KOPIO TNxxx

Digitization and Reconstruction with the KOPIO GEANT MC

David E. Jaffe, BNL

January 19, 2005

Abstract

A proposal to implement digitization in the KOPIO GEANT Monte Carlo is described. The goal is to provide output that can be used for reconstruction and to satisfy the recommendations of the Simulations and Background Review Committee.

1 Recommendations of Simulations and Background Review Committee

The "close out" report of the recent Simulations and Background Review [1] recommended that, as soon as possible,

1. full digitization of simulated data be implemented and
2. event reconstruction starting from raw data be performed in order to
3. validate the signal and background calculations made with the FastMC.

2 Scheme

The scheme is straightforward and uses tools and ideas described in TN068 [2] and [3]. It proposes to use PAW ntuples to output both the digitized detector information and the relevant, rudimentary detector geometry information. While PAW ntuples are probably not going to be our final method for analysis, they can solve the immediate problem and are presumably familiar to those who would be writing the necessary code. They can be also read by ROOT.

The detectors are described and enumerated in Tables 1, 2 and 3. Each PMT, APD, pre-amp, etc. corresponds to a detector element. This differs from TN068 [2] in which elements were defined based on GEANT3's sensitive detector definitions. For example, the light from 300 scintillator tiles in a single shashlyk module is directed to a single APD or PMT. This is one element.

Abbreviation	Description
PRSC	Preradiator scintillator
PRWC	Preradiator wire chamber
PRCT	Preradiator cathode strip
CAL	Calorimeter
BV	Barrel photon veto
OV	Outer preradiator photon veto
UV	Upstream photon veto
BC	Beam catcher
CV	Charged veto
DPV	Downstream photon veto

Table 1: Proposed abbreviation and description of KOPIO detectors. CV includes all charged veto detectors. DPV includes all photon veto detectors not included in the previously named detectors.

Detector Name	Detector Number	Components of element
PRSC	1	Quadrant, superlayer, layer, plank, end
PRWC	2	Quadrant, superlayer, layer, wire
PRCT	3	Quadrant, superlayer, layer, strip
CAL	4	Module
BV	5	Module
OV	6	Module
UV	7	Counter, end
BC	8	Module
CV	9	Counter, end
DPV	10	Counter, end

Table 2: Proposed name and numbering scheme of detectors and components of each detector element. A more detailed description of the PR is in Table 3.

	PRSC	PRWC	PRCT
Quadrants	4	4	4
Superlayers	8	8	8
Layers	9	8	8
Planks	?	—	—
Wires	—	?	—
Strips	—	—	?
Ends	2	—	—

Table 3: Number of components of each PR element. There are 8 superlayers in Z in each quadrant. Each superlayer contains either 9 layers of scintillator planes or 8 layers of wires or strips. The individual pieces of scintillator are called 'planks' and are assumed to be read from both ends. I don't know the precise number of scintillator planks, wires and strips.

With each element, we associate `A` and `T` counts which can be thought of as the digitized energy and time , respectively, for most detectors. There can be multiple `A` and `T` counts for a single element in a single event. Indeed, a sequence of `A` and `T` counts can represent the output of a waveform digitizer (WFD). For some detectors, such as the PRWC, there is no `A` information, only the time of the wire hit. Finally the Monte Carlo "parent" or "parents" that produced each `A` and `T` should be stored. This gives an **digitization ntuple** entry of `DETECTOR`, `ELEMENT`, `A`, `T` and `PARENT` for each hit. These are all integers.

Most, if not all, information that would be included in the **digitization ntuple** is available in the **predigitization ntuple** [2] already available as MC output. The main tasks to fill the **digitization ntuple** based on the **predigitization ntuple** are

1. "Consolidation" of hits: individual sensitive detector elements need to be combined to form `ELEMENT`s.

2. Digitization of each `ELEMENT`: the energy or number of photoelectrons, must be converted to an integer using some assumptions about the dynamic range and number of bits available. Similarly, the time information must be encoded as an integer.

3. Ideally, some provision would be made to include electronic noise, dead channels, etc. at some later date.

As described in Table 2, each `ELEMENT` is a uniquely defined unit of each detector and can have alternate specifications based on its components that may aid in reconstruction. For this purpose, I propose a pair of routines

1. `MapFromElement(DETECTOR,ELEMENT,`$\vec{K}$`,IERROR)`
   that returns an integer array $\vec{K}$ given a `DETECTOR` and `ELEMENT`. For the PRSC, the array $\vec{K}$ would contain the quadrant, superlayer, layer and end corresponding to the input `ELEMENT`, and

2. `MapToElement(DETECTOR,ELEMENT,`$\vec{K}$`,IERROR)`
   that returns `DETECTOR` and `ELEMENT` given $\vec{K}$.

`IERROR` $\neq 0$ always indicates an invalid input value. Obviously, $\vec{K}$ would be dimensioned to the largest size needed to accomodate the maximum number of components of any detector element. Other mapping routines might be useful, for example, rendering the CAL `ELEMENT` into two integers giving the vertical and horizontal position of the element.

For each detector element, we need to know the global position of the element. For most detector elements, we also need the dimensions of the element and the position of the readout device on the element. For this purpose, I propose three routines

1. `PositionOfElement(DETECTOR,ELEMENT,`$\vec{X}_{\rm center}$`,IERROR)`
   which returns the 3-dimensional position of the center of the detector, $\vec{X}_{\rm center}$, in the global coordinate system, given a `DETECTOR` and `ELEMENT`.

2. `DimensionsOfElement(DETECTOR,ELEMENT,`$\vec{D}$`,IERROR)`
   returns the dimensions of the element $\vec{D}$ given a `DETECTOR` and `ELEMENT`. Some detector elements, such as the BV, are not rectangular parallelopipeds so $\vec{D}$ would need to be more than 3-dimensional.

3. `PositionOfReadout(DETECTOR,ELEMENT,`$\vec{R}$`,IERROR)`
   returns $\vec{R}$, the position of the detector element readout. This is important for counters read out on both ends as well as for the CV elements that will have three PMTS on one 'end'. It is not clear to me whether $\vec{R}$ should be in global or local coordinates.

To determine $\vec{X}_{\text{center}}$, $\vec{D}$ and $\vec{R}$, geometric information must be passed from the MC to the reconstruction. I propose to do this with a **geometry ntuple** with contents that must be specified and filled by each detector. For example, for the CAL, it suffices to give the dimensions of a single module, the position of the first module and the number of modules in the horizontal and vertical directions. For the CV, one may have to simply give the global position, dimensions and readout position of every element. Needless to say, the **geometry ntuple** entries must only be filled once per MC run.

3 What needs to be done

- Identify and correct any flaws in the proposed scheme.
- Implement realistic detectors in the KOPIO GEANT MC. For example, the current BV is rectangular 'log' style geometry, not the expected cylindrical, shashlyk geometry. Likewise, the CV elements in the barrel region need to be turned into the 'tiles' proposed by Zurich and outfitted with three PMTs.
- The responsible person(s) for each detector must make contributions to the code to fill **digitization ntuple** and **geometry ntuple**.

References

[1] Simulations & Backgrounds Review, New York University, January 11-13, 2005, http://rsvp.bnl.gov/Project_Office/Reviews/RSVPSimBackJan2005.htm

[2] D.E. Jaffe, Predigitization in KOPIO GEANT MC , TN068, 23 Oct 2003.

[3] http://www.phy.bnl.gov/~djaffe/KOPIO/GEANTMC/Documentation/masterplan.23oct03.pdf