifts is covered (Fig.3). In practice we will need a slightly bigger coverage but for the moment, discussing the principle, this is not essential.

It is worthwhile to mention that the error of this "filter" is always less than *e*, and not just as an average, for the round-off errors in the initial data. We shall develop elsewhere the conditions for an average error less than a given value ; obviously this will impose less difficult conditions for the approximation problem (the representative curve will be covered with fewer hyper-planes).

The case of an arbitrary amplitude and the scaling.

So far we considered the pulse of the maximum amplitude and determined from it the coefficients of the dot products. For linear shaping circuits the shape will be the same for pulses of a smaller amplitude and it is obvious that the same expressions must be used for computing the amplitude : the dot products as well as the inequalities are linear in the variables s_i. Nevertheless it is preferable to always re-scale the samples so as to work in the upper half of the full scale ; the simplest method is to use the biggest sample and to multiply it with powers of 2 as to make it belong to the upper half of the scale.

Computing the time shift.

Obviously this parameter is not an invariant to itself and the image developed for the amplitude of the pulse does not hold anymore. Still it can be computed via dot products using the following linear system :

$$\sum b_i s_{ik} = t_{0k} \qquad k=1,2,\ldots,N \tag{5}$$

This time the task is simpler because the necessary precision is much smaller : 4-5 bits. We shall judge with LHC parameters : the sampling period will be of a few nanoseconds,up to 15, and a precision of 0.5-1 ns will be completely satisfactory. So we shall be able to use a single dot product i.e. a linear filter.The trouble is that in contrast with what we encountered for amplitude calculation, this time the proportionality does not hold : the time shift must be the same for pulses of smaller amplitude. So, we need to normalize the samples. Once the amplitude, v, is determined the normalization means simply a multiplication by the factor $\mu=v_M/v$. This means a division by v which we want to avoid. Here the lower precision demanded for the time shift becomes important : for the case of a final precision of 5 bits a "look up table" (LUT) of 7 bits is enough.

General remarks.

1. The problem analyzed in this paper is important for future HEP detectors. The method developed for solving it is of a broader interest because, in principle, it can be used for any other feature of the signal. In fact the method has applications in image processing (feature extraction for 2-D signals). **2.** Here is only shown one example. The interested reader may find more in [1]. **3.** We have shown implicitely that no linear filter can recover the amplitude of a pulse from 3 or 4 samples with a precision of 8 bit. **4.** In [1] we mentioned other possible embeddings of the representative curve ; using the hyper-planes is, in a sense, the simplest construction. **5.** Implementation of the algorithm in VLSI is straight forward (see also [1]). A digital implementation was already simulated [3] for the DATAWAVE chip announced by ITT Intermetall ; two variants, namely with one instruction per sample (i.e. 125 MHz sample clock) and with two instruction per sample (62 MHz sample clock) were elaborated. Nevertheless the real interest lies in chips specially designed to incorporate the algorithm. **6.** We have not worked out any example for nonlinear case but the method has no intrinsic restrictions to the linear one. The slight nonlinearities can be handled easily : instead of a curve we have to deal with a surface. Anyway we hope new application results in the near future.

Acknowledgments.

The present work is a part of the ELOISATRON project [9]. The examples were computed with MATLAB [10]. It is a pleasure to express my thanks to Alessandro Marchioro who asked me last year to find a precise solution "without division" for computing the amplitude of a pulse from a few samples. It is also with pleasure to acknowledge the continuous support of Prof. Antonino Zichichi who asked me to work for the ELOISATRON project and the support of Dr. Giorgio Innocenti for participating in CHEP92. Last but not least I would like to thank Dinu Coltuc for his essential collaboration and also Werner Krischer for his friendly help.

References.

1. V. Buzuloiu, *Real time recovery of the amplitude and shift of a pulse from its samples.* CERN/LAA/RT92-015, April, 1992.
2. A.Marchioro, *private communication.*
3. D. Coltuc, V. Buzuloiu, *Implementing a peak-finder algorithm on the DATAWAVE chip. Simulation results.* CERN/LAA, August 1992.
4. T. Pun and D. Schlatter, *Coordinates finding in the Time Projection Chamber : algorithms and resolution.* CERN/ALEPH/TPC 86-68, May 1986.
5. D Schaile et al. *A simultaneous hit finding and timing method for pulse shape analysis of drift chamber signals.* CERN/EP 85-95, June 1985.
6. A. Bhattacharia, V. Radeka et al. *Front end electronics development for SSC detectors.* Proposal.
7. S. Gadomski et al., *The deconvolution method of fast pulse shaping at hadron collides.* CERN/PPE Jan 1992.
8. *Development of high resolution silicon strip detectors for experiments at high luminosity at LHC.* RD20 Status Report Mai 1992. CERN/DRDC 92-28.
9. A.Zichichi , *The ELOISATRON project..* August 1989.
10. *The MathWork Inc.* Cochituate Place, 24 Prime Park Way Natick, MA01760 USA

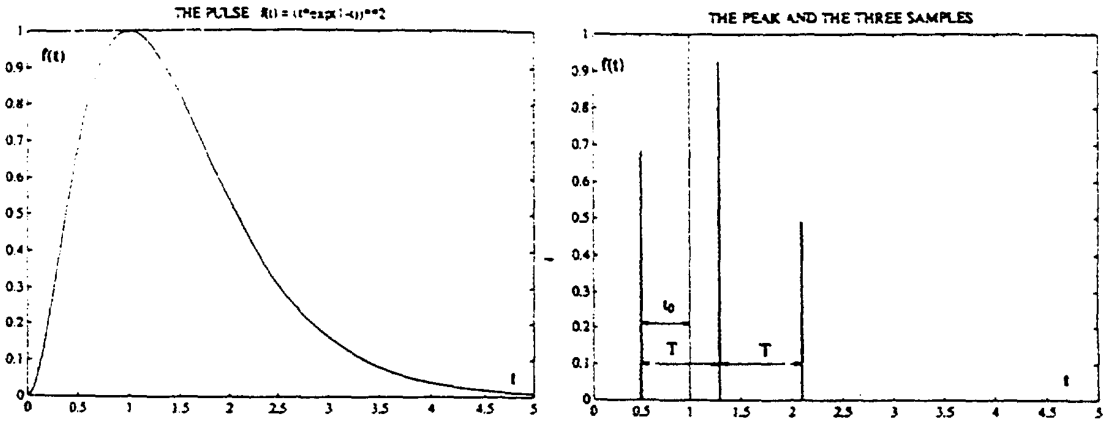

Fig. 1 The temporal representation of the pulse.

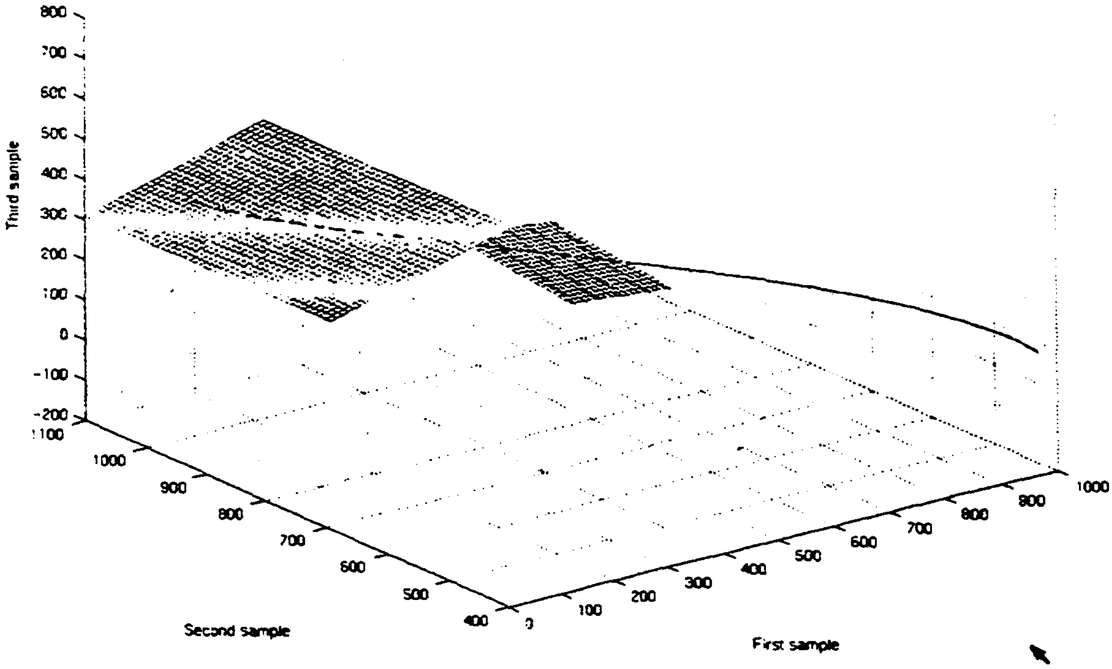

Fig. 2 The pulse representation in the space of its samples

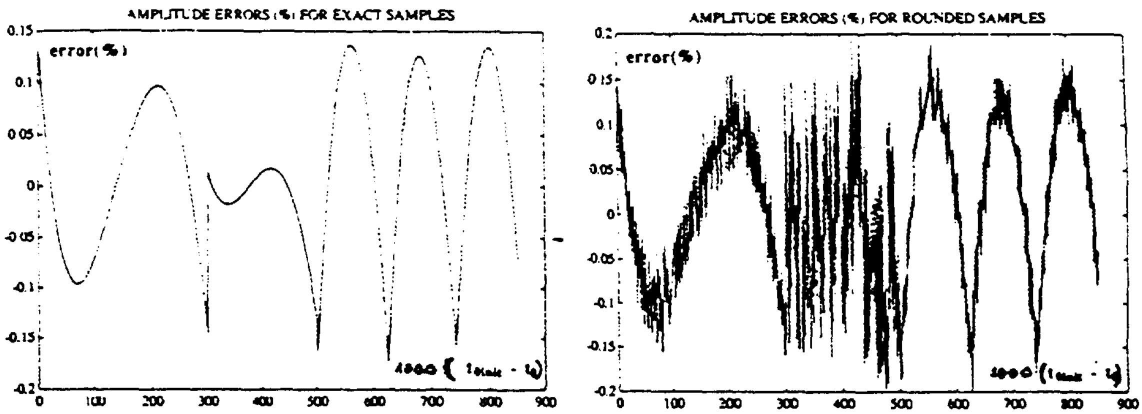

Fig. 3 The amplitude errors of the peak finder