

EXPERT ROOT - Developing #63

RTelescope диджитализация

07/06/2017 08:22 AM - Vitaliy Schetinin

Status:	Открыта	Start date:	07/06/2017
Priority:	Низкий	Due date:	
Assignee:	Mikhail Kozlov	% Done:	0%
Category:	RTelescope	Estimated time:	0.00 hour
Target version:			
Description			
Необходимо разработать класс ERRTelescopeDigitizer диджитализации для детектора RTelescope.			
Моделирование отклика или диджитализация учитывает гранулярность электроники, неэффективности, шумы, тёмные счёты и продуцирует такую структуру данных сигнала, как если бы он был получен от реального детектора.			
В первом приближении все перечисленные выше эффекты учитываются приближенно параметризованными распределениями ошибок.			
Класс должен быть наследован от FairTask. Можно посмотреть структуру в ERRTelescopeHitFinder. Фактически task перегоняет данные из одного представления (к примеру ERRTelescopePoint) в другое представление (назовем этот класс ERRTelescopeDigi). В каждом событии у нас есть коллекция поинтов на входе и должна быть коллекция Digi на выходе. Digi это в перспективе структура данных, которая будет приходить с реального детектора. В реальном детекторе мы не можем определить в какое именно место сенсора попала частица мы можем сказать его номер, также не можем определить простым способом сколько частиц одновременно родилось/попало в сенсор и сколько каждая из них оставила в нем энергии, мы можем сказать только суммарную энергию (которая потом будет в отсчетах электроники, а пока в энергетических единицах), также мы не можем говорить в рамках сенсора какая именно частица в какой момент в событии в него попала, мы фиксируем время от первой, как время появления сигнала. digi соответствует сектору детектора, с которого мы электроникой юем снимать сигнал.			
Класс данных ERRTelescopeDigi нужно наследовать от FairMultiLinkedData(позволяет отслеживать на базе каких других объектов образован) и расположить в папке data. Он должен содержать (номер сенсора, номер сектора(по ним мы потом будем восстанавливать координату пролета как центр сенсора), суммарную энергию поинтов в секторе, время (как время самого первого поинта в сенсоре)). С помощью метода digi->AddLink(FairLink("RTelescopePoint",iPoint)); должен быть слинковам со всеми поинтами, которые участвовали в его создании.			
Энергию Digi разыграть как Гауссово распределение со значением суммы энергий поинтов и сигмой 10 кэВ (по умолчанию у поинтов все в ГэВ, время с сигмой 200 пс (у поинтов сейчас в нс)			
К ERRTelescopeDigitizer добавить интерфейс SetEThreshold(energy(КэВ)) устанавливающий порог энергии записи Digi. Если энергия в digi получилась ниже его - не пишем. Это нужно, чтобы отрезать шумовые энерговыделения в телескопе.			

History

#1 - 07/14/2017 07:24 AM - Vitaliy Schetinin

- Assignee changed from Anonymous to Mikhail Kozlov

#2 - 07/18/2017 08:43 AM - Vitaliy Schetinin

Немного схитрил)) На самом деле в жизни у нас есть проволочки которые считывают со всего сектора и проволочки которые считывают со всего кольца. Так что поинт на самом деле формирует в данном случае 2 digi. В один надо заносить sectorNb, в другой sensorNb(номер кольца). как распределить выделение энергии по ним узнаю немного позже. пока пиши одну и ту же в оба (тут скорее это не имеет значения, если нам важно только координату восстановить). Также тебе нужно в digi завести параметр: side. side = 0 это кольцо (к примеру), а side = 1 это сектор.

#3 - 07/18/2017 05:29 PM - Mikhail Kozlov

Не совсем понял.

Выходит, у нас остается один класс digi, в нем по параметру side определяем сектор это или кольцо и просто удваиваем данные с разницей в одном параметре?

#4 - 07/18/2017 07:56 PM - Vitaliy Schetinin

Верно. Потому что в жизни именно так и будет. Дело в том, что если у тебя в детектор попадут две частицы и загорятся два кольца и два сектора, то у тебя будут 4 предположения о том, где пролетела частица. Два из них верных и два не верных