S2 simulation in detsim

December 14, 2020

1 Physical process description

The task is to simulate sensor responses to ionization electron that reached the EL plane. Each electron will travel through the gap with a fixed speed **dv** emmiting uniformely (in time) photons whose number is proportional to its initial energy. The probability for a photon to reach each sensor (both pmt and sipm plane sensors) is provided by external tables files. The time of photon detection is assumed to be instant, ie there is no time delay between photon being created and detected.

2 Implementation details

2.1 inputs

The properties of initial electrons are its position **x**, **y**; time it reached the EL plane **t**, and energy e. The number of initial electrons can be ∼150000. The total number of photons generated by each electron is **nph** = energy/elgain, where el gain is given as external parameter and does not depend on electron properties.

x, y, t and nph are inputs given as numpy arrays (doubles and nph is integer).

2.2 outputs

The output of the process is a numpy ndarray, a matrix of a size number_of_sensors x number_of_time_bins (both parameters are detector dependent). Number of sensors in sipm plane is ∼4000 and in pmts plane ∼ 60. Number of time bins for sipms is 2000 and for pmts 80000-200000. Values stored are doubles and to obtain realistic photon count a poisson is applied outside the function.

2.3 simulation : sipms

The probability for a photon to reach a sipm sensor is given in pandas dataframe as a function of z position inside EL gap (fixed to 10 in the table) and a distance to a sensor (depends on the x, y position of the electron and x, y position of all 4000 sipms). The actual format and dependencies can change in the future.

For each initial electron and each sensor a distance is calculated to extract vector of 10 values from the table. This is done in the LT_SiPM class with the method `factors=get_values(x, y, sns_id)`. For each z value a number of photons is calculated simply as nph_z = nph/10, and the times are calculated as t_z = t + z/dv. The corresponding bin index is calculated as t_indx = t_z/sipm_bin_size and the output ndarray at `sns_id, t_indx` is increased by factors[z_indx] ∗ nph_z.

2.4 simulation : pmts

The probability for a photon to reach a pmt sensor is given in pandas dataframe as a function of x, y position of a photon to each pmt sensor (the size is ~ 6000 xy position times 60 pmts). For now there is no z position dependence.

For each initial electron and each sensor a value is extracted from LT_PMT class with the method `factors=get_values(x, y, sns_id)` (here it will be a vector of size 1). The loop over electrons, sensors and partitions is the same as in case of sipms (the last loop will run just once), outputting similar ndarray.

Note that the signal is added at only one time per photon t_z_half = t + z/2/dv. Since the time bin size of pmts is much smaller than the time electron travels through the EL gap, this value should be uniformely spread over t_z/pmt_bin_size surrounding time bins, done after the loop using cythonized `spread_histogram` function.