LA-UR-15-27846
Approved for public release; distribution is unlimited.

Title: A Prescription for List-Mode Data Processing Conventions

Author(s): Beddingfield, David H.
Swinhoe, Martyn Thomas
Huszti, Jozsef
Newell, Matthew R.

Intended for: Report

Issued: 2015-10-08

LA-UR-15-xxxxx

A Prescription for List-Mode Data Processing Conventions

D.H. Beddingfield[1], M.T. Swinhoe[1], J. Huszti[2] and M.R. Newell[1]

[1]Los Alamos National Laboratory, NEN-1
[2]Hungarian Academy of Sciences, Centre For Energy Research

September 2015

Introduction

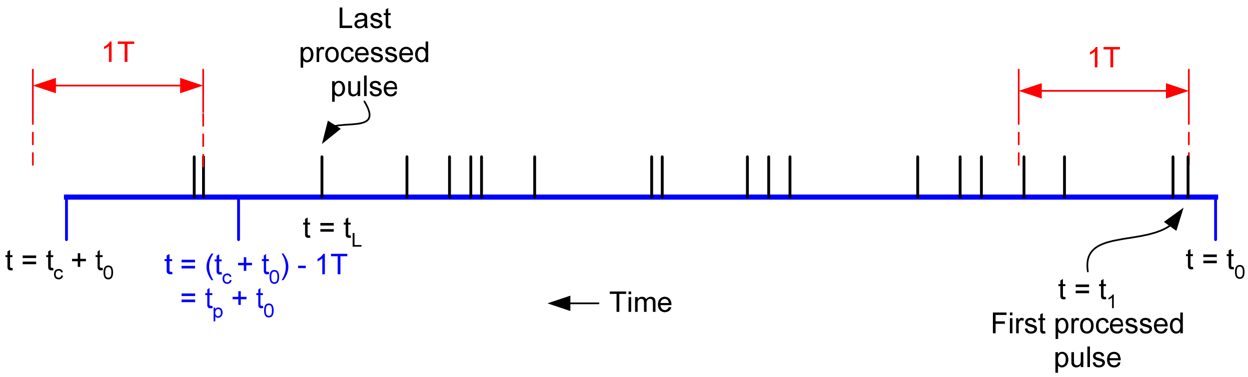
There are a variety of algorithmic approaches available to process list-mode pulse streams to produce multiplicity histograms for subsequent analysis. In the development of the INCC v6.0 code to include the processing of this data format, we have noted inconsistencies in the “processed time” between the various approaches. The processed time, t_p, is the time interval over which the recorded pulses are analyzed to construct multiplicity histograms. This is the time interval that is used to convert measured counts into count rates. The observed inconsistencies in t_p impact the reported count rate information and the determination of the error-values associated with the derived singles, doubles, and triples counting rates. This issue is particularly important in low count-rate environments.

In this report we will present a prescription for the processing of list-mode counting data that produces values that are both correct and consistent with traditional shift-register technologies. It is our objective to define conventions for list mode data processing to ensure that the results are physically valid and numerically aligned with the results from shift-register electronics.

The List-Mode Pulse Stream

The list mode pulse stream is a series of detection pulses arriving at the processing electronics over some counting period as is shown in Figure 1. The data are processed in time windows of width, T, the value of which depends upon the type of analysis to be performed. In the analysis an integer number of T-width windows are processed.

In the relatively short history of processing list mode data for safeguards purposes, we have seen inconsistencies in the value of t_p, (the true time period of the analyzed data) and in the temporal orientation of the processed gates with respect to the trigger events (before or after the trigger pulse). We have further noted the absence of defined conventions for accepting pulses that occur on gate boundaries. The latter issue is important because the data is synchronized into clock cycles. Despite these temporal issues, we have observed that the numerical derivation of the raw Singles, Doubles and Triples counts from the tabulated multiplicity distributions appears to be reliable and robust.



t_0 = time of the start of the counting period
t_c = user requested counting time
t_p = the processed time (basis for rate calculations)
t_L = time of last pulse processed ($\leq t_P$)
t_1 = time of first pulse processed ($\geq t_0$)

Figure 1. List-Mode pulse stream on an annotated time line

There are three common analyses that can be applied to the pulse stream to build up multiplicity histograms for subsequent processing into the measured singles, doubles and triples counts present in the stream. These include: 1) Traditional signal-triggered shift-register analysis, 2) Signal-triggered shift-register with fast-accidentals, and 3) Feynman analysis. The value of T shown in Figure 1 is defined for each of these three analyses in Figure 2. In Figure 2, Pd is the pre-delay setting (time after the trigger event before the foreground multiplicity gate is opened), G is the multiplicity gate width, and Ld is the long delay setting (commenced by the trigger event simultaneously with the pre-delay). The foreground and background multiplicity gate widths, G_{R+A} and G_A are identical. In the case of the signal triggered shift register with fast accidentals (FA), the G_A gate is strobed independently from the trigger events (the FA G_A gate will be addressed later). In the case of Feynman analysis the coincidence gate is strobed at a selected frequency without relying on a trigger event.

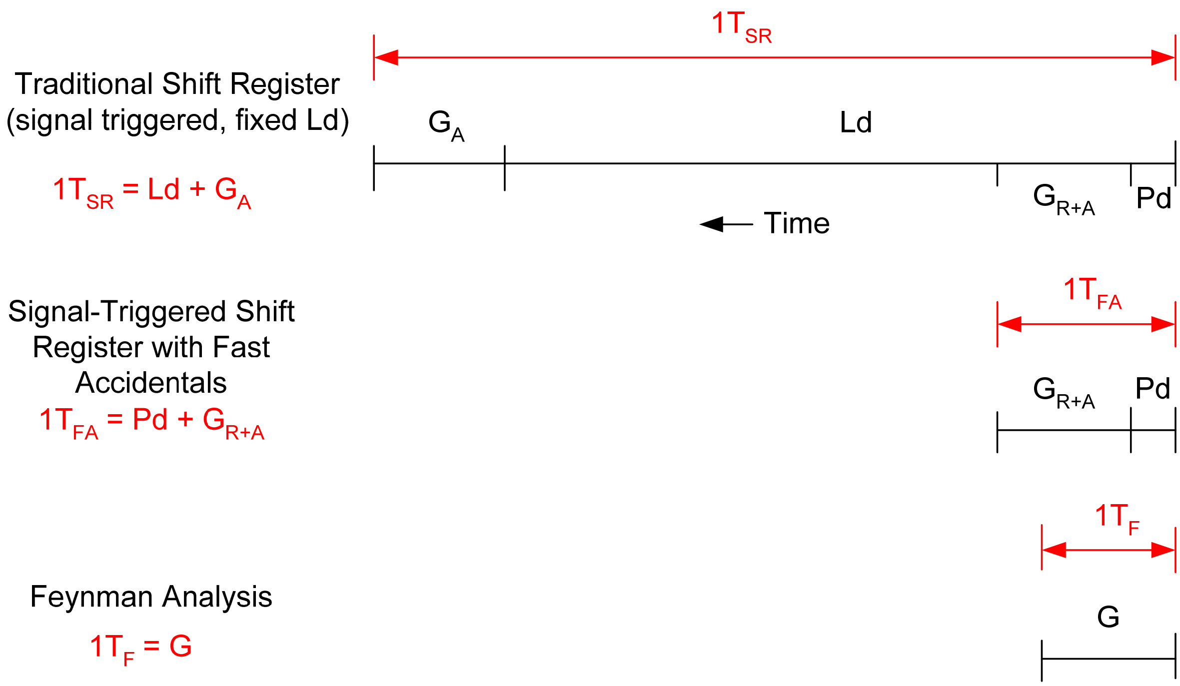


Figure 2. Definitions of T for different pulse-stream analysis methods

The Problem

When list-mode data are collected, the analysis method and the associated parameters required to determine the value of T are not necessarily determined. The only known parameter is the user requested counting time (t_c). A pulse stream of time t_c is produced for subsequent analysis. When the data are analyzed to determine a multiplicity histogram, the user defines the analysis method and appropriate parameters (Pd, G, Ld) required to determine the value of T. Because we require that an integer number of T-widths be processed, it is generally not possible to process the full counting time, t_c, and the actual processed time, t_p, of the data stream is somewhat less than t_c.

The time value of the processed data, t_p, is important because this is the time value that is used to correctly convert the measured counts determined from the multiplicity histogram into the counting rate data. The rate data are required for the subsequent point model analysis to obtain safeguards observables. Errors in the value of t_p translate directly into errors in the values of the desired safeguards observables (e.g. sample mass).

This problem is avoided in traditional shift-register electronics by pre-filling a buffer of ~1 T-width before time t_0 so that the processed time, t_p, equals the user requested counting time t_c. This approach is not available to list-mode analysis because the requisite parameters to determine the value of T are not generally available when the data are collected. Further, list-mode mode data is commonly reanalyzed with varying methods and parameters – something not possible with shift-register data.

Processed Time

It is important to have a clear definition of the processed time for use in the algorithmic reduction of list-mode to insure the fidelity of the resulting counting rate information. There seem to be a variety of intuitive options used to arrive at values of processed time that produce inconsistent results and/or results that are not in numerical agreement with shift register analysis.

The data from list-mode data collection arrive as complete sets of pulses from a particular counting period, t_c. This is fundamentally different than the traditional shift register where the data are analyzed in real time as the individual pulses pass through the shift register electronics. In effect, the list mode data is always post-processed even when used in what appears be a real-time analysis. The list-mode data are collected without consideration of the various delay and gate values required to define the required analysis time window, T. The decoupling of the data collection from the data processing results in potential issues at the start- and end-boundaries of the pulse train file.

One mistake that has been seen in list mode processing is the use of t_1, the arrival time of the first pulse (see Fig 1), as the starting time for t_p rather than t_0. Depending on the method of recording the pulse data, this can be a simple issue to address. If the data is recorded as follow-on times (time between pulses) the first pulse provides the arrival time information with respect to t_0 and can easily be processed correctly. If the data are recorded with absolute time-stamps, the information about the time offset from t_0 may be lost and the system may require modification to provide a t_0 time stamp to support the appropriate use of t_0 rather than t_1 as the starting time for t_p. This issue has no numerical impact on the development of the multiplicity histogram and the subsequent reduction to measured counts, but does have an impact on the reported counting rates that increases in severity as the counting rate decreases.

The end-boundary of the data collection presents a more challenging problem. Because we require that the data be processed in integer numbers of T-width time bins, there is an obvious problem of overrunning the end of the file. Consider the use of a traditional shift register analysis without fast accidentals. The value of T_{SR} defined in Figure 2 is applicable to this case. With typical settings, T_{SR} is a few milliseconds in width. If a trigger pulse is accepted closer than T_{SR} to the end-boundary of the file, the signal cannot be completely processed without overrunning the end of the file. This is shown on the left side of Figure 1. There are two intuitive solutions to this problem: truncate or wrap.

In the wrap solution, the pulse train at the end of the file can be wrapped to the pulse train at the beginning of the file to allow an integer T_{SR} unit to be fully processed. This solution works well for processing the accidentals gate data (G_A), but becomes problematic when the wrap is extended to the case where a trigger pulse is accepted at the end of the file and the associated R+A gate is processed from the beginning of the file. Clearly, even the wrap solution requires some truncation to avoid problems.

In the truncation solution, the processing of opening pulses is stopped 1T before the end of the file. This eliminates the algorithmic complexity and pitfalls of the wrap solution and ensures that the data processing does not overrun the end of the file. In this case, the processed time is less than the counting time by up to ~4 ms which is not a problem so long as the correct value of the processed time is used for the subsequent conversion of the processed counts to counting rates.

In the truncation solution, another possible algorithmic pitfall stems from the use of the arrival time of the last pulse before the truncation offset (t_L in Fig. 1) as the time value for the end of the processed time. In low counting rate environments, this can result in a significant bias in the reported counting rate.

To obtain a numerically correct result and apply a physically appropriate analysis we recommend that the truncation approach be adopted. For a defined analysis method with selected parameters, the last possible gate- or delay-opening event occurs at the time

$$t = (t_c + t_0) - 1T \equiv t_p + t_0,$$

which results in a processed time value of,

$$t_p = t_c - 1T.$$

This allows the complete processing of a T-width window commenced exactly 1T before the file boundary and avoids overrunning the end of the file. At time values $t_p < t \leq t_c$ it is not possible to fully process the data. The use of the defined t_p as the stopping time for data processing results in a consistent and defensible time value for conversion of measured counts into rate data.

Additional Considerations

We have presented a prescription for determination of the time interval from list-mode data processing to be used to obtain valid rate conversion applicable to the traditional shift register analysis, Feynman analysis and the fast accidentals R+A gate. But we have yet to explicitly address the A gate for fast accidentals.

When using fast accidentals, the A gate is strobed independently of the trigger events. The decoupling of the trigger event associated with the foreground (R+A) distribution and the measurement of the background (A) distribution can pose algorithmic problems for arriving at a valid result. With the adoption of the truncation solution for the basic signal processing, we propose the convention that the last gate of the fast accidentals sampling is opened as close as possible to the time of the last possible gate for the relevant t_p value. We propose this convention because of the normalization basis of the fast-accidentals distribution – the 0^{th} moment of the R+A distribution. For this normalization value to be appropriate, the processed time for the R+A and A distributions should be as close as is algorithmically possible. That is, the same time-period should be sampled for both the foreground and background counting distributions to avoid possible biases. Admittedly, the resulting bias from sampling the A distribution in temporal regions that are not sampled for inclusion in the foreground distribution is quite small. But we should strive to reduce bias sources in the data processing when possible.

The topic of algorithmic biases prompts another concern regarding pulses arriving exactly on the opening and closing boundaries of the multiplicity gates. We propose a convention that pulses arriving exactly at the opening boundary of a multiplicity gate be included in the gate while pulses arriving exactly at the closing boundary of a multiplicity gate be rejected from the gate. This avoids potential double-counting issues and is consistent with the traditional shift register electronics data treatment.

Conclusion

We have defined a prescription for the processing of list-mode data that provides numerical rate results consistent with traditional shift register analysis. This prescription provides guidance to algorithm developers for the processing of these data into multiplicity histograms and hence the subsequent conversation of derived counting data to counting rate data in a physically defensible and consistent manner. With the adoption of these conventions, we can have assurance that list-mode data are processed in a standardized manner from all algorithm developers and that the results are consistent with traditional shift register analysis.