A New Approach to Count Rate Estimation in Paralyzable Counting Systems in Nuclear Instrumentation

Sounava Chakraborty* and Dr. Amitabha Das†

*School of Nuclear Studies and Applications, Jadavpur University, Kolkata 700032, Email: sounavachakraborty@gmail.com
†Dean Academics and Professor, Department of Instrumentation and Control Engineering, Calcutta Institute of Engineering and Management, Kolkata 700040, Email: microdas@gmail.com

Keywords: Nuclear Instrumentation, Digital Pulse Processing System, Digital Detector Emulator, Dead Time, Paralyzable Model.

Abstract

Detecting and counting events generated from a radioactive source is one of the fundamental activities in nuclear or reactor instrumentation. Any practical counting system requires a minimum time interval, called the dead time, between two consecutive events to detect them as separate events. Since radioactive events are inherently statistical in nature, there is always a finite probability that the interval between two consecutive events will be less than the dead time of the counting system. This results in a substantial count loss affecting the accuracy of counting measurements. This paper focuses on the paralyzable model of dead time behaviour and has developed a new approach to estimating the true count rate of events generated from a radioactive source in presence of possible loss of events. For experimental work we have used a digital pulse processing system which contained a data acquisition firmware on a Field Programmable Gate Array (FPGA) and an Analog to Digital (ADC) board with the sampling rate of 125 Mega Samples per Second (MSPS). The input to the FPGA firmware was a digital detector emulator which generated the detector signal. We have acquired the emulated signal superimposed with noise for different event rates over which our algorithm of estimating the input count rate was applied. The whole estimation work was performed in Matlab script. The proposed algorithm has obtained better results than the established method of count rate estimation applied to systems with paralyzable dead time.

If another event arises within this time period, the detector becomes unresponsive to the second event. Such two closely placed events within the time period \( \tau \) of the counting system remains unrecognized as two distinctive events and are summed up to a single event, regarded as a pile up. Fig. 2 illustrates this phenomenon. This results in a loss of count which depends on the behavior of the dead time that whether it is non-extending (non-paralyzable) or extending (paralyzable). In the nonparalyzable model, proposed by Feller [1] and Evans [2], is based on the assumption that if any event occurs during the dead time period, it remains untreated and is lost.

![Figure 1. Typical shape of nuclear pulses](image)

The detector resolves its operation only after following a fixed or non-extending interval of time period followed by an event as shown in the Fig. 2(b). If \( m \) be the measured or observed count rate, \( n \) be the true count rate and \( \tau \) be the system dead time, the non-paralyzable model response \([1],[2]\), is given as,

\[
    n = \frac{m}{1-mr} \quad (1)
\]

The paralyzable model \([1],[2]\) assumes that occurrence of any event prior to the completion of the dead time interval \( \tau \) will not only inhibit the detector from producing the second output pulse, but also will extend the dead time by same amount of time \( \tau \) from the instant at which the second event occurs. Fig. 2 (c) illustrates the response of a paralyzable system. A detailed analysis and discussion on different models of dead time can be found in \([5]\) and \([6]\).

Interval distribution of radiation events is random in nature, and it follows a Poisson process \([3]\). For a paralyzable system, the true count rate is given by \([1],[2]\) as

\[
    m = n.e^{-n\tau} \quad (2)
\]

1 Introduction

The typical shape of pulses generated by a nuclear detector in response to a nuclear event like detection of a radioactive particle is shown in Fig. 1. Since the interval between two consecutive pulses varies randomly, one pulse may start before the previous one had died down. The phenomenon of starting of one pulse before the previous pulse ends is called pulse pile up.

In all radiation measurement techniques, the detector or the counting system have a minimum dead time period \( \tau \) that it must follow after registering one event or producing one count.
Knoll [3] suggested to solve (1) iteratively for the estimation of true count rate from the observed count rate and system dead time. Muller [4] derived a generalized formula for the true count rate \( n \) in (1). According to him the true count rate can be determined by a series expansion of the form,

\[
n = \frac{m}{1 - m\tau - \frac{1}{2}(m\tau)^2 - \frac{1}{3}(m\tau)^3 - \frac{2}{5}(m\tau)^4 - \ldots}
\]  

(3)

This paper concerns with the paralyzable system and proposes a new methodology of estimating the true count rate. We were able to obtain better results for various count rates in comparison to the method based on (3).

We structure the rest of the paper in the following manner:
Section 2: Description of the proposed algorithm.
Section 3: Description of the experimental setup.
Section 4: Presentation of the results and discussion on the results.
Section 5: Concluding remarks.

2 Algorithm

To execute the algorithm we first need to acquire the detector signal for a certain period of time \( t \). Fig. 3 shows an example of detector signal acquired over a period of 0.0060 second.

The different steps of the algorithm are described below.

Step 1: Peak detection
The height of each detector pulse is determined in the first step by detecting the peak of each pulse.

Step 2: Median of peaks
Once the peaks of all the pulses are obtained, the median of the peaks is determined. Here we exploit the robustness of median against the outliers. We observe in Fig. 1 that due to pile up there will be an increase in the pulse height. In Fig. 3 we observe that most of the pulses are of uniform height and due to pile up some of the pulses will have higher or outlier peaks. The value of the median peak will not be affected by the outlier peaks whereas the mean value of the peaks would have been affected by the presence of outlier peaks.

Step 3: Logic pulse generation and estimation of the measured count rate
In this step we utilize the median peak to determine the width of each detector pulse. Once a detector pulse exceeds a predetermined threshold level (Thl) based on the median peak, the logic pulse changes state to logic 1 and remains in logic 1 state till the detector pulse falls below the threshold level Thl. Fig. 4 illustrates the process logic pulse generation. Dividing the total number of logic pulses obtained in a single acquisition period \( t \) gives the measured count rate \( m \).

Step 4: Determination of the width of detector pulse not affected by pile up
In this step the widths of the logic pulses are arranged in ascending order. Fig. 5 shows a typical example of how the ascending widths are distributed. The horizontal axis in Fig. 5 represents the index of the widths. The leftmost index points to the minimum width while the rightmost index points to the maximum width. We observe a flat portion followed by a sudden rise in the pulse width. The flat portion represents the pulses that are not affected by pile up whereas the steep rising portion represents increase in the width of the pulses due to pile up. We select the width of the pulse at the onset of the rising slope as the representative width \( wr \) of the pulses not affected by pile up.

Step 5: Pulse recovery
After determining \( wr \), we perform the following operation for each of the logic pulses:

\[
Number\ of\ Recovered\ pulses = \text{ceil} \left( \frac{wp}{wr} \right)
\]

where \( wp \) = width of the logic pulse and \( \text{ceil}(x) \) = smallest integer greater than or equal to \( x \).

The ceiling operation essentially attempts to estimate the total number of pulses that can be recovered from a single piled-up pulse. As for example if one piled-up pulse has a width of 9 data samples, and \( wr = 5 \), the number of recovered pulses will be \( \text{ceil}(9/5) = 2 \). Figure 6 illustrates this phenomenon.

Step 6: Estimation of true count rate
In the final step we sum up the total number of recovered pulses and then divide it by the acquisition time \( t \) to estimate the true count rate \( n \).

The steps 1 to 6 can be executed for each acquisition period \( t \) and we can have a continuous estimation of the true count rate \( n \).

Figure 3: (a) Occurrence of events, (b) Response of non-paralyzable system, (c) Response of paralyzable system.

Figure 4 illustrates the process logic pulse generation.

The horizontal axis in Fig. 5 represents the index of the widths. The leftmost index points to the minimum width while the rightmost index points to the maximum width. We observe a flat portion followed by a sudden rise in the pulse width. The flat portion represents the pulses that are not affected by pile up whereas the steep rising portion represents increase in the width of the pulses due to pile up. We select the width of the pulse at the onset of the rising slope as the representative width \( wr \) of the pulses not affected by pile up.
3 Experimental Setup

The Block diagram of the experimental setup is shown in Fig. 7 and the connected hardware is shown in Fig. 8. The setup comprises of,
a) Digital Detector Emulator: The emulator can be set to generate detector pulses with the specified average count rate in Kilo Counts per Second (KCPS), rise time and fall time in micro-second, amplitude in volt and a noise level in decibel.

b) Analog to digital converter (ADC): A 14-bit ADC with sampling rate of 125 Mega Samples per Second (MSPS) was used.

c) Xilinx Virtex-6 Field Programmable Gate Array (FPGA) board: The data acquisition firmware was implemented on the FPGA board. The input to the FPGA board was the sampled data from the ADC that was stored in the embedded first-in-first-out (FIFO) memory. The stored data were in turn sent to the personal computer (PC) for further analysis.

d) PC: It was connected to the FPGA board through Ethernet connectivity. The algorithm described in this paper was implemented on the PC using Matlab.

### Figure 8: Hardware setup

<table>
<thead>
<tr>
<th>Observation Number</th>
<th>Emulator count rate (KCPS)</th>
<th>Count rate estimated by the proposed algorithm (KCPS)</th>
<th>Count rate estimated by (3) (KCPS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15</td>
<td>13.81</td>
<td>14.25</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>15.70</td>
<td>15.57</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>17.92</td>
<td>21.25</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>12.88</td>
<td>12.50</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>12.20</td>
<td>15.19</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>14.50</td>
<td>15.75</td>
</tr>
</tbody>
</table>

Table 1: Emulator data at 15 KCPS

<table>
<thead>
<tr>
<th>Observation Number</th>
<th>Emulator count rate (KCPS)</th>
<th>Count rate estimated by the proposed algorithm (KCPS)</th>
<th>Count rate estimated by (3) (KCPS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>19.72</td>
<td>19.45</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>21.47</td>
<td>26.06</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>17.18</td>
<td>15.15</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>22.19</td>
<td>14.59</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>18.06</td>
<td>16.44</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>19.72</td>
<td>18.33</td>
</tr>
</tbody>
</table>

Table 2: Emulator data at 20 KCPS

<table>
<thead>
<tr>
<th>Observation Number</th>
<th>Emulator count rate (KCPS)</th>
<th>Count rate estimated by the proposed algorithm (KCPS)</th>
<th>Count rate estimated by (3) (KCPS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25</td>
<td>25.19</td>
<td>24.69</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>26.78</td>
<td>32.95</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>24.41</td>
<td>21.41</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>25.67</td>
<td>23.39</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>23.80</td>
<td>29.25</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>25.17</td>
<td>26.33</td>
</tr>
</tbody>
</table>

Table 3: Emulator data at 25 KCPS

4 Experimental Results and Discussions

For emulating the real time detector signal, the digital detector emulator was set to produce the detector signal at 0.2 Volt amplitude, 0.10 micro-second rise time, 10 micro-second fall time. The output signal from the emulator was superimposed with a 10 dB white Gaussian noise. After the acquisition of the pulses at a specified average count rate in KCPS, the proposed algorithm as discussed in Section 2 was implemented to achieve the true count rate of the counting system considering losses and presence of pile-up in the emulated detector signal.

Total collection time of the emulated data is obtained by multiplying total data points with the ADC output i.e. 125 MSPS sampling rate or 8 nano-second sampling time period. Tables 1 to 5 show the experimental results of estimating the true count rate of the counting system for different input count rates and also compare them with the count rates as estimated by using (3). In all the cases the threshold ($Th_l$) was set as 0.25 times the median peak.
The results show that the averages of the estimated count rates in all cases according to the proposed algorithm discussed in Section 2 are found to be better than the averages of the estimated count rates given by (3).

5. Conclusion

In all nuclear radiation pulse counting systems, measurement accuracy is one of the most important factor especially in high count rate applications. Since the timing interval between two consecutive events from a radioactive source is inherently random in nature, there is a finite probability that two events almost occur at the same instance of time. This causes pile-up in the detection system resulting in a major count loss in the counting system. This paper has established a novel algorithm of estimating the true count rate of the paralyzable counting system. The proposed algorithm is found to be reliable and better than the established methods [4] as shown in Section 4.

The future prospect of this work is to implement this algorithm in an all-digital FPGA system, where the pulse rate estimation will be carried out in a real time environment.

Acknowledgements

The authors would sincerely like to thank Dr. Amitava Gupta, Head of School of Nuclear Studies and Applications, Jadavpur University, for his immense support and encouragement towards this paper. Special thanks should also be given to the facilities provided by the Data Acquisition (DAQ) Section of Variable Energy Cyclotron Centre, Kolkata, to carry out the experimental work of this paper.

References


