

EXPERT ROOT - Developing #66

BeamDet идентификация частицы

07/25/2017 02:35 PM - Vitaliy Schetinin

Status:	Закрыта	Start date:	07/25/2017
Priority:	Низкий	Due date:	
Assignee:	Mikhail Kozlov	% Done:	0%
Category:	BeamDet	Estimated time:	0.00 hour
Target version:	v-0.4		

Description

Необходимо разработать класс ERBeamDetPID(particle identification) для идентификации параметров частицы, летящей в BeamDet.

Результатом работы будет один объект типа ERBeamDetParticle - fProjectile.

Компонентами ERBeamDetParticle являются 4-х вектор ее состояния (TLorentzVector), PDG код, probability - вероятность того, что мы угадали частицу правильно.

PDG и probability:

Пользователь подразумевает какую частицу(ион) он должен регистрировать, поэтому PDG код необходимо сделать интерфейсом.

Для идентификации частицы используется tof и dE в TOF.

tof - время пролета между пластиками, dE - суммарная энергия оставшаяся на них.

Для задания отбора по этим параметрам необходимо сделать интерфейс: SetBoxPID(tof1,dE1,tof2,dE2). Probability=1, когда tof между tof1, tof2, dE между dE1 и dE2, если не попало в границе probability = 0

В дальнейшем понадобится такой же отбор, но с заданием границ как эллипс.

Нужно сделать интерфейс probability threshold. Если probability получилось ниже, чем probability threshold, то писать event в выходной файл не нужно по умолчанию равен нулю.

TLorentzVector состояния на мишени

Мы определяем Энергию иона по времени пролета:

1) tof = tof1-tof2 + tof_offset

tof_offset - калибровочный параметр. Задается пользователем, нужен интерфейс.

2) Вычисляем beta фактор иона

beta = расстояние между tof(из BeamDetSetup)/tof/C (C - скорость света)

3) делаем отбор событий на абсурдность. Если не прошли - не пишем в файл.

beta между 0 и 1

4) считаем гамма фактор и модуль импульса

gamma = 1./sqrt(1.-beta*beta);

p = beta*Mprojectile*gamma; (Mprojectile из TDatabasePDG и знания PDG)

5) По импульсу и знания направления на мишени формируем TLorentzVector состояния на мишени и сам объект

fProjectile(выходной объект задачи в этом событии)

Px = p*sin(theta)*cos(phi)

Py = p*sin(theta)*sin(phi)

Pz = p*cos(theta)

E = гуглим как из импульса и массы получить полную энергию

History

#1 - 07/25/2017 05:05 PM - Mikhail Kozlov

Как определять вероятность, отличную от единицы, если не попадаем в промежутки по tof или dE?

#2 - 07/25/2017 06:05 PM - Mikhail Kozlov

2) Что такое объект fProjectile?

#3 - 07/25/2017 06:13 PM - Vitaliy Schetinin

- Description updated

#4 - 07/25/2017 06:24 PM - Mikhail Kozlov

Вопросы про объекты отпали, до меня дошло.

А вот про вероятность до сих пор не понял. Если она нулевая при непопадании в границы, то зачем интерфейс на порог по вероятности.

#5 - 07/25/2017 06:31 PM - Vitaliy Schetinin

Она либо 1, либо 0. Более сложные вещи будем потом имплементировать. Если попали в заданную область значения - 1, если нет, то 0

#6 - 07/26/2017 06:40 PM - Mikhail Kozlov

Для идентификации нужны данные и с этапа диджитализации для dE и tof, а и с этапа определения параметров трека для получения вектора импульса.

Как одновременно в симуляции получить и то и другое?

#7 - 07/27/2017 08:13 AM - Vitaliy Schetinin

Не совсем понял, что значит в симуляции. Но на этапе FairTask доступны все ветки, которые были во входных файлах (основном можно еще дополнительные добавлять. AddFriend()) и во всех предыдущих задачах. Через ioman

#8 - 07/27/2017 10:43 AM - Mikhail Kozlov

Оговорка, хотел написать в макросе вместо симуляции.

#9 - 07/27/2017 11:50 AM - Mikhail Kozlov

В ioman есть только приватный метод AddFriends().

Сейчас делаю fRun->AddFriend("reco.root"), но у меня падает программа на моменте получения значения угла из объекта трека.

Не могу понять с чем связано. То ли неправильно считывается объект из дерева, то ли просто напутал со ссылками и указателями при вытаскивании данных из объекта fBeamDetTrack.

#10 - 07/31/2017 10:56 AM - Vitaliy Schetinin

Во первых проблема была не в том, что ты не правильно работаешь с фалами, а то, что task BeamDetPID ожидает, что объект BeamDetTrack был создан. Но это происходит не всегда.

Тут на самом деле вскрылась довольно большая проблема, которая заключается в том, что мы не можем прекратить выполнение Pipeline из какого либо таска. Сейчас отбор событий у нас происходит с помощью MarkFill(kFALSE), который говорит не писать событие в выходной файл. Но при этом все последующие таски в ране будут выполняться. Это порочно, потому что в следующих тасках мы должны спрашивать был ли создан такой то объект. При этом мы даже не можем на текущий момент опросить состояние флага, который ставится с помощью MarkFill(kFALSE), просто нет такого интерфейса.

Данная ситуация видимо создана по причине того, что FairRoot предполагает, что отбор событий был совершен на этапе дака. Такой вывод я сделал, потому что sbtroot и r3broot не используют MarkFill. То есть у них количество событий на входе всегда равно количеству событий на выходе. Наша действительность немного другая и отбор событий происходит уже в анализе.

На данный момент я просто добавил в ERBeamDetPID проверку на существование объекта ERBeamTrack и все перестало падать. Кроме

того, я добавил пример макроса с AddFriend.

Я вижу два пути развития событий:

- 1) Я ошибаюсь на счет FairRoot и все таки найдется нативный способ остановки pipeline и перехода к следующему событию. Решается написанием письма Радеку
- 2) Я перегружу менеджеры FairRunAna. Будет ERRunAna, который умеет останавливать PipeLine и переходить к следующему событию.

#11 - 07/31/2017 11:17 AM - Mikhail Kozlov

Если у нас несколько тасков в одном макросе, то у нас событие обрабатывается всеми тасками, а потом осуществляется переход к следующему событию - правильно, а не так, что сначала прогоняется вся диджитизация, потом поиск трека и т.д?

#12 - 07/31/2017 11:26 AM - Vitaliy Schetinin

Да. Сначала обрабатывается всеми тасками

#13 - 07/31/2017 01:27 PM - Mikhail Kozlov

У меня неправильно было задано имя BeamDetTrack в ioman в классе ERBeamDetPID! Сейчас нормально все работает и без проверки существования объекта.

#14 - 07/31/2017 03:09 PM - Vitaliy Schetinin

Ну это проблемы не решает. Буду писать свой ран менеджер с блэджком.

Я попробовал использовать G4IonTable как калькулятор массы иона по разным наборам параметров. И понял, что это не возможно, потому что Geant оже хочет инициализации кучи своих объектов. Так что вывод простой. На этапе реконструкции пользователь сам задает все необходимое для идентификации. В данном случае надо просто добавить интерфейс для ввода массы.

#15 - 07/31/2017 04:44 PM - Mikhail Kozlov

Каким образом гарантировать, что масса пользовательского иона равна в симуляции и реконструкции?

#16 - 07/31/2017 04:46 PM - Vitaliy Schetinin

Только разумностью пользователя

#17 - 07/31/2017 06:41 PM - Mikhail Kozlov

Как можно в одном дереве получить результаты и симуляции и реконструкции? Чтобы можно было в TreeViewer проверить качество восстановления координаты на мишени.

#18 - 08/09/2017 08:09 AM - Vitaliy Schetinin

- Status changed from *Открыта* to *Закрыта*