CHAPTER 4

SIMULATION STUDY
Chapter 4

Simulation Study

4.1. Monte Carlo Method

Monte Carlo Simulation is a statistical method where the result of an experiment is simulated in computer, generating Random numbers. This is an important tool for understanding the behaviour of the detectors in the particle physics experiments, in which they are or will be implemented. The term ‘Monte Carlo’ was put forwarded by John Von Nuemann and Ulam during World War II as a code for the secret work at Loss Alamos and was suggested by the gambling casinos at the city of Monte Carlo in Monaco. This method was then applied to problems related to the atomic bomb. The work involved direct simulation of behavior concerned with random neutron diffusion in fissionable material. Later this method was used to different areas including sampling of random variables from probability distributions. These choices are determined by some probability functions and then one can establish a model which simulates the process. Because, sampling from a particular distribution involves random numbers, it is called Monte Carlo simulation. Although this method was known for a long time. However, until the advent of electronic computers, this method could not be used on any significant scale. Computer is used to generates a sequence of random numbers, called pseudo-random numbers, which are used to simulate the random observations. Thus, Monte Carlo Method is an accelerated simulation of random phenomena by computer.

4.1.1. Pseudo–random number

Random numbers can be generated in computer using some known analytical functions. The name pseudo random number is given as they are not truly random in nature, being generated by systematic arithmetic process. The first algorithm for generating random variables was suggested by J. Neumann[1]. It is referred to as the mid–square method. These numbers must satisfy special tests for assessing their random nature, being generated by systematic arithmetic process. The whole series of numbers is
uniquely determined by the starting value called the seed. Which does not effect the 
process of simulation if its distribution is uniform and period is large.
The most common algorithm for generating pseudo-random numbers is called mixed 
congruential method or power residue method. Here,

\[ R_{n+1} = (mR_n + a) \mod (N) \]

Where \( R_n \) is the \( n^{th} \) random number, \( m \) is the multiplier, \( N \) is very large positive integer 
and ‘\( a \)’ is another integer. The \((n+1)^{th}\) element is obtained as a remainder when \((m R_n + a)\) 
is divided by \(N\). For a given values of \(m, N\) and first number (seed) \(R_1\), a sequence of 
numbers fairly evenly distributed over the range \((0, N)\) is obtained. A proper choice of 
constants can give random sequence in the range \((0, 1)\), which is a standard and is used to 
derive any other probability distribution function.

4.1.2. Simulation of random variables

Simulation of a real process requires random numbers following a particular 
distributive law. These can be derived by transforming or mapping the standard random 
variable (uniformly distributed over \((0,1)\)) on to the parameter space of non uniform 
distribution. For drawing a set of discrete random variables \(x_1, x_2, \ldots, x_n\) with 
corresponding probabilities \(p(x_1), p(x_2) \ldots, p(x_n)\), 

\[ \sum_{i=1}^{n} p(x_i) = 1 \]

the following steps are followed,

(i) A uniform random number \(R_u\) between 0 and 1 is drawn using a suitable 
    subroutine by mixed congruential method.

(ii) If \(0 < R_n < p(x^1)\), then \(x\) chosen as \(x_1\), if \(p(x^1) < R_n < p(x^1) + p(x^2)\), then \(x\) 
is chosen as \(x^2\) and so on.

For drawing a continuous random variable \('X'\) in the interval \((a, b)\), the corresponding 
probability density function \(f(x)\) is first normalized in the given interval.

\[ N_r \int_{a}^{b} f(x)dx = 1 \]

where \(N_r\) is the normalization constant. The value of \('Z'\) can be obtained from the 
cumulative distribution,
\[ F(X) = N \int_{a}^{x} f(x) \, dx = R \] 

It follows that \( F(a) = 0 \) and \( F(b) = 1 \) and \( F(Z) = f(x) > 1 \). \( F(X) \) is equated to random number \( R \). From equation (4.4) it is clear that all values of \( X \) can be obtained by the inverse transformation of the function \( F(X) \), i.e. \( X = F^{-1}(R) \) \hspace{1cm} (4.5)

This method is called inverse transform method.

4.2. Simulation of Pulse Generation in RPC

4.2.1. A model

The physical processes occurring inside the RPC are modeled using Monte Carlo method. The details of pulse production including primary ionization, avalanche multiplication \([3,6]\), which are considered to build the model and induced signal are discussed. Here we study the response of 2mm and 1mm gas gap RPC and a gas mixture of 50: 50 P10 and Freon gas and giving an electric field of 46KV/cm in the gas gap. The induced charge collected by the RPC is calculated using the simulation model and simulated data are compared with recorded data.

4.2.2. Primary ionization

The charge deposit is characterized by the average number of clusters per unit length and the probability distribution for the number of electrons per cluster. These numbers are calculated using Heed \([2]\) and reported in \([3]\). The graph of average number of clusters per mm for different gases versus \((\gamma-1)\) of the particle as given in \([3]\) is used to find an average ~6 cluster/mm for our gas mixture and cosmic ray muons of average energy 2 GeV. So the average distance between clusters is \( \lambda = 167 \mu m \).

The distance between clusters is exponentially distributed, so the probability to find the first cluster between position \( x \) and \( x + dx \) is,

\[ P(x) = (1/\lambda) \, e^{-\lambda x} \] 

The cluster size distribution for our gas mixture is parameterized using the graph of number of electrons per cluster \( n_e \) versus probability to find \( p_e \) as given in \([2]\)
\[ \log_{10}(n_e) = 0.8 - 0.4 \log_{10}(p_e) \]  \hspace{1cm} (4.7)

For simulation, we put the primary clusters with distance according to equation (4.6), after normalizing within the interval \((0, d)\) where \(d\) = RPC gas gap thickness. The number of electrons for each clusters is simulated from the cluster size distribution using equation (4.7).

### 4.2.3 Avalanche multiplication

Each electron in a cluster starts an avalanche due to electron multiplication. This is characterized by the Townsend co-efficient \(\alpha\) and attachment co-efficient \(\eta\), which depend on the applied electric field. These parameters are calculated using the program Imonte [4] and plotted as function of Electric fields for different gas in [3]. While the electrons in the gas gap drift towards the anode, their multiplication will fluctuate around an average given by an exponential law.

\[ \langle n(y) \rangle = e^{(\alpha-\eta)y} \]  \hspace{1cm} (4.8)

where \(y\) = distance measured from cluster position. It shows that the average electron number depends on effective Townsends co-efficients. \(\alpha_{\text{eff}} = \alpha - \eta\).

### 4.2.4 Induced signals

The movement of the electrons in the electric field finally induces a current signal on the RPC electrodes, depending on the drift velocity\(v\). The signal induced by negative and positive ions is much smaller due to their slow drift velocity and are neglected. Electron drift velocities for different gases are predicted by Magboltz [4] and reported in [3]. The current signal induced on an electrode is given by,

\[ i(t) = (E_w V_w/e_o) N(t) \]  \hspace{1cm} (4.9)

Where \(e_o\) is the electron charge, \(E_w\) (Weighting Field) is the electric field in the gas gap if we put the electrode to potential \(V_w\) and ground all other electrodes, \(v\) is the electron drift velocity and \(N(t)\) is the number of electrons present at time \(t\), \(N(t)\) is calculated by simulating the avalanches of individual primary electrons. The weighting fields depend on detector geometry. For our RPC

\[ E_w/V_w = \varepsilon_i/(2b + d\varepsilon_i) \]  \hspace{1cm} (4.10)

(79)
Where $\varepsilon_r$ = bakelite permittivity, $b$ = Bakelite thickness & $d$ is the thickness of the gas gap. Effects of space charge may be taken into account by allowing the avalanche growth only up to a certain size.

4.2.5: Space charge effect

Due to low mobility of the positive ions, space charge develops, which reduces the electric field [5]. This effect can be simulated by a very simple method of reducing the individual avalanche multiplication in each event by a suitable factor and by stopping the avalanche growth at the number of electrons corresponding to that of the maximum for 1mm RPC. In our simulation we have used multiplication factor $(10^{10})$ for 2mm gas gap RPC only.

4.2.6 Simulation method

Fortran program is written using the model described above and run on LINUX operating system [Ubuntu], where pseudo-random numbers are available as the function 'rand()'. The input parameters are:

1. Average number of primary clusters giving average distance between primary clusters ($\lambda$).
2. Electrons drift velocity in the gas as a function of electric field.
3. The probability distribution for the number of electrons per cluster.
4. Townsend co-efficient $\alpha$ and attachment co-efficient $\eta$.

The algorithm for a single event is as follows:

1. Positions of primary clusters are simulated using an exponential distribution [equation(4.6)] with mean = 6.
2. Primary electrons are put to each cluster following the cluster size distribution equation (4.7).
3. Avalanche for each single electron is simulated using equation (4.8) and procedure outlined in reference [2], this provides $N(t)$, the number of electrons at time ‘$t$’.
Induced current signal is then calculated using equation (4.9). This is multiplied by the internal resistance of the RPC [11 MΩ] to get the pulse height in mV.

Simulated rise time for each event is recorded and using the area of the induced current profile, the induced charge is calculated.

For 2mm RPC, same procedure is followed. Effects of space charge is taken into account by reducing the avalanche growth at each cluster and further by limiting it by the saturation value arrived for the 1mm RPC.

4.3 : Result and Discussion

1mm RPC:

Experimental data for output pulse heights (mV) and rise times (ns) of 1mm RPC, are analyzed using a FORTRAN program. Data are first read into an array and corresponding charge (pC) calculated using the pulse area; maximum, minimum and range are found. Distribution of pulse height and charge are compared with simulation result for the same number of events. Class intervals are suitably adjusted and both experimental and simulated data are histogrammed and compared in Figure 4.1 and Figure 4.2.

2mm RPC:

For the case of 2mm RPC same procedure is followed as for 1mm RPC data, but while performing simulation, space charge effects are considered by reducing the avalanche growth at each cluster by a constant multiplying factor and further by stopping the avalanche growth at the number of electrons corresponding to that of the maximum for 1mm RPC. Results for pulse height distribution and induced charge distribution are shown in Figure 4.3 and Figure 4.4 respectively.

We have presented an RPC simulation procedure starting from primary ionization, avalanche production finally giving induced charge and current. The output signal in the form of induced charge (pC) and pulse height (mV) are compared with experimental charge and pulse height distribution for the same number of simulated events. It is found that, there is discrepancy in the lower and higher pulse height regions of pulse height spectra [Figure 4.1 and Figure 4.3] for both 1mm and 2mm RPCs. This is
due to streamer formation, which is not taken into account in simulation the model. The
widths of the pulse height spectra are less for experimental data as compared with
simulated data in both the cases. 2mm RPC has a lower width than 1mm RPC, indicating
a better performance. The induced charge spectra deduced from experimental
measurements of pulse height and rise time of RPC pulses, are found to be broader for
both 1mm and 2mm RPCs, when compared with the corresponding simulation results
[Figure 4.2 and Figure 4.4].

![Experimental and Simulated Pulse height spectrum for 1mm RPC](image1)

**Figure 4.1:** Experimental and Simulated Pulse height spectrum for 1mm RPC

![Experimental and Simulated induced charge spectrum for 1mm RPC](image2)

**Figure 4.2:** Experimental and Simulated induced charge spectrum for 1mm RPC

(82)
Figure 4.3: Experimental and Simulated Pulse height spectrum for 2mm RPC

Figure 4.4: Experimental and Simulated induced charge spectrum for 2mm RPC

In the next chapter 5 the results of the experimental works are presented.
2. Igor Smirnov, Heed, program to compute energy loss of fast particles in gases, Version 1.01, CERN.
4. S. Baigi, Magblotz, program to compute gas transport parameters, version 2.2 CERN.