# Abstract

The fact that pulses from a radiation detector are randomly spaced in time can lead to interfering effects between pulses, which are more likely to occur as the count rate increases. These effects are generally called pile-up. It is known that in a scintillation detector the highest possible radiation counting rate is controlled by the pulse pile-up characteristics of the detection system. In high true count rate (TCR) spectrometry, the pulse pile-up effect is considerable, and its spectrum distortion can only be determined by Monte Carlo simulation. Monte Carlo simulation code is developed to simulate the pulse piled up spectra for high count rates for given spectra. In this work Cs-137 spectrum is used and pulse piled up spectra is simulated at various count rates.

# Introduction

Pile-up phenomenon is well known; it can be divided into two general types, which have somewhat different effects on pulse height measurements. The first is known as peak pileup: it occurs when two or more pulses are sufficiently close together to be treated as a single pulse by the analysis system. The pile-up of pulses has the effect of removing them from the proper position in the pulse height spectrum, and the area under the full-energy peak in the spectrum will no longer be a true measure of the total number of full-energy events. The second type of pile-up is called tail pile-up and involves the superposition of the tail of a pulse to the next one. The effect of tail pile-up on the measurement is to worsen the energy resolution by adding wings to the shape of the recorded peaks. Since tail pile-up does not change the location of acquired events within the energy spectrum[1].

In the statistical analysis of pile-up, it is assumed that any inherent dead time of the detector and the preamplifier is small compared with the pile-up resolution time τ of the pulse processing system[2]. This time τ is defined as the minimum time that must separate two events to avoid pile-up. Thus, the events which arrive at the amplifier with Poisson distribution are assumed to pile-up only if they occur within a time spacing less than τ[3].

True events are assumed to occur at a rate n and, due to pile-up, the recording system will perceive counts at a lower rate m, so

In general, the probability for a given count to be formed from the pile-up of (x+1) true events are

The paralyzable formulation have been used in Monte Carlo pile-up simulation. The algorithm is given in Figure 1.

Calculate the probabilities of pile up for any True count rate

Generate a random number to determine pile up degree

Sample of pulses from spectral distribution

Sample of pulses from spectral distribution

Record one count in the energy channel corresponding the pile up energy

Monte Carlo Spectrum with pile up effect

Figure :Algorithm for Monte Carlo Pulse pile up simulation

# Results

The pile up effect was simulated at various count rates as depicted in the figures (2-6) below. At lower count rates the pile-up effect isn’t prominent but with increase in count rate it becomes clearer. The pile-up of peak in spectrum is observed in higher count rate at twice the channel number as shown in figure (6).

Chart, histogram

Description automatically generated

Figure : DPHS at count rate 8000 c/s

Chart, histogram

Description automatically generated

Figure :Figure 2: DPHS at count rate 40000 c/s

Chart, histogram

Description automatically generated

Figure :Figure 2: DPHS at count rate 80000 c/s

Chart, histogram

Description automatically generated

Figure :Figure 2: DPHS at count rate 400000 c/s

Chart, histogram

Description automatically generated

Figure :Figure 2: DPHS at count rate 600000 c/s

# Conclusion

Monte carlo code has been written and the pile up spectra of Cs-137b has been predicted. The results seems to be pretty reasonable as expected the pulse pile up at double energy is pilled.

# References

[1] G. F. Knoll, *Radiation detection and measurement*. John Wiley & Sons, 2010.

[2] R. P. Gardner and S. H. Lee, “Monte Carlo simulation of pulse pile up,” 1999.

[3] A. A. Mowlavi, M. de Denaro, and M. R. Fornasier, *Monte Carlo Simulation of Pile-up Effect in Gamma Spectroscopy*. IntechOpen, 2011.

# Appendix

%%---------------------------------------------%%  
 global SPwoBg Pt  
%opening spectrum file  
file=fopen('Cs137.txt','r');  
%opening backgroud file  
file2=fopen('background.txt','r');  
  
formatSpec='%f';  
%reading spectrum data  
Spectrum=fscanf(file,formatSpec);  
%reading background data  
BG=fscanf(file2,formatSpec);  
%excluding background from spectrum  
SPwoBg=Spectrum-BG;  
%--------------------------------------%  
% load('SpectrumCs137.mat');  
  
  
%sum will be required for probabilities when sampling from spectrum  
Pt=sum(SPwoBg);  
  
%---------------------------------%  
% Input %  
%---------------------------------%  
TCR=[8000 40000 80000 400000 600000]; %true count rate  
%TCR=600000; %true count rate  
tau=1E-6; %resolving time  
n\_pu=2; %number of pileup pulses to simulate  
%-----------------------------------------------------%  
for cr=1:length(TCR)  
%probability of pile up for True count rate  
%Using Paralyzable model  
P\_pu=zeros(1,n\_pu);  
for j=1:n\_pu  
P\_pu(j)=exp(-TCR(cr)\*tau)\*(1-exp(-TCR(cr)\*tau))^(j-1);  
end  
%normalize probability  
P\_pu=P\_pu./(sum(P\_pu));  
%-----------------------------------------------------%  
  
%Number of samples for simulation  
Samples=2E4;  
%vector to save count for both pileup and without pileup counts  
channel\_counts=zeros(2,length(SPwoBg));  
%----------------------------------------------------%  
%loop over each sample  
for i=1:Samples  
 %determine the number of piled up pulses  
 %impiles wether the pulse will be piled up/at what degree  
 D\_p=degree\_pileup(P\_pu,n\_pu);  
 %Sampling random pulse from orignal spectrum  
 [~,CH1]=Pulse\_height();  
 %vector to save channel number in case of pile up pulses  
 CH=zeros(1,D\_p);  
 %loop over each pilled up degree  
 for t=1:D\_p  
 %Sampling random pulses from orignal spectrum  
 [~,CH(t)]=Pulse\_height();  
 end  
 %saving pilled up count in the corresponding channel  
 channel\_counts(1,sum(CH))=channel\_counts(1,sum(CH))+1;  
 %saving non pilled or free count in channel  
 channel\_counts(2,CH1)=channel\_counts(2,CH1)+1;  
end  
  
%--------------------------------------------------------------%  
% Plots %  
%--------------------------------------------------------------%  
figure(cr)  
%plot of piled up spectrum  
plot(1:length(SPwoBg),channel\_counts(1,:))  
hold on  
%plot of spectrum with out pile up  
plot(1:length(SPwoBg),channel\_counts(2,:))  
hold off  
xlabel('Channel number'); ylabel('dN/dH');  
title(['Differential Pulse Height Spectrum at Count rate ' num2str(TCR(cr))]);  
legend('Pile up','without Pile up')  
end  
  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

## Functions %%%

%%----------------------------------------%%  
%Number of pile up pulses/degree of pile up  
function Np=degree\_pileup(P,np)  
 r=rand; %random number btw 0,1  
 for i=1:np  
 x\_l=sum(P(1:i-1));  
 x\_u=sum(P(1:i));  
 %if random number lies with in range of proabability  
 % of free pulse or higher order pile up  
 if (r>x\_l)&&(r<x\_u)  
 Np=i;  
 break  
 end  
 end  
end  
  
%Sampling from spectrum  
function [x,i]=Pulse\_height()  
 global SPwoBg Pt  
 K=length(SPwoBg);  
 r=rand; %random number  
 %pulse heights are scaled btw 0,1  
 %for which random number lies within range, that pulse/Channel is  
 %sampled  
 for i=1:K  
 P\_min=sum(SPwoBg(1:i-1))/Pt;  
 P\_max=sum(SPwoBg(1:i))/Pt;  
 if (r>P\_min)&&(r<P\_max)  
 x=SPwoBg(i);  
 break  
 end  
 end  
end