
Subject: PulseFinder

Posted by [mjanssen](#) on Mon, 19 Jun 2006 08:12:21 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi all,

writing the PulseFinder some question came up.

1. a TrackerPulse has also the possibility to store TrackerData, so the ADC spectrum which belong to the pulse. If the data are not zero suppressed, it make no sense to store the hole spectrum.

If one split the spectrum, it is necessary to store the created objects in a new collection. How should this collection be named.

2. after the transformation in to pulses the data (ADC-Spectra) which do not belong to a pulse are no longer needed. Should they be deleted (which is not foreseen in Marlin)
(the some question came up in the TrackerRawDataToDataConverter and other converting processors)

3. I will implement a simple threshold method which can deal with positive and negative signals.
When should the double-pulse candidate bit be activated?

Subject: Re: PulseFinder

Posted by [killenberg](#) on Mon, 19 Jun 2006 09:07:33 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hallo Matthias,

1. We need the extra collection because the pulse only stores a pointer, and the object it points to must be stored somewhere else, right? How about "TPCDataInPulses"? This collection might never be used itself, because usually this data will be accessed using the pulse.

2. For not zero suppressed data only those pulse spectra belonging to pulses should be stored in the collection from 1.

For the zero suppressed version this information is the TPCData (or part of it if not all entries give valid pulses). But the TPCData should not be made transient.

For the converting processors the output of this processor is transient, so there should always be a processor after the converter which does something with the converted data. Then the original (not converted) data and the data after the next analysis step is on disk, but not the converted data. The converter really should be a dump converter and not do any modification to the data.

Subject: Re: PulseFinder
Posted by [wiene](#) on Mon, 19 Jun 2006 09:11:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

I would like to add one more question concerning TrackerPulses.
There are types of readout which do not provide Tracker(Raw)Data.
For those readouts, the raw data can directly be stored as TrackerPulse. But this raw data does not yet include any corrections (contrary to our convention for TrackerPulses).

Shall we add a collection "TPCRawPulses" for such cases?

Concerning Matthias' first question, I agree that only the "interesting" part of the pulse spectrum should be stored.

What about "TPCPulseData" as collection name?

Question 2:

What do you mean by "delete"? Delete from where?

Cheers, Peter

Subject: Re: PulseFinder
Posted by [killenberg](#) on Mon, 19 Jun 2006 10:52:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

Does the readout provide quality flags for the pulses? If not, how about using TrackerData with only one entry in the chargeValues? Then it should be possible to use the default electronics channel by channel corrector, for instance. (The pulse finder should not have any problems with this).

Subject: Re: PulseFinder
Posted by [wiene](#) on Mon, 19 Jun 2006 15:56:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

Using TrackerData (which contains partially corrected data) for raw data storage would not change anything. You would still store uncorrected data in data structures which are expected to contain corrected data. And using TrackerRawData is not possible either since it only offers ints (rather than floats) to store the pulse information. This is not suitable for electronics like those based on TDCs.

Peter

Subject: Re: PulseFinder

Posted by [killenberg](#) on Mon, 19 Jun 2006 16:22:33 GMT

[View Forum Message](#) <> [Reply to Message](#)

So one will need a processor which brings the TDC data into the "default" reconstruction chain, and the collection name indicates that there is something special about the collection and it does not follow the conventions (as you proposed using the special name "TPCRawPulses"). But I still think this should happen as early as possible to avoid double code. ChannelByChannelCorrector and PedestalSubtractor run on TrackerPulses. They have to be reimplemented for the TDC if you use TrackerPulses.

How about "TDCData"? To be correct the raw data collection then should be named "FADCRawData". And both end up in the (electronics calibrated) "TPCData".

Greetings

Martin

(btw. Does the TDC need pedestal correction at all?)

Subject: Re: PulseFinder

Posted by [wiene](#) on Mon, 19 Jun 2006 18:11:50 GMT

[View Forum Message](#) <> [Reply to Message](#)

The re-implementation of the correctors is a strong argument to move the raw data of such readouts into TrackerData objects with special collection names.

I would not suggest to use collection names like "TDCxxx" because Timepix readouts will probably use the same data flow as the TDC based ones.

Peter

Subject: Re: PulseFinder

Posted by [mjanssen](#) on Mon, 03 Jul 2006 08:49:38 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Peter,
hi Martin,

sorry for answering so late, I thought I will be subscribed automatically to the the threat I created , but Somethings went wrong.

A new idea came in my mind, which solve some problems, and make things more clear. We should split the PulseFinder in a ZeroSupressor (only threshold based search) and a PulseCalculator (which create a pulse out of every TPCData object (which is now zero suppressed)
It will not split double-pulses, but mark them, as it is foreseen in the pulse finder.

The clarify question 2: In some cases (non zero suppression by DAQ) in the LCIO file the full ADC spectrum is stored. This information are no longer needed after the zero suppression and therefore should not be stored in the new output file. Of course the inputfile will not be changed.

cheers Matthias

Subject: Re: PulseFinder
Posted by [mjanssen](#) on Mon, 03 Jul 2006 09:05:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

one year ago, the idea was to store the TDC data directly a TrackerPulse.
The processing chain will be different for the TDC data till the TrackerPulses. This means that all correction will be differend for the TDC data. I don't know which correction must be applied.

The steps after the PulseFinder (on corrected TrackerPulse) should be the same for TDC data and FADC data.

The objects containing pedestal, mapping etc. can be the same in both cases.

cheers Matthias

Subject: Re: PulseFinder
Posted by [killenberg](#) on Mon, 03 Jul 2006 09:08:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Matthias,

as LCIO does not provide the possibility to read more than one file simultaneously, you can not access the non-zero-suppressed data if something strange happened during zero suppression and the original collection is not saved. So I think the data should be stored, even if it is a waste of disk space, as long as the shortcomings caused by SIO are not solved. (If you really run out of disk space / have file size problems, you can turn it off in this special case only).

Greetings

Martin

Subject: delete a collection

Posted by [mjanssen](#) on Mon, 03 Jul 2006 09:55:58 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi,

As I discovered today Marlin provide the possibility to not store collection in the output file.

The LCIOOutputProcessor has two new optional parameters:

drops the named collections from the event

type: [StringVec]

example:

DropCollectionNames TPCHits HCalHits

drops all collections of the given type from the event

type: [StringVec]

example:

DropCollectionTypes SimTrackerHit SimCalorimeterHit

So that is the solution for some of our problems.

The question is if we still need the SetOutputTransient Parameter in our processors?

cheers Matthias

Subject: Re: delete a collection

Posted by [killenberg](#) on Mon, 03 Jul 2006 10:11:43 GMT

[View Forum Message](#) <> [Reply to Message](#)

If I remember correctly, there should be some processors which produce transient data by default

(those which only do dump conversion for instance). These processors should have the SetOutputTransient parameter to switch off this feature (or should it better be called SetOutputNonTransient?). All other processors don't need it any more, as Marlin already provides the functionality.

Greetings

Martin

Subject: Re: delete a collection
Posted by [mjanssen](#) on Mon, 03 Jul 2006 11:20:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

Yes, I agree with you, only the processors, which write transient output, should have a flag to deactivate this.

As far as the normal condition for Marlin output is non transient the flag should be named SetOutputTransient, but the default value will be 1 (=true).

cheers Matthias

Subject: MarlinTPC on ilcsoft web site
Posted by [wiene](#) on Thu, 27 Jul 2006 10:18:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

MarlinTPC is now included in the list of software packages on the <http://ilcsoft.desy.de> site.

I also put a link to the draft of the write-up on this page.

Peter

Subject: TPCCondData: TPCConditions and ChannelCorrection
Posted by [jabernathy](#) on Thu, 03 Aug 2006 19:47:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

I have one question each about the TPCConditions and ChannelCorrection classes.

In our simulation we have a transverse/defocussing parameter in units of $\mu\text{m} / \sqrt{\text{cm}}$. In the TPCConditions there are two parameters. One is `get(Long/Trans)DiffusionCoef` in units of $\sqrt{\text{mm}}$. The second is `get(Long/Trans)Defocussing` which is in units of mm. What is the meaning of these parameters and is it possible to interchange between our two systems?

Also in our simulation we simulate the electronics with two different classes. One is for the construction of a signal (number of electrons per time bin) and the other creates the fadc spectrum. There are five parameters per channel in total for these classes which are needed by the reconstruction processors. Is the correct way to store these parameters in the `ChannelCorrection::setChannelCalibrationFactors()` method or is that used for something else? How have other groups passed parameters from the simulation to the reconstruction?

Subject: Re: TPCCondData: TPCConditions and ChannelCorrection
Posted by [killenberg](#) on Fri, 04 Aug 2006 08:55:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Jason,

the diffusion coefficient describes the broadening of the charge cloud when drifting a certain distance. It should be compatible with the coefficient you are using. The unit has been reduced ($\sqrt{\text{mm}} = 1000 * \sqrt{10} * \mu\text{m} / \sqrt{\text{cm}}$), if I got it right).

The defocussing is the broadening in the readout structure, i. e. the width of a point-like charge cloud after amplification. It absorbs all defocussing effects in the readout structure. The diffusion in the transfer gaps for instance depends on the fields, which might be different in the three gaps of a triple GEM structure. Even the defocussing from a resistive foil on the pads might be in here.

The ChannelCorrection class is exactly the place to store these calibration factors. The order of the parameters will have to be discussed, so the ChannelByChannelCorrector, LinearityCorrector and GainCorrector can understand them.

Astrid did not use the ConditionsData to pass the information. I think you will be the first to implement it and we will use this as a starting point.

Greetings

Martin

Subject: Re: MarlinTPC on ilcsoft web site
Posted by [jabernathy](#) on Tue, 08 Aug 2006 18:33:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

I am getting an error when trying to download MarlinTPC:

```
cvs export: syntax error in /afs/ihf.de/group/lc/detector_simulation/cvs/marlintpc/CVSRO
OT/config: line 'head 1.1;' is missing '='
cvs export: failed to create lock directory for
`/afs/ihf.de/group/lc/detector_simulation/cvs/marlintpc/MarlinTPC'
(/afs/ihf.de/group/lc/detector_simulation/cvs/marlintpc/MarlinTPC/#cvs.lock): Permission denied
cvs export: failed to obtain dir lock in repository
`/afs/ihf.de/group/lc/detector_simulation/cvs/marlintpc/MarlinTPC'
cvs [export aborted]: read lock failed - giving up
```

Any ideas?

Thanks!

- Jason

Subject: Re: MarlinTPC on ilcsoft web site
Posted by [wiene](#) on Tue, 08 Aug 2006 19:07:36 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Jason,

please check out the code via ccvssh as explained on the web site

http://www-zeuthen.desy.de/linear_collider

This works.

I already reported this problem with creating a tarball of the repository some time ago to the repository maintainer. He answered that this will only work after we have tagged our first release.

We will see...

Peter

Subject: Re: MarlinTPC on ilcsoft web site
Posted by [vogel](#) on Tue, 08 Aug 2006 19:27:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Jason,

it is well-known that our CVSweb system is prone to occasional failure - sometimes for explicable, sometimes for inexplicable reasons... The next time you encounter any problems, just contact the maintainer of the CVS repository, Harald Vogt. He will usually be able to help!

Cheers, Adrian

Update from 2007-08-21: Harald is going to retire, Wolfgang Friebel will be his successor.

Subject: Re: PulseFinder
Posted by [jabernathy](#) on Tue, 08 Aug 2006 19:30:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

I noticed in PulseFinder there is a possibility the TrackerData is stored in a separate collection but it is not associated with the TrackerPulse it was used to create.

Should there be a line similar to:

```
newPulse->setTrackerData(newData);
```

near line 220 in PulseFinder.cc?

Thanks!

- Jason Abernathy

Subject: Re: MarlinTPC on ilcsoft web site
Posted by [jabernathy](#) on Tue, 15 Aug 2006 10:43:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

OK, thanks guys.

I guess the forums was probably not the best place to post that question.

- Jason

Subject: TrackCheater Covariance Matrices
Posted by [benjeffery](#) on Mon, 22 Jan 2007 18:50:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,
I was hoping to use the TrackCheater processor in MarlinReco but it does not currently provide covariance matrices for the tracks. I see that the helix fitter used returns the diagonal terms, are these suitable for use?
Thanks.

Subject: Tracking at Test Beam
Posted by [faucci](#) on Fri, 09 Mar 2007 20:44:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi all,
How can I write the track of the test beam (essentially a straight line parallel to Z axis) in LCIO format? The problem is the phi angle, it is not well defined.
Thanks,
Michele

Subject: Re: Tracking at Test Beam
Posted by [killenberg](#) on Tue, 13 Mar 2007 10:07:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Michele,

this is a general problem with polar coordinates: Straight lines along the z axis have an undefined phi angle. In the real detector these tracks do not occur, because such particles coming from the vertex will go along the beam pipe.

The main problem will not be phi, but $\tan(\lambda)$, which is infinity in this case. ($\tan(\lambda) = p_z / p_T$, and $p_T = 0$)
So unfortunately the LCIO track, which is designed to describe a helix caused by a B field along the z axis, is not suitable to store straight tracks in z direction.

Do you really need the particle to go along z? A TPC will have problems with this anyway, as the track is along the drift direction. How about rotating the coordinate system? We had our test beam tracks going along y.

If you really need a straight track along z, you could use a LCFixedObject with six floats for instance and create a "StraightTrack" class with cartesian coordinates from it, using 3 floats for a starting point (x_0, y_0, z_0) and three floats for the direction (dir_x, dir_y, dir_z).

Greetings
Martin

Subject: New Home for MarlinTPC
Posted by [killenberg](#) on Fri, 20 Apr 2007 08:37:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

MarlinTPC has a new home. A subversion server has been set up on
pi.physik.uni-bonn.de
The cvs repository in Zeuthen has been taken off line.

A short introduction how to get MarlinTPC and use the subversion repository can be found on
<http://pi.physik.uni-bonn.de/~svn>

To document MarlinTPC a wiki has been set up:
<https://twiki.cern.ch/twiki/bin/view/ILCTPC/MarlinTPC>
It provides a user workbook with information how to compile and use MarlinTPC, and a developer workbook with details about processor programming, usage of the version control system etc.

If you want to be informed about new versions of MarlinTPC you can subscribe to the MarlinTPC mailing list:
<https://lists.desy.de/sympa/info/ilcsoft-marlintpc>

If you want to become a registered developer with write access to the subversion repository, please contact me.

Greetings
Martin

Subject: LCFIVertex v00-01 released
Posted by [hillert](#) on Mon, 30 Apr 2007 11:15:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear colleagues,

The initial version of the LCFIVertex package, v00-01, has been released and can be downloaded from the ILC software portal

http://www-flc.desy.de/ilcsoft/ilcsoftware/LCFIVertex/psc_project_view .

Best Regards,

Sonja Hillert

Subject: TrackFitterLikelihood and Minuit
Posted by [jabernathy](#) on Fri, 04 May 2007 18:25:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

Peter has pointed out that the TrackFitterLikelihood is not compiling in the MarlinTPC trunk (sorry about that).

The problem is that it uses the newest version of Minuit. It is C++ based and written at CERN.

At the time I wrote the Fitter a standalone version of Minuit was available. It has no dependencies on outside packages. I compiled it and used that version.

Now it is available as a part of ROOT (5.08.00) as ROOT::Minuit2.

This leaves three options that I can see.

- 1) Including the standalone version with the Fitter (along with hopefully including it with the gmake process)
- 2) Adding ROOT as a dependency to MarlinTPC (ugly)
- 3) Using the standalone FORTRAN version of Minuit.

1 and 2 can probably be combined with clever programming. Which is the most likely case? Do many people already have ROOT installed and this would not be a problem?

Thank you for your input.

Subject: Re: TrackFitterLikelihood and Minuit
Posted by [wiene](#) on Sat, 05 May 2007 05:54:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Jason,

I prefer option 1, i. e. to include a standalone C++ version of Minuit in MarlinTPC (although this also has some disadvantages).

Maybe it is best to add a directory tools/minuit where the Minuit codes goes, similar to e. g. tools/LCObjectCopier.

Cheers, Peter

Subject: Re: TrackFitterLikelihood and Minuit
Posted by [killenberg](#) on Mon, 07 May 2007 11:20:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

I propose using Minuit2, which is compatible/identical with the package included in ROOT (new API with capital letters of the member function), but is also available as standalone package:

<http://seal.web.cern.ch/seal/work-packages/mathlibs/index.html>

This means the user only has to install Minuit2 if there is no ROOT available on the system. If ROOT is available one can use this, but one does not have to install the complete ROOT system in case it is not installed.

It only depends on the Makefile which Minuit2 version MarlinTPC is linked to. I will try to make the Makefile aware of both options and adapt the API of the r68 version of TrackFitterLikelihood.

Greetings

Martin

Subject: Reconstructing the z coordinate of a TPC
Posted by [killenberg](#) on Thu, 28 Jun 2007 20:40:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

currently I am working on merging the HitFinderUVic and the HitFinder in the trunk. I found the following problem:

In TPC prototypes the z coordinate is usually calculated as the drift distance (implemented in HitFinder trunk). This is convenient for analyses where for instance the resolution is investigated as a function of the drift distance.

In real TPCs at a collider the origin is certainly located at the vertex, which means at the cathode. Here z is calculated as $z = \text{maximumDriftLength} - \text{driftVelocity} * \text{driftTime}$ like implemented in HitFinderUVic.

In my opinion both methods can be useful, depending on the use case. So I am planning to implement a processor parameter CalculateZAsDriftTime, which defaults to 0 and can be set to 1 in case of prototype studies.

What Do you think about it?

Greetings

Martin

P.S.: Jason, you are using the two parameters TPC_zEndplate and TPC_electronics_backend_thickness to calculate the maximal drift distance. Obviously this is the information which you are using/is provided by your simulation. I found that there is a function called getMaxDriftLength() in gear::TPCParameters, which I would like to use. I don't now if the simulation writes the gear file or also reads it. If the former is the case, would it be possible to also write the maximal drift length to the gear file? In the latter case one just additionally writes it to the gear file and the simulation does not have to be changed at all. One just has to look that the gear file is consistent if created manually, as it is redundant information.

Subject: Re: Reconstructing the z coordinate of a TPC
Posted by [harderk](#) on Thu, 28 Jun 2007 21:09:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Martin,

Wouldn't it be better to use two different variables rather than change the meaning of one variable depending on processor parameters (that are not going to be stored in lcio files)?

Cheers,
Kristian

Subject: Re: Reconstructing the z coordinate of a TPC
Posted by [killenberg](#) on Thu, 28 Jun 2007 21:58:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Kristian,

in both cases we are reconstructing the same thing: The z coordinate in mm (which is part of the x,y,z coordinate of the lcio::TrackerHit). It is just a matter where we put the origin of the coordinate system: At the cathode in the case of a TPC at a collider, at the end plate in case of a prototype.

Greetings

Martin

btw: All our processors save their name, their parameters as well as the software revision in the run header, so we can reconstruct what has been used to create the data.

Subject: Re: Reconstructing the z coordinate of a TPC
Posted by [jabernathy](#) on Thu, 28 Jun 2007 23:08:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello!

There were two reasons I decided to use that format for calculating the position of the primary ionization:

I wasn't certain everyone was using the "nominal point of interaction" as the origin of the coordinate system. Using that formula removes dependence on a specified origin.

The function "getMaxDriftLength()" from GEAR seemed ambiguous. Is the maximum drift length the distance from the IP to the readout electronics? Or is it the distance between the two endplates?

Subject: Re: Reconstructing the z coordinate of a TPC
Posted by [killenberg](#) on Fri, 29 Jun 2007 06:59:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

to the second point: I don't think there is an ambiguity. The maximum distance an electron can drift is from the cathode to one of the endplates.

to the first point: The two parameters are not standard gear, is that correct? For the time being we should assume two half TPC with equal drift distance, the origin located at the cathode. Offsets from the nominal vertex can be handled by correction processors later. And I don't see why one should like to have the origin at 2/3 of the drift length of the left half TPC.
So I would still like to use "getMaxDriftLength()".

btw: How have you solved the problems that the two halves are "looking" into different directions?

Greetings

Martin

Subject: Re: Reconstructing the z coordinate of a TPC
Posted by [wiene](#) on Fri, 29 Jun 2007 09:03:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

I prefer to always put $z = 0$ at the cathode, also for prototypes, to avoid confusion with different coordinate systems which might arise if you do not properly check the processor parameters in the run header.

Cheers, Peter

Subject: Re: Reconstructing the z coordinate of a TPC

Posted by [harderk](#) on Fri, 29 Jun 2007 09:14:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Martin,

So basically you expect subsequent processors to check for code version and settings of the hit finder in order to get things right. I still think this is asking for trouble. You raise a good point though that there is only one variable for z coordinates in `lcio::TrackerHit`. One way around this could be to use a different name for the output hit collection whenever you use the non-standard z definition. That way you would force people to be aware of the different definition of z.

Cheers,

Kristian

Subject: Re: Reconstructing the z coordinate of a TPC

Posted by [killenberg](#) on Fri, 29 Jun 2007 09:26:48 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello,

Kristian, you are right, we should always use the `TrackerHit` coordinate in the same sense. Peter also thinks we should always have $z=0$ at the cathode, so I will implement it this way.

The technique to use collection names to determine which data is stored is a good idea. We also need it to determine whether `TrackerRawData` is coming from the TPC or from silicon sensors.

Cheers

Martin

Subject: Re: Reconstructing the z coordinate of a TPC

Posted by [jabernathy](#) on Mon, 02 Jul 2007 22:13:08 GMT

[View Forum Message](#) <> [Reply to Message](#)

My mistake. I misunderstood the physical layout of the TPC.

The "`getMaxDriftLength()`" would work perfectly and I'll make the necessary changes to my code.

The ' $z = 0$ ' at the cathode sounds great!

- Jason

Subject: Re: LCFIVertex v00-01 released
Posted by [engels](#) on Wed, 01 Aug 2007 07:16:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi everyone,

just wanted to add that the ILC software portal has changed. The LCFIVertex package is now located at:

http://ilcsoft.desy.de/portal/software_packages/lcfivertex/index_eng.html

Cheers,
Jan

Subject: TPC Event Display?
Posted by [ralf](#) on Thu, 23 Aug 2007 11:07:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

I just hacked a very simple a very specific event display for our small TPC prototype. In our group the question came up: why didn't you use CED? Or is there sth. else that works?
So I thought: why not forward this question to a broader audience

Well, here my thoughts to that (also keeping in mind that a display for MarlinTPC and the large EUDET prototype will be needed too):
From my experience, ROOT is not an option because 3D drawing is slow and very limited (works for a quick solution for a very limited geometry.)
CED is nice because it is very fast, but I think it needs some more features:

Sth. like Projections (especially on pad plane: XY, but maybe also YZ and XZ)
This of course includes handling of pad plane geometry and display.
Color code depending on charge deposited charge in a voxel / on a pad (with a palette, e.g. dark to bright). The palette would of course have to be displayed in the display with corresponding charge values.
Axis with labels and ticks (incl. values).

Some of these features would probably also be useful for other detector components.

Further features a good TPC event display could/should have are IMHO histograms for interesting values like charge per pulse etc. and maybe even a further reconstruction that results for example

in histograms of angular distributions of the tracks etc..

And it would be nice to be able to export the histograms for further analysis...

Of course ROOT pops up in my mind thinking of the histograms. Would that be very bad? (An event display won't run on the GRID).

And if ROOT is bad: is it so bad, that we reinvent the functionality (histograms etc.) and a data format for that?

For the GUI, QT would be my choice (it's also already necessary for other ilcsoft, so no new dependency would be introduced).

Has sbd. else already done some development? Maybe there is sth. else than CED (although I would like not to reinvent the ilcsoft-wheel, but improve and extend existing code)?

I think many people have at least thought about it or even gained some experience while hacking their own specific display...

An last but no least, I think we should keep in mind that it is crucial that the display will be moderatly fast... well, maybe not 25fps fast, but not like 1 event every 30s.

What do you think an display would need, should have, would be nice to have?

I could probably at least start working on that and do some parts needed.

Subject: Re: TPC Event Display?

Posted by [ralf](#) on Thu, 23 Aug 2007 12:46:50 GMT

[View Forum Message](#) <> [Reply to Message](#)

Okay, I just saw that Jason will present some infos about an event display in the MarlinTPC meeting today. Seems like I have missed the information about that development. Sorry for that

And maybe an clarification of my previous post: I talked about an online event display (of course there will be much overlap with an offline display).

EDIT:

Offline the idea came up to discuss this topic in a broader audience, so I opened up a new, more general discussion in this thread.

Subject: Evolving the Track Class

Posted by [kutschke](#) on Tue, 11 Sep 2007 20:03:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Colleagues,

I would like to start a discussion about ways in which we can improve the LCIO Track class. The discussion might also touch related classes such as hits and vertices. I have created a wiki page to launch the discussion and I invite all of you to contribute. The page is on the SLAC Confluence wiki at:

<http://confluence.slac.stanford.edu/display/ilc/Evolving+the+LCIO+Track+Class>

I prefer comments within the wiki but, if I am in the minority, we can continue the discussion within this forum. I will monitor both the wiki and this forum.

The instructions for getting an account on the wiki are at linked from:
<http://confluence.slac.stanford.edu/display/ilc/Home>

Regards,

Rob Kutschke

Subject: MarlinTPC: TPCHit
Posted by [ralf](#) on Fri, 28 Sep 2007 09:15:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello everybody.

At DESY we have a lot of old prototype measurement data which is still stored in the deprecated TPCHit format.

Since I needed to calculate some pedestals of this data recently, I wrote a processor similar to the TrackerRawDataToDataConverter which reads in data in TPCHit format and has the same output as the TrackerRawDataToDataConverter.

Now the question: has any other group data in this old format they want to reconstruct with MarlinTPC? Then I would maybe check this processor into the MarlinTPC repository. If not, I would rather not do it and keep the processor only here locally, since the TPCHit class is deprecated.

CU, Ralf

Subject: Re: MarlinTPC: TPCHit
Posted by [wiene](#) on Mon, 01 Oct 2007 07:33:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Ralf,

I would suggest to add your processor to MarlinTPC even if nobody else has old data containing TPCHits. But I would not add it to the the reconstruction directory where the EDM compliant reconstruction code resides but create a directory for it under the tools dir.

Cheers, Peter

Subject: problem on LCFIVertex with LCFIAIDAPlotProcessor
Posted by [satoru](#) on Thu, 28 Feb 2008 11:23:44 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

I am using ilcsoft on KEK machine (which I have installed),
and now try to run LCFIVertex.

I would like to use LCFIAIDAPlotProcessor, so I tried to compile it with AIDAJNI aidajni-3.2.6 and jdk1.5.0_12.

After doing cmake as documented in LCFIVertex reference manual,
make seems to be successful.

But when I try to run with the compiled library, I got following errors from Marlin:

```
<!-- ERROR loading shared library : /home/uozumis/ilcsoft/LCFIVertex/HEAD/lib/libLCFIVertex.so  
-> /home/uozumis/ilcsoft/LCFIVertex/HEAD/lib/libLCFIVertex.so: undefined symbol:  
_ZNSt24__default_alloc_templatelb1ELi0EE8allocateEj -->
```

```
ProcessorMgr::registerProcessor: unknown processor with type Plot !
```

```
ProcessorMgr::registerProcessor: unknown processor with type LCFIAIDAPlotProcessor !
```

I tried to change the java version to j2sdk1.4.2_12, but it doesn't work.

Could I have some helps on it if you have any idea?

Thank you very much,

- Satoru Uozumi

Kobe University

Subject: Re: problem on LCFIVertex with LCFIAIDAPlotProcessor
Posted by [vogel](#) on Thu, 28 Feb 2008 19:25:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Satoru,

I encountered a similar problem (Error loading shared library: undefined symbol) which was related to Gear: I had built the main Marlin executable without Gear support, but one of my processors in another software package was automatically built with Gear. When I tried to run Marlin with the external library, a global object was missing from the main executable. It helped to re-build the main Marlin executable with BUILD_WITH=GEAR;

Since you are using AIDAJNI, maybe you could try to re-build your main Marlin executable with BUILD_WITH=AIDAJNI; (plus possibly other modules which you also need, like Gear or whatever). Either run cmake; with the command line option -DBUILD_WITH="AIDAJNI GEAR ..." or edit your CMake cache with the ccmake; tool and add AIDAJNI; to the line BUILD_WITH; (you need to toggle to advanced mode to see this). If you find RAIDA; there, I suppose you should remove it.

Without knowing how AIDAJNI exactly works, I think this kind of problem should not be related to your Java version in any way.

This is just a guess, but maybe it helps.

Adrian

Subject: Re: problem on LCFIVertex with LCFIAIDAPlotProcessor
Posted by [satoru](#) on Fri, 29 Feb 2008 03:13:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Adrian,

Thank you very much for your advice !
After re-compiling Marlin and GEAR with AIDAJNI as you suggested,
the error was gone !

Thanks!
- Satoru

Subject: Conditons Data for MarlinTPC
Posted by [killenberg](#) on Fri, 02 May 2008 11:48:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

In preparation for data taking with the large prototype (LP) TPC I would like to present the current status of the TPCCondData (lccd/lcio conditions data objects in the MarlinTPC package) and discuss what has to be changed/implemented.

Current status (MarlinTPC r1076):

ADCCChannelMapping

Mapping of H/W channles to GEAR pad indices

ChannelID

PadID

Type

ChannelCorrection

Per channel calibration

Quality flags (broken, noisy)

Calibration factors

Time offset

Pedestal

Per channel

Value

Width

TPCConditions

Calibrated TPC Parameters

DriftVelocity

Diffusion (trans/long)

"Defocussing"

Amplification

GasConditions

Mixture

Pressure
Temperature
OxygenContent
WaterContent

FieldSettings

Nominal drift field
Nominal B-Field

Especiall for GEMs

GEM voltages
Transfer fields

WeatherConditions

Temperature
Humidity
Pressure

TimePixPixelMode

Mode
Status (broken/noisy)

Results from the LP software meeting on 2008-04-30
[http://ilcagenda.linearcollider.org/conferenceDisplay.py?confid=2692:](http://ilcagenda.linearcollider.org/conferenceDisplay.py?confid=2692)

The readout specific items ADCChannelMapping, ChannelCorrection, Pedestal and FieldSettings need a ModuleID to be able to handle multiple modules

The FieldSettings need information whether GEMs or Micromegas have been used, and store the Micromegas voltage.

The FieldSettings will be replaced by VoltageSettings, which is the more basic information. A handler implemented as LCCDChangeListener will provide the field settings and can recalibrate

the fields in case improved information on the cathode position or GEM spacing is available. Data objects for TriggerConditions (cosmic or beam trigger, which scintillator slab has fired) and BeamConditions (particle typ, particle energy) have to be defined. The magnetic field map and the map with electric field distortions are stored as conditions data (detailes description see below). There is a B-Field in GEAR, but it is only a homogeneous field. Especially the E-field can change due to ion backdrift, so it cannot be stored in the geometry description.

The magnetic and electric field consist of a simple conditionsdata class to store the field on a 3D grid

3 ints (x, y, z) for the voxel index
3 floats for the field in x, y ,z

The grid infomation (coordinate of voxel 0,0,0 and bin width in x, y, z) are stored as collection parameters.

Handler classes allow simple access to the fields (for instance MagneticFieldHandler::getMagneticField(float *position)). These handlers update the fields and perform the overlay e. g. of static electric field and field distortions from ion backdrift. As an example the proposed class structure for the magnetic field is attached. Actually the design has already been improved wrt. this image: getMagneticField returns a struct with the three values instead of a pointer to simplify the memory management.

Proposal: The E-field and B-field classes and the handlers should not be stored together with the other conditions data but have a separate section, so one does not always have to link the whole, rather large field-handler package.

File Attachments

1) [magnetic_field_class_diagram.png](#), downloaded 739 times

Subject: Default Analyses in MarlinTPC
Posted by [killenberg](#) on Fri, 02 May 2008 12:59:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

I would like to discuss which default analysis processors should be available for the large prototype (LP) TPC.

Currently available (MarlinTPC r1076):

BiasedResidualsProcessor

Distribution of residuals wrt. track where all hits are included in track fit. Works for straight tracks and helices.

HitAndTrackChargeProcessor

Histograms of charge per hit, charge per track and charge per track length.

LinearGeometricMeanResolutionProcessor

Calculate residuals with the test hit included and excluded from the track fit. Implementation for straight lines using linear regression.

LinearThreePointResolutionProcessor

Calculate residuals using the three-point method. Implementation for straight lines.

TimePixClusterSizeProcessor

Size of clusters (number of pixels and cluster radius) on the TimePix chip

TimePixOccupancyProcessor

Count how many times a pixel has been hit on the TimePix chip

TimePixTOTDistributionProcessor

Distribution of TOT values of all pixels on the TimePix

XYZDistrubutionProcessor

Distributions of reconstructed x, y, and z coordinate of all hits in an event

XYZDistrubutionProcessor

Distributions of reconstructed x, y, and z coordinate of hits on tracks

Also needed:

ResidualsReferenceTrackProcessor

Distribution of residuals wrt. reference track

PadOccupancyProcessor

The geometric mean processor should be extended to helix tracks (lower priority since this method is time consuming. The biased residuals are ok for long tracks with many hits and can be corrected by $\sqrt{nHits / (nHits - nDoF)}$).

What else is missing?

Subject: Re: Conditons Data for MarlinTPC
Posted by [jabernathy](#) on Fri, 02 May 2008 20:32:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

Would it be useful if there were a field somewhere which stored the precision of the readout electronics?

I think it would be useful in determining whether a pulse was an overflow or underflow. However, I'm not sure if it would be a per-channel parameter.

Currently in the simulation I am working with the signal is shaped and then digitized. The digitizer has N number of bits and I store that in a ChannelCorrection::calibration factor, along with other factors.

Subject: Re: Conditons Data for MarlinTPC
Posted by [jabernathy](#) on Fri, 02 May 2008 21:18:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

Another condition which may be useful is the polarity of the readout electronics.

Subject: TPC Hits not being produced in MOKKA
Posted by [Ryan_Page](#) on Thu, 16 Oct 2008 09:26:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

I have been using MOKKA's particle gun to generate single electron events, however if I set the gun direction to 1. 0. 0. I don't get any hits in the TPC. The particle gun has an offset of 0.02 in z, this is to avoid missing hits in the vertex detector, where the two sensors meet. If I increase this offset to 5 I start to get hits in the TPC. I have also smeared the angle phi and the problem still exists, so its not one particular angle. I have also increased theta by 10 degrees, which then produces hits in the TPC. Does anyone know if such a gap should exist in the TPC or if there is another reason for this behavior? Any help would be greatly appreciated.

Thank you

Ryan

Subject: Re: TPC Hits not being produced in Mokka
Posted by [vogel](#) on Tue, 21 Oct 2008 12:58:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

Quote:Does anyone know if such a gap should exist in the TPC or if there is another reason for this behavior?

Hi Ryan,

the TPC as it is implemented in the latest Mokka geometry models actually consists of two gas volumes that are separated by a thin cathode structure. This cathode consists of two mylar sheets with a thickness of 150 μm each, and a thin copper coating of 10 μm . The copper surfaces are 5 mm apart, and the space between the sheets is filled with air. (This need not be the final design of the cathode, but it should be fine as a working model.) The sensitive volume of the TPC only extends from $|z| = 2.5$ mm up to the endplate.

Unfortunately, the release notes do not contain any information about this new TPC implementation, but you can find it out as follows (type the commands at the Idle> prompt of Mokka):

```
# to display the volume hierarchy and names
```

```
/vis/drawTree
```

```
# "TPCSensitive" and "TPCCathode" seem to be what you're interested in
```

```
/vis/scene/create
```

```
/vis/scene/add/volume TPCSensitive ! 0
```

```
/vis/sceneHandler/create OGLIX
```

```
/vis/viewer/create
```

```
/vis/viewer/set/picking
```

```
/vis/viewer/set/culling global false
```

```
/vis/viewer/flush
```

```
# left-click on the white disk (front view of the volume) to display some detailed information
```

```
# calculate the actual geometrical extent from the "half length Z" and "translation" values
```

```
# middle-click in the viewer window to exit the picking mode
```

```
/vis/scene/create
```

```
/vis/scene/add/volume TPCCathode
```

```
/vis/sceneHandler/create OGLIX
```

```
/vis/viewer/create
```

```
/vis/viewer/set/picking
```

```
/vis/viewer/set/culling global false
```

```
/vis/viewer/set/viewpointThetaPhi 90 0
```

```
/vis/viewer/zoomTo 500
```

```
/vis/viewer/flush
```

```
# left-click on the different volumes (side view of the cathode structure)
```

```
# middle-click to exit the picking mode
```

```
# now have a closer look at the mylar sheet
```

```
/vis/viewer/panTo 2.4 0 mm
/vis/viewer/zoomTo 10000
/vis/viewer/flush
# left-click on the different volumes (side view of one cathode sheet)
# middle-click to exit the picking mode
exit
Hope that answers your question,
Adrian
```

Subject: VTXNoiseHits
Posted by [walsh](#) on Wed, 05 Nov 2008 21:58:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

Does anybody ever used the processor VTXNoiseHits of MarlinReco? Does anybody know how to use that properly?

I am trying to simulate beam background using that processor but my job keeps crashing in PandoraPFA processor. I already changed the density per layer but the crash moves from one event to another.

```
#8 <signal handler called>
#9 0xf52d669a in PandoraPFAProcessor::CreatePFOCollection ()
```

If I switch off VTXNoiseHits the job runs smoothly.

Subject: Re: Conditons Data for MarlinTPC
Posted by [rosemann](#) on Thu, 17 Sep 2009 12:52:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Martin,

pushing up the old topic a bit. I think there are two items missing in the "TPCConditions" section:

- readout frequency of the electronics
- electronics polarity

And somehow I can't really spot a nice way to incorporate the conversion between ADC counts

and charge -- either in terms of number of primary electrons or the actual charge on the readout. Or is that supposed to be the "calibration factors" for the "ChannelCorrection" ?

Hoping to revive the thread,
Christoph

Subject: [MarlinTPC] PulseFinder questions and issues
Posted by [rosemann](#) on Fri, 25 Sep 2009 15:13:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

I reworked the PulseFinderProcessor of MarlinTPC/reconstruction. While encouraging testing to report any trouble, there are several issues.

1) Flagging pulses as "overflow".

When the electronics saturate (max value of bit resolution reached), then this needs to be tagged. Unfortunately the pulse finder is the wrong place to do this. Especially with /pedestal subtracted/ data the maximum adc value can only be reached, if the pedestal value (for this pad, in this run) is zero. We might need to flag this /overflow/ at a different stage. It is possible to compute the unsubtracted value (by adding the pedestal value, unelegant) or if the unsubtracted data is available to find this value (also very unelegant). But where? One place might be the RawDataToTrackerDataConverter. Or is it a processor on its own? Unfortunately only the TrackerPulses have the quality bit, which makes the producer of the TrackerPulse objects the only good place.

Any suggestions?

2) Pulse splitting

Currently the splitting of pulses is implemented, and executed regardless. I could make this optional, to allow for later splitting; maybe in a different processor.

While I strongly believe in modularisation, I'm not sure if this a good case. Somehow I would rather implement different algorithms within the existing processor. But I'm open to both suggestions as well as discussion.

3) Negative adc counts

This is only a minor issue, please assume here a positive electronics polarity. With my test cases it is actually wanted, that when the pulse "charge" (actually: summed ADC counts) is computed, negative values are actually subtracted from the total charge. But this might not be true for other,

common cases. Also the pulse time calculation by using the center of gravity is affected by this.

Cheers,
Christoph

Subject: Re: Conditons Data for MarlinTPC
Posted by [ralf](#) on Fri, 25 Sep 2009 15:23:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Another topic, that will need attention in the next weeks/months, is the way we want to store the conditions data in the database.

The way I understand it the database holds a folder structure and every folder contains the objects for one specific condition. In the folder you can access the objects by the timestamp and a tag. So if you write new (for example: corrected) conditions for a specific time there, you can later choose which ones to use by the tag. I would suggest that we tag every version. Another question is how to tag? Maybe always start with 1 (assuming that the conditions data are right in the beginning) and for smaller changes sth. like 01.01 and for larger changes 02.00 ...

For the basic folder structure, I would suggest sth. like:
/lctpc/large_prototype/<location>/<module>/<condition>/

One could even think of subdividing this structure by the types of conditions (see Martins list above)?!

Before we do this, we would also have to decide, which data is stored together in an object and which alone in a single object. For the Pedestals this is for example already fixed and the value and the width are in the same object.

Does anybody have an opinion about this? Maybe this would be a topic for a MarlinTPC Meeting or the mailing list (so every group is aware of this fact).

Subject: Re: [MarlinTPC] PulseFinder questions and issues
Posted by [killenberg](#) on Fri, 25 Sep 2009 15:56:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Christoph,

1) Flagging pulses as "overflow"

I would propose that the pedestal subtractor also knows the overflow limit and does not subtract the pedestals on the overflow samples, so this sample preserves the overflow information. I would not flag the whole TrackerData object as overflow because there could be a good pulse and an overflow pulse in non zero suppressed data.

2) Pulse splitting

I think it's good that you can turn off the splitting, one might want the unsplit pulses (if only for debugging).

3) Negative adc counts

To my understanding negative values should only occur in pre or post samples. In this case you want the negative values, otherwise the results would be biased by and depend on the number of pre/post samples. If you don't want that you have to lower the threshold and turn off the pre/post samples.

Cheers

Martin

Subject: Re: [MarlinTPC] PulseFinder questions and issues
Posted by [rosemann](#) on Fri, 25 Sep 2009 17:07:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Martin,

killenberg wrote on Fri, 25 September 2009 08:56

1) Flagging pulses as "overflow"

I would propose that the pedestal subtractor also knows the overflow limit and does not subtract the pedestals on the overflow samples, so this sample preserves the overflow information.

Alright, seems like a good idea. At least this way I don't have to change my code

Quote:

2) Pulse splitting

I think it's good that you can turn off the splitting, one might want the unsplit pulses (if only for debugging).

OK, I will incorporate this switch as an optional parameter; its default will be "splitting is on".

Thanks,
Christoph

Subject: Re: [MarlinTPC] PulseFinder questions and issues
Posted by [killenberg](#) on Fri, 25 Sep 2009 19:48:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Christoph,

Quote:

OK, I will incorporate this switch as an optional parameter;

Please make it a normal processor parameter. Optional parameters are intended to be used only if they are set, like a cut which is not applied at all if not set. A parameter whose default parameter is used should not be optional.

Example:

```
if ( parameterSet("LowCutPhi") )  
{  
    // apply the low phi cut  
}  
//otherwise don't use _lowCutPhi Parameter at all
```

Cheers

Martin

Subject: Re: [MarlinTPC] PulseFinder questions and issues
Posted by [rosemann](#) on Fri, 25 Sep 2009 20:09:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Martin,

killenberg wrote on Fri, 25 September 2009 12:48Hello Christoph,

Quote:

OK, I will incorporate this switch as an optional parameter;

Please make it a normal processor parameter. Optional parameters are intended to be used only if they are set, like a cut which is not applied at all if not set. A parameter whose default parameter is used should not be optional.

I would have assumed that this is the case here. If it's for debugging only, then (and only then) the pulses would not be split. But I can also make it a standard parameter with the default that pulses are split.

BTW: Another issue with splitting; for now I don't check split pulses for the minimal length. This because a "true" multiple-pulse doesn't fall off below the end threshold before again a rise occurs. This may shorten the length, but I didn't want to introduce yet another parameter (e.g. like "MinimalMultiPulseLength").

Cheers,
Christoph

Subject: Re: [MarlinTPC] PulseFinder questions and issues

Posted by [ralf](#) on Sat, 26 Sep 2009 16:46:52 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi,

rosemann wrote on Fri, 25 September 2009 10:07

killenberg wrote on Fri, 25 September 2009 08:56

1) Flagging pulses as "overflow"

I would propose that the pedestal subtractor also knows the overflow limit and does not subtract the pedestals on the overflow samples, so this sample preserves the overflow information.

Alright, seems like a good idea. At least this way I don't have to change my code

I think this is not a good idea, but maybe I don't understand the proposed solution correctly...

As I understand it, the following problem occurs:

In the case of non zero suppressed data you end up with a mixed collection of data vectors that are and are not pedestal subtracted... IMHO not a good idea. In addition, you can have ADC data vectors which include pulses that are in overflow and pulses that are not. So with this technique you will also affect the pulses which aren't.

CU Ralf

Subject: Re: [MarlinTPC] PulseFinder questions and issues

Posted by [killenberg](#) on Sat, 26 Sep 2009 20:45:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Ralf,

all data vectors are pedestal subtracted. Only those samples (time bins) which are in overflow are not subtracted to preserve the overflow information. So all pulses which are not in overflow are reconstructed correctly.

Something like

```
pedCorData[i] = (rawData[i]==_ADCMaxVal ? _ADCMaxVal : rawData[i] - pedestal)
```

Cheers

Martin

Subject: Re: [MarlinTPC] PulseFinder questions and issues

Posted by [ralf](#) on Sat, 26 Sep 2009 21:23:24 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Martin,

now I understand the idea When reading it the first time I misunderstood the word "samples"...

This sounds reasonable and as a bonus it is easy to implement
I will get to it next week.

CU, Ralf.

Subject: Re: [MarlinTPC] PulseFinder questions and issues
Posted by [ralf](#) on Mon, 28 Sep 2009 16:16:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

One question arises when implementing the overflow treatment in the PedestalSubtractor:
From where to get the ADCMaxVal?

IMHO it should come from the conditions database. But this is not set up right now. So it would be an effort for testing etc, to write a simple file just for this value ...

Therefore I would propose to make it an optional processor parameter. If it is set, the value from there is taken. If not, the processor looks for an entry in the database.

If neither is there, it throws an exception

I would rather not set a default value there... but the other possibility is not to do anything in case there is no value given.

What do you think?

CU, Ralf.

Subject: Re: [MarlinTPC] PulseFinder questions and issues
Posted by [killenberg](#) on Mon, 28 Sep 2009 16:42:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Ralf,

I discussed with Christoph and we both think that the value has to be there. Not to do anything at all and just continue is not a good idea. So I would really throw an exception which says that either the conditions data (or should it come from GEAR?) or the processor parameter has to be there. It's exactly the way the HitTrackFinderTopoProcessor treats the drift velocity.

Cheers

Martin

Subject: Re: [MarlinTPC] PulseFinder questions and issues
Posted by [rosemann](#) on Tue, 10 Nov 2009 16:50:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

OK, I'm still onto this processor. Doesn't seem to end.

First I've noticed some slightly confusing things, which I remedied (so far only in my private branch). Like the fact that the readout frequency is silently taken from a gear file, even it's not what you (might) want.

Anyhow, I have a pending question to which I need input: what time to store in the produced TrackerPulse collection?

The Icio documentation states only "The time of the pulse - arbitrary units." Now there are several answers to this question, and we somehow need to decide, which one we choose.

1) simply take the time bins (integer values); leave the fiddling with the electronics to the hit finding

2) take the "true" time by multiplying with the bin width; incorporating electronics information by the readout frequency. In which units then, nanoseconds?

3) take the z position (in mm) by also incorporating the drift velocity; leave then any field inhomogeneities as a correction afterwards

Please state your opinion.

Cheers,
Christoph

Subject: Re: [MarlinTPC] PulseFinder questions and issues
Posted by [killenberg](#) on Tue, 10 Nov 2009 17:02:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Christoph,

according to the TPC Data Model which we developed during the MarlinTPC Kickoff Meeting (it never made it beyond the draft status, <http://www.desy.de/~wiene/marlintpc/tpcdatamodel.pdf>) the time in the TrackerPulse is in ns. The conversion to space (mm) follows when reconstruction of the TrackerHits is done, where also the GEAR geometry information comes into the game.

Cheers

Martin

Subject: Re: Conditons Data for MarlinTPC
Posted by [ralf](#) on Fri, 04 Dec 2009 10:50:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

Regarding the folder structure in the conditions database, here is a new proposal:
/lctpc/large_prototype_1/<location>/<runnumber>/<condition >[/<module>]

I introduced the run number in my proposal. This is due to the fact, that the tagging of conditions can only be done on a folder basis. Since I guess one would want to tag conditions data for each run separately, I think we have to include the run number in the path.
The question is now where in the path?

Subject: Re: Conditons Data for MarlinTPC
Posted by [killenberg](#) on Fri, 04 Dec 2009 11:06:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Ralf,

I do not know if LCCD works correctly if the data is stored in a different folder for each run. And there is also some data that is not correlated to a run, the gas conditions or the environmental conditions for instance. These are written to the data base with a time stamp in regular intervals, no matter if a run is active or not.

For those data that are (or can be) correlated with a run this structure might be good.

I also have some questions:

What does tagging of data mean? Is it something specific for LCCD/CondDBMySQL?
What is location?

Is module the readout module on the end plate? If yes we should not have it in the folder structure but as a data member of the condition.

Cheers

Martin

Subject: Re: Conditons Data for MarlinTPC
Posted by [ralf](#) on Fri, 04 Dec 2009 13:18:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

killenberg

I do not know if LCCD works correclty if the data is stored in a different folder for each run. And there is also some data that is not correlated to a run, the gas conditions or the environmental conditions for instance. These are written to the data base with a time stamp in regular intervals, no matter if a run is active or not.

You're right, I haven't thought of these...

But this shouldn't be a problem with LCCD. You have to specify a folder for every condition object anyway and it's no problem if they differ. The only problem would be if you use run specific conditions and reconstruct data from two or more runs in one call of MarlinTPC. I guess this is no the usual use case.

So my proposal should read ([] brackets for optional folders):

```
/lctpc/large_prototype_1/<location>/[<runnumber>]/<condition >/[<module>]
```

So we could store run depended conditions with the run number and the rest without.

killenberg

For those data that are (or can be) correlated with a run this structure might be good.

I also have some questions:

What does tagging of data mean? Is it something specific for LCCD/CondDBMySQL?

What is location?

Is module the readout module on the end plate? If yes we should not have it in the folder structure but as a data member of the condition.

Here the answers:

Tagging means, that you have in LCCD the possibility to assign a tag (std::string) to a folder. This is necessary to retrieve old data if you have written new one in a folder for the same time period. Then the older data is available only via the tag.

With location I mean the physical place of the measurement like "DESY_T24-1" for example. This idea came actually from CALICE who found it useful.

Yes, I meant the readout module. This I added as an additional, optional (therefore the [] brackets) sub-folder in case there are conditions where it makes sense to split them like this. For most conditions, you wouldn't need it. Like for the pedestals, this distinction is done in the condition object. Right now I can't think of a situation, where it is necessary, but I wanted to include the idea as an option nevertheless...

CU, Ralf.

Subject: LCIO::EVENT::TrackerPulse additions
Posted by [rosemann](#) on Wed, 03 Mar 2010 14:04:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

after noticing that the two main data members of TrackerPulse (charge and time) don't have a data field for their respective error I put the issue into the feature request/bug tracker of LCIO (look here).

Two additional things come to my mind:

1) For user-friendliness AND for consistency I asked for two additional members. Is there any reason/belief that somehow a correlation between the two parameters (charge&time) exists, so we have an argument for needing a covariance matrix?

2) For the reconstruction chain further down the processing line it might be needed what the largest single slice inside a pulse is. For now this information is lost and needs to be re-retrieved when needed by looping over the spectrum.

Now I wonder if there is an argument to also call for an additional data member holding this specific information.

Cheers,
Christoph

Subject: LCIO::EVENT::TrackerHit quality information
Posted by [rosemann](#) on Wed, 03 Mar 2010 18:45:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

in the recent LCIO version there is an additional data member for the Trackerhit; the quality (with accessor/mutator). The layout for the usage was proposed formerly, but seems incomplete to me:

- bit 0: contains at least one broken channel.
- bit 1: at least one pulse contributing to the hit is not OK.
- bit 2: hit at the border of the pad plane.
- bit 3 - 19: not used for the time being.
- bit 20 - 30: reserved for private usage.
- bit 31: not used for the time being

Existing reconstruction code can sometimes be more specific than that. What comes to my mind is:

- contains a dead pad
- contains an overflow pulse
- contains a pulse that has been split
- contains a pulse with strange shape
- is at the edge/border

Did I forget something? I would propose that the most usual defects get a bit assigned and only the least part is undefined.

Cheers,
Christoph

Subject: Re: LCIO::EVENT::TrackerHit quality information
Posted by [killenberg](#) on Wed, 03 Mar 2010 20:17:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Christoph,

your proposal looks good, but I have some comments:

contains a dead pad = bit 0: contains at least one broken channel

contains an overflow pulse

contains a pulse that has been split

contains a pulse with strange shape

These three obsolete bit 1: at least one pulse contributing to the hit is not OK. Or do you have anything else that could be wrong with a pulse?

is at the edge/border = bit 2: hit at the border of the pad plane.

I also have some missing items:

channel is noisy

contains non-dead channel without pulse

is multi hit candidate (for instance contains local minimum in charge distribution, or bit is set by topological hit/track finder, which is planned when track candidates are split)

Should we reserve a sequence of bits to store the number of possible hits that contributed?

With this the new proposal looks like:

bit 0: contains at least one broken channel (dead)

bit 1: contains at least one noisy channel

bit 2: hit at the border of the pad plane.

bit 3: contains an overflow pulse

bit 4: contains a pulse that has been split

bit 5: contains a pulse with strange shape

bit 6: contains non-dead channel without pulse

bit 7: is multi hit candidate

bit 8 - 19: not used for the time being.

bit 20 - 30: reserved for private usage.

bit 31: not used for the time being

Cheers
Martin

Subject: Re: LCIO::EVENT::TrackerHit quality information
Posted by [rosemann](#) on Thu, 04 Mar 2010 09:24:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Martin,

I just wanted to collect the ideas first. I guess the different "qualities" lead to different ways of handling, therefore I wanted to have a finer granularity.

An addition: if we want to use a flag for "multiple hit candidate", then we should also have a flag for "is a split hit" (I guess).

Cheers,
Christoph

Subject: LCIO::TRACK type keyword definition
Posted by [rosemann](#) on Mon, 19 Apr 2010 10:52:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi all,

as well as for the TrackerHit and TrackerPulse we need an actual definition of the "getType()" keyword of the Track.

I'm pretty much unaware (at least until now) of the requirements, so I'll stay quiet for now and ask: what are the needed flags for the track?

To start a list:
- helix or straight line

Cheers,
Christoph

Subject: Alignment for TPC Modules
Posted by [rosemann](#) on Mon, 19 Apr 2010 11:06:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

we started a discussion on the topic of alignment in the MarlinTPC meeting of April 15, 2010.

My proposal is to have a baseline geometry description (on the basis of best knowledge at the beginning), encoded in the GEAR xml file.

Then a two-step procedure to

determine the misalignment of the individual modules and store it as alignment constants in the conditions database

apply the alignment during runtime in a second execution on hit level

I see several advantages. We directly have the versioning system of the conditions database, as well as the underlying idea of alignment represented in the right category -- as conditions data. We also stay with the already established reconstruction method in two steps.

Any thoughts, comments?

Please note: for the Hit level alignment, I would suggest an additional flag for the hit (see another forum thread for this).

Cheers,
Christoph

Subject: Re: LCIO::EVENT::TrackerHit quality information
Posted by [rosemann](#) on Mon, 19 Apr 2010 11:07:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

another addition, possibly depending on the outcome of the discussion of "how to do alignment":

- flag if a hit has already been aligned

See this thread for the alignment discussion.

Cheers,
Christoph

Subject: Re: LCIO::TRACK type keyword definition
Posted by [killenberg](#) on Mon, 19 Apr 2010 12:18:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Christoph,

the current TrackFitterBase class defines bit 30 as FITFAILEDBIT. In addition it uses bits 16-23 to store the fitter type (also defined in TrackFitterBase).

I think we also need a flag for double (multi) track candidates. Do we want a flag whether a multi track candidate has been split?

Cheers

Martin

Subject: Re: LCIO::EVENT::TrackerPulse additions
Posted by [killenberg](#) on Mon, 19 Apr 2010 13:24:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Christoph,

There certainly is a correlation between the charge sum and the time since both are calculated from the ADC same values. If for instance the tail is cut off due to a too high threshold the pulse is too short, the time is shifted (at least when using centre of gravity) and the charge sum is too low.

However, I don't think we need a covariance matrix.

For the maximum value one could either store the maximum value and the time of the maximum, or the index of the maximum within the rawdata vector. The latter would only save the looping, not converting sample steps into ns. It depends on the use case which one is more convenient /

efficient.

Btw: What is this needed for? For a normal pulse charge sum and maximum should be proportional. I think for the few pulses which need special treatment we can afford to loop the raw data again.

Cheers

Martin

Subject: Pulse (and Hit) quality
Posted by [killenberg](#) on Thu, 06 May 2010 13:28:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

I am looking for a place where to define the constants used to store the pulse quality. An example which comes with LCIO proposes to store a vector with names and bit positions as collection parameters. I think this is a little bit cumbersome to use, especially as quality can not be queried bitwise.

I would prefer to have it in a header file.

Currently the pulse finder uses 1, 10 and 100 as constants, which is better to read in a data dump than setting bits. But I find it difficult to retrieve the flags in a programme. Here setting individual bits is better.

If we have less than 8 flags one could use multiples of 16 as flags and use hex output as dump, which is also easy to read.

So here is my proposal:

Header file:

```
namespace marlintpc{

  struct PulseQuality{
    static int const MULTI_HIT_CANDIDATE = 1;
    static int const ANOMALOUS_SHAPE    = 1<<4;
    static int const OVERFLOW          = 1<<8;
    static int const WAS_SPLIT         = 1<<12;
  };
}
```

```
}
```

Example code:

```
#include "PulseQuality.h"
```

```
int quality=0;
```

```
// set the overflow bit
```

```
quality |= PulseQuality.OVERFLOW;
```

```
// delete the anomalous shape bit
```

```
quality &= ~PulseQuality.ANOMALOUS_SHAPE;
```

```
// check whether "was split" is set
```

```
if (quality & PulseQuality.WAS_SPLIT) cout << "Pulse was split!<< endl;
```

```
// dump the quality word
```

```
cout << "Quality word is 0x" << hex << quality << dec<< endl;
```

```
// as overflow is set this prints 0x100
```

What do you think about it? One could also use a namespace instead of a struct and use `pulsequality::OVERFLOW`, or just `OVERFLOW` and 'using namespace'.

Cheers

Martin

P.S. We also don't have a place for the hit quality names. One bit is defined in the `TrackFitterBase`, but this probably is not a good place.

Subject: Re: Pulse (and Hit) quality

Posted by [rosemann](#) on Thu, 06 May 2010 14:42:50 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Martin,

killenberg wrote on Thu, 06 May 2010 06:28[...]I am looking for a place where to define the constants used to store the pulse quality.

The right way to do it is externally (I had brought this up also in our SW meeting), so that no arbitrary convention is used in the code. I am currently re-writing the sort-of manual, the slightly outdated draft can be found here: TPC data model Draft

Quote:I would prefer to have it in a header file.

I see what you mean, but on the other hand you could simply refer to the outside document wherever you use it in the code. Even if the bit field usage is clear to you (which I guess), anybody that wants to use it can look it up in the same file as quoted above; or in the future document. An older, first idea (I'm behind the updating schedule) is at <http://www-flc.desy.de/flc/flcwiki/MarlinTPCStructure>

Quote:Currently the pulse finder uses 1, 10 and 100 as constants, ...

Sorry, you're right. I am again asynchronous to the trunk version; I had this updated to the most-to-current scheme some time ago (see the version in my branch).

Quote:But I find it difficult to retrieve the flags in a programme.[...]

I agree, but any scheme that makes the reading nicer still only relays the fact that these choices of numbers are arbitrary.

If we use this eye candy encapsulation, I would rather use one single class/object that handles all of these definitions, rather than different ones for the different objects.

In case you are not in urgent need of a solution, I would ask you to bear a little longer with the current setup; until we have a full list of these "arbitrary numbers" and then make a single class out of it. Please don't do the definition of numbers in header files; rather use the initialisation list.

To round up the information, there are three different threads related to the pulse quality, hit quality and track type definition.

cheers,
Christoph

Subject: Re: Pulse (and Hit) quality
Posted by [killenberg](#) on Thu, 06 May 2010 16:22:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Christoph,

thanks for your reply. I completely agree that the definition should be done externally. But we should also have it implemented as constants because the source code is much clearer if it reads `if (quality & OVERFLOW) {...}` instead of `if (quality & (1<<4)) {...}`

Quote:An older, first idea (I'm behind the updating schedule) is at <http://www-flc.desy.de/flc/flcwiki/MarlinTPCStructure>

Yes, that's the way I would have implemented it. I just proposed the "eye candy" thing because the 1, 10, 100 looked like it.

Quote:Please don't do the definition of numbers in header files; rather use the initialisation list. Yes, you are right. However, for `static const` I sometimes make an exception because

They are not real variables. I think the compiler "hardcodes" them during the optimisation anyway and does not even allocate memory for them.

You have the variable names and the values in the same place.

Doxygen includes the values.

You don't need a .cpp and object file.

I usually interpret them as the "C++-version" of `#define`

Cheers

Martin

Subject: Questions about fitting in MarlinTPC
Posted by [killenberg](#) on Fri, 07 May 2010 08:28:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

Christoph had a few questions about the track fitting in MarlinTPC and we both think they are of general interest. So I answer them here.

Quote:1.) Is there a "simple" χ^2 fit processor for straight lines?

You can use the `TrackFitterSimpleChiSquareProcessor` and fix the curvature to 0. Use the parameters `OmegaStart=0` and `FixOmega=true`.

An other alternative is the `LinearRegressionProcessor`. The linear regression is equivalent to a χ^2 minimisation with all errors being 1.

Quote:2.) A χ^2 helix fitter seems to be implemented in TrackFitterSimpleChiSquareProcessor. Why are defocussing and diffusion parameters of the fit? To fit a function one only needs the functions, its parameters and errors. Why are there other parameters, which are not part of the fit? The SimpleChiSquare fitter does not use the errors of the hits (for instance to be able to run with the current hit finders which do not set the errors. At least the TopoFinder does not). In a first version all errors were set to 1, but it turned out that the errors in xy and z can be very different, and they both depend on z.

The SimpleChiSquare fitter estimates the errors as

$$\begin{aligned}\sigma_{xy} &= \sqrt{\text{TransDefocussing}^2 + \text{TransDiffusionCoef}^2 * z} \\ \sigma_z &= \sqrt{\text{LongDefocussing}^2 + \text{LongDiffusionCoef}^2 * z}\end{aligned}$$

where z is the drift distance. The names are misleading. Originally the idea was to use defocussing and diffusion coefficients from the conditions data. But it turned out that calculating the required values from them is not straight forward. So "Defocussing" is the intrinsic detector resolution at zero drift, and "Diffusion" is the drift distance (diffusion) dependent part of the resolution.

The documentation is definitely wrong, and we probably should also change the names to be clear.

Quote:

3.) Is it correct that neither dEdx (and its error) nor the covariance matrix are stored, or not even calculated?

Unfortunately yes. dEdx is not calculated at all. The covariance matrix is available in the Minuit fit, but it is not stored. The fitter is still alpha and got stuck in the phase when I was struggling that the fit converged at all. In the end it turned out that there was a problem with the start parameters and the pattern recognition (track finding). So the fit should converge, but I did not have the time to finish it yet.

Quote:

4.) The base class / base fitter is TrackFitterSimpleChiSquare. Is it correct that some things are not cleanly implemented, i. e. hard-coded?

The actual base class is TrackFitterBase. This implements a generic version of calculateResiduals() (residual = distance perpendicular to the helix). TrackFitterSimpleChiSquare inherits from this and implements the fitting function. TrackFitterSimpleChiSquarePads inherits the fitter function from TrackFitterSimpleChiSquare, but

reimplements calculateResiduals() to return the residuals along the pad row.

Minuit needs limits for the track parameters to converge reliably. These are currently hard-coded:

omega=[-1 .. 1] (only tracks with radius larger 1 mm)
D0=[-100 .. 100] (impact parameter +- 100 mm)
phi=[-2 pi .. 2 pi] (two full circles, no real limit)
tanLambda=[-100 .. 100] (only tracks with less than 100 cm z variation on 1
cm on the xy projection)
D0=[-1000 .. 1000] (only tracks with Z0 +- 1 m)

The only problematic value is D0, since this is a sort of "vertex constraint". In prototype geometry D0 can be large, depending on the orientation of the readout module in global coordinates. +- 1 m in z is fine for the LP, but still hard coded is not good. How about using +- 100 mm around the start value from the seed track, for both D0 and Z0?

The other values should be no real limitation, although it should be in the documentation in case someone has problems in exotic cases.

Cheers

Martin

Subject: Questions about fitting in MarlinTPC
Posted by [killenberg](#) on Fri, 07 May 2010 08:30:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

Christoph had a few questions about the track fitting in MarlinTPC and we both think they are of general interest. So I answer them here.

Quote:1.) Is there a "simple" chi² fit processor for straight lines?

You can use the TrackFitterSimpleChiSquareProcessor and fix the curvature to 0. Use the parameters OmegaStart=0 and FixOmega=true.

An other alternative is the LinearRegressionProcessor. The linear regression is equivalent to a chi² minimisation with all errors being 1.

Quote:2.) A chi² helix fitter seems to be implemented in TrackFitterSimpleChiSquareProcessor. Why are defocussing and diffusion parameters of the fit? To fit a function one only needs the

functions, its parameters and errors. Why are there other parameters, which are not part of the fit? The SimpleChiSquare fitter does not use the errors of the hits (for instance to be able to run with the current hit finders which do not set the errors. At least the TopoFinder does not). In a first version all errors were set to 1, but it turned out that the errors in xy and z can be very different, and they both depend on z.

The SimpleChiSquare fitter estimates the errors as

$$\begin{aligned}\sigma_{xy} &= \sqrt{\text{TransDefocussing}^2 + \text{TransDiffusionCoef}^2 * z} \\ \sigma_z &= \sqrt{\text{LongDefocussing}^2 + \text{LongDiffusionCoef}^2 * z}\end{aligned}$$

where z is the drift distance. The names are misleading. Originally the idea was to use defocussing and diffusion coefficients from the conditions data. But it turned out that calculating the required values from them is not straight forward. So "Defocussing" is the intrinsic detector resolution at zero drift, and "Diffusion" is the drift distance (diffusion) dependent part of the resolution.

The documentation is definitely wrong, and we probably should also change the names to be clear.

Quote:

3.) Is it correct that neither dEdx (and its error) nor the covariance matrix are stored, or not even calculated?

Unfortunately yes. dEdx is not calculated at all. The covariance matrix is available in the Minuit fit, but it is not stored. The fitter is still alpha and got stuck in the phase when I was struggling that the fit converged at all. In the end it turned out that there was a problem with the start parameters and the pattern recognition (track finding). So the fit should converge, but I did not have the time to finish it yet.

Quote:

4.) The base class / base fitter is TrackFitterSimpleChiSquare. Is it correct that some things are not cleanly implemented, i. e. hard-coded?

The actual base class is TrackFitterBase. This implements a generic version of calculateResiduals() (residual = distance perpendicular to the helix). TrackFitterSimpleChiSquare inherits from this and implements the fitting function. TrackFitterSimpleChiSquarePads inherits the fitter function from TrackFitterSimpleChiSquare, but reimplements calculateResiduals() to return the residuals along the pad row.

Minuit needs limits for the track parameters to converge reliably. These are currently hard-coded:

omega=[-1 .. 1] (only tracks with radius larger 1 mm)
D0=[-100 .. 100] (impact parameter +- 100 mm)
phi=[-2 pi .. 2 pi] (two full circles, no real limit)
tanLambda=[-100 .. 100] (only tracks with less than 100 cm z variation on 1
cm on the xy projection)
D0=[-1000 .. 1000] (only tracks with Z0 +- 1 m)

The only problematic value is D0, since this is a sort of "vertex constraint". In prototype geometry D0 can be large, depending on the orientation of the readout module in global coordinates. +- 1 m in z is fine for the LP, but still hard coded is not good. How about using +- 100 mm around the start value from the seed track, for both D0 and Z0?

The other values should be no real limitation, although it should be in the documentation in case someone has problems in exotic cases.

Cheers

Martin

Subject: Re: Questions about fitting in MarlinTPC
Posted by [rosemann](#) on Fri, 07 May 2010 15:48:41 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Martin,

several things come to my mind.

killenberg wrote on Fri, 07 May 2010 01:28Quote:1.) Is there a "simple" χ^2 fit processor for straight lines?

You can use the TrackFitterSimpleChiSquareProcessor and fix the curvature to 0. Use the parameters OmegaStart=0 and FixOmega=true.

An other alternative is the LinearRegressionProcessor. The linear regression is equivalent to a χ^2 minimisation with all errors being 1.

I think we should better do this again. Of course it's possible to write the fitter in a way that it can handle both kinds of model (straight line and helix). But it is probably more suggestive to use the processor name to specify it's type. And then use the (still to be defined) track type of the candidate to determine, whether the fitter can handle this kind of track.

Moving on to errors:

killenberg

Quote:2.) A χ^2 helix fitter seems to be implemented in TrackFitterSimpleChiSquareProcessor. Why are defocussing and diffusion parameters of the fit? To fit a function one only needs the functions, its parameters and errors. Why are there other parameters, which are not part of the fit? The SimpleChiSquare fitter does not use the errors of the hits (for instance to be able to run with the current hit finders which do not set the errors. At least the TopoFinder does not). In a first version all errors were set to 1, but it turned out that the errors in xy and z can be very different, and they both depend on z.

The SimpleChiSquare fitter estimates the errors as

$$\begin{aligned}\sigma_{xy} &= \sqrt{\text{TransDefocussing}^2 + \text{TransDiffusionCoef}^2 * z} \\ \sigma_z &= \sqrt{\text{LongDefocussing}^2 + \text{LongDiffusionCoef}^2 * z}\end{aligned}$$

where z is the drift distance. The names are misleading. Originally the idea was to use defocussing and diffusion coefficients from the conditions data. But it turned out that calculating the required values from them is not straight forward.

So "Defocussing" is the intrinsic detector resolution at zero drift, and "Diffusion" is the drift distance (diffusion) dependent part of the resolution.

The documentation is definitely wrong, and we probably should also change the names to be clear.

The "RowBasedHitFinder" calculates the right covariance matrix for the hits [actually I'm implementing this right now, since the z error was missing information from the pulse level -- which is committed to the LCIO head already].

Actually I can't really follow your idea of parametrising x and y errors in terms of drift distance z. The uncertainty of the position measurement (x,y,z) is determined by the reconstruction algorithm, since these values are computed by it.

killenberg

Quote:

3.) Is it correct that neither dEdx (and its error) nor the covariance matrix are stored, or not even calculated?

Unfortunately yes. dE_{dx} is not calculated at all. The covariance matrix is available in the Minuit fit, but it is not stored. The fitter is still alpha and got stuck in the phase when I was struggling that the fit converged at all. In the end it turned out that there was a problem with the start parameters and the pattern recognition (track finding). So the fit should converge, but I did not have the time to finish it yet.

OK. Seems to me, that we possibly should rework the fitting scheme a little. This might turn into a much larger thread, to sort this out -- but I think we should get this right. I'm still a bit puzzled about the approach to the "fitting problem", and if it is a good way like it is now. I'm treading in dangerous waters here, since you obviously invested both time, thought and work into the existing scheme. But maybe we should start with a more pragmatic approach, from which we could draw conclusions about a more generic ansatz.

Concerning the parameter error calculation -- this directly comes out of the minimising program, so we would need an interface to it. I don't see right away if there is a simple solution. It might be possible, but thinking straight has become a little harder recently

I hope, I've got this right -- but the calculation of the dx should not be too complicated, even for a helix.

In my opinion, the only really unfortunate part comes in, when we associate hits (to identify the charge) with dE . In this case we would need geometrical information about the readout plane layout again; at least for row/pad based schemes. (Maybe I shouldn't try to write this at Friday evening, hopefully my idea comes across)

killenberg

Quote:

4.) The base class / base fitter is `TrackFitterSimpleChiSquare`. Is it correct that some things are not cleanly implemented, i. e. hard-coded?

The actual base class is `TrackFitterBase`. This implements a generic version of `calculateResiduals()` (residual = distance perpendicular to the helix).

`TrackFitterSimpleChiSquare` inherits from this and implements the fitting function.

`TrackFitterSimpleChiSquarePads` inherits the fitter function from `TrackFitterSimpleChiSquare`, but reimplements `calculateResiduals()` to return the residuals along the pad row.

Correct me, if I'm wrong. In principle the whole base class idea is to have this as a generic interface, to provide a common name for the "residual" and "distance" access (I use the naming of Ralf and Matthias, hopefully it is known beyond our borders...). Eventually the current naming

scheme for this "track with/without hit" - thing is not so good. I don't know if there is any convention concerning this. But I think using the same name, but different arguments to distinguish is not a good idea:

```
[code title=TrackFitterBase.h]
```

```
/** Return the distance of the hit \c testHitNumber of the testTrack  
 * to the point of closest approach on the referenceTrack wrt.\ the  
 * hit.
```

```
[...]
```

```
*/  
virtual EVENT::DoubleVec calculateResiduals(EVENT::Track *testTrack,  
                                             unsigned int testHitNumber,  
                                             EVENT::Track *referenceTrack = NULL);
```

```
/** Return the distance of the given hit to the test track  
 * where this hit has been removed.  
 * The residual is calculated as the distance  
 * to the point of closest approach on the referenceTrack wrt.\ the hit.
```

```
*/  
virtual EVENT::DoubleVec calculateResiduals(EVENT::TrackerHit *testHit,  
                                             EVENT::Track *trackWithoutTestHit);
```

```
[/code]
```

And besides the names -- has there ever been a discussion about the definition? I would think, that this is one of the common topics, that must have arisen in result discussions before. So I would somehow think, that we should have the standard definition in -- and everything else is up to the "user". Having different meanings by processor name seems not a very good idea to me.

And finally the hard-coded stuff:

killenberg

Minuit needs limits for the track parameters to converge reliably. These are currently hard-coded:

$\omega = [-1 .. 1]$ (only tracks with radius larger 1 mm)
 $D_0 = [-100 .. 100]$ (impact parameter ± 100 mm)
 $\phi = [-2\pi .. 2\pi]$ (two full circles, no real limit)
 $\tan\lambda = [-100 .. 100]$ (only tracks with less than 100 cm z variation on 1 cm on the xy projection)
 $D_0 = [-1000 .. 1000]$ (only tracks with $Z_0 \pm 1$ m)

The only problematic value is D_0 , since this is a sort of "vertex constraint". In prototype geometry

D0 can be large, depending on the orientation of the readout module in global coordinates. +/- 1 m in z is fine for the LP, but still hard coded is not good. How about using +/- 100 mm around the start value from the seed track, for both D0 and Z0?

The other values should be no real limitation, although it should be in the documentation in case someone has problems in exotic cases.

I somehow still have a little bit of trouble with Minuit working this way. If both the track finding works reliably and the hit errors are reasonably well described the fit should converge. We could think at some point about outlier rejection, downweighing, blabla. But eventually Chi² fitting is not a heavily complicated problem. In principle it comes down to matrix inversions and a few matrix multiplications. I can have a closer look into that part.

Cheers,
Christoph

Subject: Re: Questions about fitting in MarlinTPC
Posted by [killenberg](#) on Mon, 10 May 2010 15:18:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Christoph,

Quote:

I think we should better do this again. Of course it's possible to write the fitter in a way that it can handle both kinds of model (straight line and helix).

The chi² fitter just happens to be able to do straight lines. For some studies it is necessary to fix the curvature and set it to a specific value, so I implemented this. If one sets it to 0 it's a straight line. I totally agree that we should have a dedicated straight line processor.

Quote:

The "RowBasedHitFinder" calculates the right covariance matrix for
This is what I expected. So it's time we get a proper implementation of the chi² fitter.

Quote:

Actually I can't really follow your idea of parametrising x and y errors in terms of drift distance z. The uncertainty of the position measurement (x,y,z) is determined by the reconstruction algorithm, since these values are computed by it.

If you plot the residuals in xy and z against z this is the parametrisation you get. So it's an a

posteriori estimate for the mean error (if the individual errors are not available). since the individual errors are there now we definitely should use them.

Quote:

Seems to me, that we possibly should rework the fitting scheme a little.

Yes, we should to that. I am also not happy with it. I think it's too complicated and tangled. I discussed with several people but could not come up with something simpler that provided the functionality I wanted. We should discuss it in a separate thread.

Quote:

Concerning the parameter error calculation

It should only be a few lines to read the covariance matrix from Minuit and fill it into the track. I admit I was just lazy looking it up when I still had problems with the convergence.

The same for dEdx: One just has to write a few lines, no principle problem. Maybe one can put a generic calculation into the base class so one does not have to copy it for all the implementations.

Quote:

I somehow still have a little bit of trouble with Minuit working this way. If both the track finding works reliably and the hit errors are reasonably well described the fit should converge. We could think at some point about outlier rejection, downweighing, blabla.

Yes, once the pattern recognition works Minuit converges fine. And we definitely should have an outlier rejection. Maybe just do a double pass fit?

Cheers

Martin

Subject: Re: Questions about fitting in MarlinTPC
Posted by [killenberg](#) on Tue, 11 May 2010 06:57:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Christoph,

Quote:

I think we should better do this again. Of course it's possible to write the fitter in a way that it can handle both kinds of model (straight line and helix).

The χ^2 fitter just happens to be able to do straight lines. For some studies it is necessary to fix the curvature and set it to a specific value, so I implemented this. If one sets it to 0 it's a straight line. I totally agree that we should have a dedicated straight line processor.

Quote:

The "RowBasedHitFinder" calculates the right covariance matrix for
This is what I expected. So it's time we get a proper implementation of the χ^2 fitter.

Quote:

Actually I can't really follow your idea of parametrising x and y errors in terms of drift distance z. The uncertainty of the position measurement (x,y,z) is determined by the reconstruction algorithm, since these values are computed by it.

If you plot the residuals in xy and z against z this is the parametrisation you get. So it's an a posteriori estimate for the mean error (if the individual errors are not available). since the individual errors are there now we definitely should use them.

Quote:

Seems to me, that we possibly should rework the fitting scheme a little.

Yes, we should to that. I am also not happy with it. I think it's too complicated and tangled. I discussed with several people but could not come up with something simpler that provided the functionality I wanted. We should discuss it in a separate thread.

Quote:

Concerning the parameter error calculation

It should only be a few lines to read the covariance matrix from Minuit and fill it into the track. I admit I was just lazy looking it up when I still had problems with the convergence.

The same for dEdx: One just has to write a few lines, no principle problem. Maybe one can put a generic calculation into the base class so one does not have to copy it for all the implementations.

Quote:

I somehow still have a little bit of trouble with Minuit working this way. If both the track finding works reliably and the hit errors are reasonably well described the fit should converge. We could think at some point about outlier rejection, downweighing, blabla.

Yes, once the pattern recognition works Minuit converges fine. And we definitely should have an outlier rejection. Maybe just do a double pass fit?

Cheers

Martin

Subject: Fitting scheme

Posted by [rosemann](#) on Tue, 11 May 2010 15:33:16 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi,

after giving it some thought, I would like start the discussion about the current and future scheme of fitting in MarlinTPC.

Probably Martin (are you there?) will comment on this, but this represents my understanding of the current scheme:

The base class is "TrackFitterBase", which basically just defines an interface.

CR: I'm not too happy about the naming scheme for the /residuals/ access, but in principle I think it is OK to have something like this. It forces to implement some functions, but I can't see the disadvantage of this.

The actual fitter class are derived (inherited) from the above base class and are not Marlin processors.

CR: I can agree to both. We would like to call the fit iteratively. With processors it is believed to be possible (I talked to Steve about "processors calling processors"), but I don't think we should make life more complicated.

CR: As a different thought I eventually think that a much simpler "fit object" might be more natural. I would throw the fit object the LCIO::TrackerHits and deal with the result.

The calling of the fitter class is done by a processor.

CR: This seems obvious, since we want to do it within MarlinTPC. But I guess we might want to do the "bookkeeping" part in the processor, rather than in the fitter class.

The TrackFitterFactory can be used to handle different fitted tracks by their type; by using slightly peculiar "types" defined in the TrackFitterBase class.

CR: I have to admit that this is a possible solution to the problem: "how do I determine the residuals of a particular already fitted track". But it is not the most straightforward (at least to me).

---- BREAK ----

I would approach this whole problem slightly different. I would start with the description of what we want to do, then try to write it and think about how we might want to use it differently/figure out what is missing.

What do we want do?

- fit tracks on already found track hypotheses with associated hits
- calculate residuals (and/or distance) to obtain resolutions
- reject outliers
- ...

I can say that we (in Hamburg) now start(ed) to develop/implement a full fledged (with input and output errors) chi square fitter for a straight line and a helix. Hopefully more ideas will come along the way.

Cheers,
Christoph

Subject: Re: Fitting scheme
Posted by [jabernathy](#) on Tue, 11 May 2010 18:44:26 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello all,

Here are my thoughts a fitting scheme:

I like the idea of a TrackFitterBase. From experience, trying to put all of the functionality in one class can be unwieldy.

However, after that it is more complicated.

As it was pointed out before, there are different methods (and meanings) of the residuals. One has to be careful that the correct routine is called for a fitted track.

One idea that I had previously was to let the fitting processor be responsible for calculating the residuals. For example, a reconstruction steering file could contain this processor:

```
<processor name="MyTrackFitterLikelihood" type="TrackFitterLikelihoodProcessor">
<parameter name="Sigma0" type="double">0.81013581</parameter>
<parameter name="FitSigma0?" type="boolean">true</parameter>
<parameter name="Noise" type="double">0.001</parameter>
<parameter name="SigmaZ" type="double">0.5</parameter>
<parameter name="ProduceTrackTuples?" type="bool">true</parameter>
<parameter name="PathIntegralStepSize" type="double">25</parameter>
<parameter name="OmegaTau" type="double">10</parameter>
<parameter name="DriftVelocity" type="double">35</parameter>
<parameter name="ReadoutElectronicsInnerZ" type="double">2245</parameter>
<parameter name="TPCOuterRadius" type="double">1800</parameter>
<parameter name="FitNonHomogenousFields" type="bool">true</parameter>
</processor>
```

Then the analysis code could run the same processor except with a flag telling the fitter to only calculate the residuals:

```
<processor name="MyTrackFitterLikelihood" type="TrackFitterLikelihoodProcessor">
<parameter name="CalculateResiduals">true</parameter>
...
</processor>
```

One advantage to using this method is that the fitter could check the track to see if it was fit using this method, and quit gracefully, if it wasn't. It also allows most of the code to be kept in one place. Note, I am not suggesting that the processor `_is_` the fitter, they could still use a sub-class to manage the fitting and residual calculations.

One disadvantage of this method is that I don't think analysis logic should be placed in the fitter. If the residual calculation is done with the fitting processor then it is probably going to be an "all-or-nothing" choice of which tracks to calculate residuals for. This may not be a bad thing - a `_real_` analysis processor could just ignore any unneeded information.

Another method that I thought of was keeping the track fitter processors in pairs: one for fitting and one for residuals. So there would be a `TrackFitterLikelihoodProcessor` in the reconstruction chain and `TrackResidualLikelihoodProcessor` in the analysis chain. This may not be a bad idea. It allows logic to be located where it needs to be. Code could be shared by using the common object inherited from the `TrackFitterBase`.

I hesitate to suggest using a factory method for calculating the residuals because it seems to be moving in an orthogonal direction to the intent of Marlin - keeping the chain "modularized". Also, if a factory method/processor is used then it could be difficult to manage the processor parameters because each residual method is parameterized differently.

However, I must admit, I don't know a great deal about the analysis side of things.

Is it common for a objects to be reconstructed with one method and then fit with another?

When analysis is done is it usually done to all of the tracks or just ones which are interesting?

Subject: Re: Fitting scheme

Posted by [rosemann](#) on Wed, 12 May 2010 11:52:39 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Jason,

jabernathy[...]

I like the idea of a TrackFitterBase. From experience, trying to put all of the functionality in one class can be unwieldy.

However, after that it is more complicated.

Yes, I think one base class is what we actually want and need; to simplify the usage down the road.

jabernathy

As it was pointed out before, there are different methods (and meanings) of the residuals. One has to be careful that the correct routine is called for a fitted track.

Yes, I agree. I'm heavily influenced by the terminology of the DESY group, so I suggest this naming scheme:

residual -- the geometrical distance in x/ϕ at constant y/r between an associated hit (from track finding) to its fitted track, when it is excluded from the fit

distance -- the geometrical distance in x/ϕ at constant y/r between an associated hit (from track finding) to its fitted track, when it is included from the fit

The resolution is then the geometric mean of both values. (For reference check e.g. Ralf Dieners

thesis or physics/0402054)

So my proposal would be to call the two functions in the TrackFitterBase, including a third one (which is optional, but might be more interesting down the line)

```
calculateDistances(...)  
calculateResiduals(...)  
calculateResolution(...)
```

We would still need some kind of definition or interface to express the alternatives, e.g. perpendicular distances to the track (although this can get pretty ugly).

By doing this in the actual fitter class we would hide the iterative fitting calls (which are needed for the residuals calculation) within this code. I'm not really aware if there are any limitations to the actual fit method. To be more specific, I don't know if the Likelihood method can work like this.

jabernathy

One idea that I had previously was to let the fitting processor be responsible for calculating the residuals.

[...]

One advantage to using this method is that the fitter could check the track to see if it was fit using this method, and quit gracefully, if it wasn't. It also allows most of the code to be kept in one place. Note, I am not suggesting that the processor `_is_` the fitter, they could still use a sub-class to manage the fitting and residual calculations.

One disadvantage of this method is that I don't think analysis logic should be placed in the fitter. If the residual calculation is done with the fitting processor then it is probably going to be an "all-or-nothing" choice of which tracks to calculate residuals for. This may not be a bad thing - a `_real_` analysis processor could just ignore any unneeded information.

I think it is not a good idea to let the processor handle the logic, for several reasons. The extra overhead needed for the determination of the fit method is not large, but somehow confusing. As you mention, several things would get mixed by doing it this way.

The underlying problem is of course something else -- the LCIO objects lack some, eventually fundamental functionality. For example, if it was possible to attach a hit to track, but mark it "unused", then we would have the solution. This would encode all information from the fitting inside the data object. It would also go beyond the scope of fitting.

Probably I will have to take this issue to the LCIO maintainers. In principle this would be the solution to a lot of the problems we have there right now.

The next is more like a comment:

jabernathy[...]

Is it common for a objects to be reconstructed with one method and then fit with another?

When analysis is done is it usually done to all of the tracks or just ones which are interesting?

a) what do you mean by "reconstruction method" ?

b) yes and no. You usually have a lot of requirements for tracks (and its components), and you usually only take a fraction of all tracks that passed these requirements into account. (Something like: χ^2 value, number of hits, position, hit quality, track angle, environmental factors, ...)

Cheers,
Christoph

Subject: Re: Fitting scheme

Posted by [jabernathy](#) on Wed, 12 May 2010 17:03:58 GMT

[View Forum Message](#) <> [Reply to Message](#)

rosemann wrote on Wed, 12 May 2010 04:52Hi Jason,

So my proposal would be to call the two functions in the TrackFitterBase, including a third one (which is optional, but might be more interesting down the line)

```
calculateDistances(...)  
calculateResiduals(...)  
calculateResolution(...)
```

In fact, it is even more complicated than that. The "residuals" from the likelihood method are in units of charge or electrons.

rosemannI think it is not a good idea to let the processor handle the logic, for several reasons. The extra overhead needed for the determination of the fit method is not large, but somehow confusing. As you mention, several things would get mixed by doing it this way.

I agree, this is what led me to the idea of having pairs of fitting processors: one for fitting and one for doing the analysis.

rosemannThe underlying problem is of course something else -- the LCIO objects lack some, eventually fundamental functionality. For example, if it was possible to attach a hit to track, but mark it "unused", then we would have the solution. This would encode all information from the fitting inside the data object. It would also go beyond the scope of fitting.

There is an LCRelation from LCIO. I believe these are used to abstractly associate two LCIO objects with a single weighting parameter.

Do you mean marking a track as unused for the purpose of calculating the "distance" measure that was previously defined?

If that is the case then this would be a good motivator for using the dual-class approach. The algorithm for calculation these distances could be contained in a "TrackResidualChiSquaredProcessor".

"rosemann"a) what do you mean by "reconstruction method" ?

The type of fitter used. So, is it likely that multiple types of track fitters will be used to reconstruct/analyse the data during one execution of Marlin processor chain?

Subject: Re: Fitting scheme
Posted by [killenberg](#) on Fri, 14 May 2010 15:21:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Christoph,
hello Jason,

please excuse my delayed reply.

rosemann
residual -- the geometrical distance in x/ϕ at constant y/r between an associated hit (from track finding) to its fitted track, when it is excluded from the fit distance -- the geometrical distance in x/ϕ at constant y/r between an associated hit (from track finding) to its fitted track, when it is included from the fit

The resolution is then the geometric mean of both values. (For reference check e.g. Ralf Dieners thesis or physics/0402054)

I was not aware that residual and resolution are used with this definition at Desy. Now I understand why you don't like it the way it is used in the current scheme. Here is how I used it:

residual -- The geometric distance (calculated by a method defined by the actual fitter implementation) of a 3D space point (hit) to a track hypothesis (=track with a certain set of parameters).

This is much more general than the definition you use. Especially with the usage of the term "resolution" I feel a little uncomfortable. For me resolution is the width of the residual distribution, for example the RMS of a histogram, not a value of a single measured hit.

But also restricting the measured distance to x/ϕ at constant y/r is not general enough:

Hit coordinates are stored in global coordinates. The pads are in local coordinates. For the LP for instance you hit one module in each module row. So you have hits from three different local polar coordinate systems, which are stored in global Cartesian coordinates.

For the Timepix there is no bias of a pad row, the hits are just measurements of x and y . One wants to measure the distance perpendicular to the helix.

The geometric mean is used because it is an unbiased estimator for the resolution. But we always used the mean of the widths of the two distributions (with and without the specific hit used in the fit). I don't know which is correct, probably both methods are equivalent if you calculate it through.

However, there are other methods to calculate an unbiased estimator. For instance one can use the distance calculated with the hit included in the track fit and multiply it by $\sqrt{n} / \sqrt{n-\text{DoF}}$, where n is the number of hits in the track and DoF is the number of degrees of freedom in the track fit.

Another method is to use an unbiased track estimate, for instance from a separate measurement with a hodoscope. Or using MC truth in simulation.

This is why the interface has two functions:

```
EVENT::DoubleVec TrackFitterBase::calculateResiduals(EVENT::TrackerHit *testHit,  
EVENT::Track *trackWithoutTestHit);  
IMPL::TrackImpl * fitTrack (EVENT::Track const *seedTrack);
```

The first function calculates the residuals to the track that has been provided. The naming of the variable should definitely been changed, it is somewhere between misleading and wrong. How about "trackHypothesis"? There is also a calculateResiduals function which takes the track and the hit number on the track. It was intended to be a convenience function which calls the other one. But I see that it probably causes only confusion.

The fitTrack does what Christoph wants to do with the simpler "fit object". You put in an seed track (probably also not a good name since it is not a seed for pattern recognition) which serves as a container for the hits and provides start parameters for a numerical minimisation. And you get out the fitted track.

To avoid code duplication a χ^2 fitter will call calculateResiduals() during the minimisation. And the histogramming processor calls it to get the residual values.

In addition, once the χ^2 algorithm is implemented one can inherit from it and just change the calculateResiduals (for instance from 'perpendicular to the helix' to 'along the pad row').

A Likelihood fitter does not use the geometrical residuals in the fit process. Here it behaves like Jason's pair proposal: One function for fitting, one for residual calculation.

I always considered the decision which definition of resolution (in the sense of width of residual distribution) to use as part of the analysis. This is why there are several histogramming processors for different definitions. The processor for geometric mean resolution for instance loops all hits, removes the hit from the track, refits it and then retrieves the residual with and without the hit included in the fit. It has to be done 'manually' and is not in the fitter interface.

According to the factory:

I was looking for a way to have only one histogramming processor for each "resolution" definition which works independently from the residual implementation (i.e. track fitter implementation) that was used. The Japanese group proposed that one possibly would use different fitters depending on the track properties (I can imagine using a different fit for particles coming from the vertex). So I would not store the fitter type as a collection

parameter but as a number in the track itself (bits 16-23 in the track type word). So every fitter implementation needed a unique ID, which is the peculiar list of types in the base class.

As the geometric mean processor does refitting and some implementations need additional parameters for this, the factory has an interface which takes LCPParameters. The fitter processor stores these parameters in the tracks collection, so you pass its collectionParameters to the factory, together with the ID you retrieved from the track.

A combinatorial Kalman filter might not only want to reject hits but also to add hits that were not associated to the track. To have access to the hits collection (or whatever collection might be needed) a pointer to the event can be passed.

The factory reads out the implementation specific parameters and calls the constructor of the fitter. For every new fitter implementation the factory has to be extended. This is the price you have to pay to have the histogramming processor independent from the fitter implementation.

I admit the scheme is complicated and a bit tangled, but I could not come up with a simpler solution for the functionality I wanted.

Cheers

Martin

Subject: Re: Conditons Data for MarlinTPC
Posted by [jabernathy](#) on Sat, 15 May 2010 00:47:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

What about adding a parameter for gain in the ChannelCorrection class?

This would allow a consistent method for doing channel-by-channel gain correction in the GainCorrectorProcessor.

Subject: TPC Tracks not being produced
Posted by [Ryan_Page](#) on Tue, 18 May 2010 14:10:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

When running the LEPTracking processor I am getting this printed to screen:

TPCFND: No TPC hits

This is coming from trkfnd.F which is located in MarlinReco/HEAD/Tracking/BrahmsTracking/src/f77. I running software according to ilcinstall 01-08-01, except MarlinReco where I am using the HEAD version, as the release version was causing the TPCDigi to crash. Has any one else had this problem?

Thanks

Ryan

Subject: Re: TPC Tracks not being produced
Posted by [aplin](#) on Tue, 18 May 2010 14:53:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Ryan,

could you provide me a bit more info, are there any TPCHits in the REC file, either SimTrackerHits or TrackerHits?

try anajob on the file to see.

Did you see this problem with the original TPCDigiprocessor in v01-08-01?

Cheers,

Steve.

Subject: Re: TPC Tracks not being produced
Posted by [Ryan_Page](#) on Tue, 18 May 2010 15:18:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Steve,

These are the collections that are created:

```
-----  
COLLECTION NAME          COLLECTION TYPE          NUMBER OF ELEMENTS  
=====
```

DroppedTPCTrackeHits	TrackerHit	0
ECALBarrel	CalorimeterHit	30
EcalBarrelCollection	SimCalorimeterHit	30
EcalBarrelPreShowerCollection	SimCalorimeterHit	1
HCALBarrel	CalorimeterHit	45
HcalBarrelRegCollection	SimCalorimeterHit	47
LCAL	CalorimeterHit	0
LDCTracks	Track	1
LDCTracksMCP	LCRelation	1
LHCAL	CalorimeterHit	0
MCParticle	MCParticle	3
MUON	CalorimeterHit	13
MuonBarrelCollection	SimCalorimeterHit	13
RelationCaloHit	LCRelation	75
SETCollection	SimTrackerHit	2
SETTrackerHits	TrackerHit	1
SITCollection	SimTrackerHit	2

SITTrackerHits	TrackerHit	2
SiTracks	Track	1
SiTracksMCP	LCRelation	1
TPCCollection	SimTrackerHit	224
TPCTrackerHits	TrackerHit	224
TPCTracks	Track	0
TPCTracksMCP	LCRelation	0
UsedTPCTrackerHits	TrackerHit	224
VTXTrackerHits	TrackerHit	6
VXDCollection	SimTrackerHit	6

The sim hits are digitised to create tracker hits, but then no tracks are created I assume because of the error I see.

I am pretty sure I saw this before however I thought it was due to the digitiser.

Thanks

Ryan

Subject: Re: TPC Tracks not being produced
 Posted by [aplin](#) on Tue, 18 May 2010 20:27:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Ryan,

given that this looks like a single muon event, could you send me the steering file, gear file and the lcio sim file. That way it would be much quicker to find out what is going wrong.

Cheers,

Steve.

Subject: Re: TPC Tracks not being produced
 Posted by [aplin](#) on Thu, 20 May 2010 17:45:26 GMT

Hi Ryan,

I certainly can understand your confusion.

The first thing is that in the events with the problem there are no hits inside the TPC as the muon is traveling in the very forward direction. You can imagine what I was thinking when I realised this

The thing is that LEPTracking really should not be reporting an "error", and causing you undue concern. At this point it seems as if you may have unearthed some very strange behaviour, so the bug report is certainly very welcome.

If you look at the modified output I got when running over your files below, you will clearly see that TPCDigi is reporting that there are no TPCSimtrackerHits and hence no TPCTrackerHits are produce.

The question is why LEPTracking tries to run the tracking given that it has an if statement which checks if the pointer to the TPCHitCollection is NULL. As you can see from the output it is not, which is extremely puzzling ...

```
[ DEBUG "TPCTracking" ] LEPTrackingProcessor called: Pointer to TPCHitCol=0x8b828a0
```

Thanks for the bug report, for now you can just ignore the warning.

Cheers,

Steve.

```
-----  
[ DEBUG4 "MyTPCDigiProcessor" ] finished looping over simhits, number of voxels = 0  
[ DEBUG4 "MyTPCDigiProcessor" ] the number of adjacent hits is 0 _doubleHitResZ 5  
[ DEBUG4 "MyTPCDigiProcessor" ] number of rec_hits = 0  
[ DEBUG4 "MyTPCDigiProcessor" ] finished row hits 0 0  
[ DEBUG4 "MyTPCDigiProcessor" ] _NSimTPCHits = 0  
[ DEBUG4 "MyTPCDigiProcessor" ] _NBackgroundSimTPCHits = 0  
[ DEBUG4 "MyTPCDigiProcessor" ] _NPhysicsSimTPCHits = 0  
[ DEBUG4 "MyTPCDigiProcessor" ] _NPhysicsAbove02GeVSimTPCHits = 0  
[ DEBUG4 "MyTPCDigiProcessor" ] _NPhysicsAbove1GeVSimTPCHits = 0  
[ DEBUG4 "MyTPCDigiProcessor" ] _NRecTPCHits = 0  
[ DEBUG4 "MyTPCDigiProcessor" ] _NLostPhysicsTPCHits = 0  
[ DEBUG4 "MyTPCDigiProcessor" ] _NLostPhysicsAbove02GeVPtTPCHits = 0
```



```
[ DEBUG4 "MyTPCDigiProcessor"] _NLostPhysicsAbove1GeVpTPCHits = 0
[ DEBUG4 "MyTPCDigiProcessor"] _NRevomedHits = 0
[ DEBUG "TPCTracking"] LEPTrackingProcessor called
[ DEBUG "TPCTracking"] Fetching TPCHitCollection TPCTrackerHits
[ DEBUG "TPCTracking"] LEPTrackingProcessor called: Pointer to TPCHitCol=0x8b828a0
[ DEBUG "TPCTracking"] Number of TPCHit before filtering: 0
[ DEBUG "TPCTracking"]
[ DEBUG "TPCTracking"] LEPTrackingProcessor: Trying to remove hits alla CurlKiller
[ DEBUG "TPCTracking"]
[ DEBUG "TPCTracking"] Number of TPCHit after filtering: 0
TPCFND: No TPC hits
[ DEBUG "TPCTracking"] For Event:22TKTREV return:1
[ DEBUG "TPCTracking"] number of TE's = 0
[ DEBUG "TPCTracking"] number of TK's = 0
```

Subject: Re: Fitting scheme (part 1/2 for Jason)
Posted by [rosemann](#) on Wed, 02 Jun 2010 09:26:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Martin and Jason,

I probably need to add several "oops" in my previous mails. I really have to get some things straight; it seems that I misunderstood a couple of things. (And of course: sorry for replying so late, but I really had to think a bit).

First I reply to Jason (I'd rather answer Martin in a second post, so not to confuse any further)

jabernathy wrote on Wed, 12 May 2010 10:03[...]

In fact, it is even more complicated than that. The "residuals" from the likelihood method are in units of charge or electrons.

This makes life a lot more complicated. I suggest that we split the fitting schemes. I don't see so many different methods coming up in the future. So then we have two fundamentally separate ways of fitting, still with hopefully only one base class/interface.

[...]

rosemannThe underlying problem is of course something else -- the LCIO objects lack some, eventually fundamental functionality.[...]

jabernathy

There is an LCRelation from LCIO. I believe these are used to abstractly associate two LCIO objects with a single weighting parameter.

Since there is further evidence that the functionality is needed for tracking in general, I'd rather go with the try to extend the LCIO track class. For the time being it might be better to create some intermediate solution; maybe this is something for the next MarlinTPC meeting.

Ever so slowly I see some things converging on the horizon...

Cheers,
Christoph

Subject: Re: Fitting scheme (part 2/2 for Martin)
Posted by [rosemann](#) on Wed, 02 Jun 2010 09:47:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Martin and Jason,

now come the "oops" parts.

killenberg[...][rosemann](#)

residual -- the geometrical distance in x/ϕ at constant y/r between an associated hit (from track finding) to its fitted track, when it is excluded from the fit distance -- the geometrical distance in x/ϕ at constant y/r between an associated hit (from track finding) to its fitted track, when it is included from the fit

The resolution is then the geometric mean of both values. (For reference check e.g. Ralf Dieners thesis or physics/0402054)

Wrong on my part, the (generic) distance is not at a fixed value, but really the perpendicular distance. While I personally think that this is a small nightmare in reading the formulas (e.g. think about: polar coordinates for the hits and a helix), it needs to be written only once. It seems that I had completely misunderstood something; I believed that the definition of resolution was like I stated in the previous post (taken at fixed row position) -- but it turns out, that there was no such agreement and the actual use (also at DESY) is the perpendicular definition.

I'm still looking, if there ever was a common agreement about this definition; maybe even written down. Otherwise we should somehow specify this (but where?).

killenberg[...]The geometric mean is used because it is an unbiased estimator for the resolution. But we always used the mean of the widths of the two distributions (with and without the specific hit used in the fit). I don't know which is correct, probably both methods are equivalent if you calculate it through.

I'm not sure if I can follow you here. Afaik the geometric mean method is the one described in the paper I mentioned before (physics/0402054).

killenbergHowever, there are other methods to calculate an unbiased estimator. For instance one can use the distance calculated with the hit included in the track fit and multiply it by $\sqrt{n} / \sqrt{n - \text{DoF}}$, where n is the number of hits in the track and DoF is the number of degrees of freedom in the track fit.

I've never seen this way of defining the resolution. This would be a nice way to determine it without needing to re-fit the track $n-1$ times for n hits on the track. Do you have a reference where this is described or shown?

killenbergAnother method is to use an unbiased track estimate, for instance from a separate measurement with a hodoscope. Or using MC truth in simulation. Agreed, but both are academic for the time being. But maybe still this year we will have something like a hodoscope for the LP...(?)

killenberg[skip the longer part of the description]

I admit the scheme is complicated and a bit tangled, but I could not come up with a simpler solution for the functionality I wanted.[...]

I have thought a lot about the scheme and it seems to me that either way it will be some kind of conflict. What I'm doing right now is sort of "start all over" and see where I'm going. The base class is a good idea and I will build on it.

I definitely propose to discuss this in the next MarlinTPC meeting. Maybe I will come to the same conclusion as Martin. Maybe there are other solutions.

I will report my thoughts and findings also here, to allow other thoughts into to process.

Cheers,
Christoph

Subject: Re: Fitting scheme (part 2/2 for Martin)
Posted by [killenberg](#) on Wed, 02 Jun 2010 12:56:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

rosem

killenb... For instance one can use the distance calculated with the hit included in the track fit and multiply it by $\sqrt{n} / \sqrt{n-\text{DoF}}$, where n is the number of hits in the track and DoF is the number of degrees of freedom in the track fit.

...Do you have a reference where this is described or shown?

I use Wikipedia because it is easy to link.

For the 1D case of a Gaussian distribution there is only one free parameter, the mean of the distribution. In this case the factor is $n-1$, known as Bessel's correction. The variance calculated using $n-1$ is the sample variance.

For the 2D case of a linear regression (which is an ordinary least squares estimator) I use the German text book

Lothar Sachs, "Statistische Methoden", Springer Verlag 1979, ISBN 3-540-09226-9, chapter 8.6

In

Frederick James, "Statistical Methods in Experimental Physics, 2nd Edition", World Scientific 2006, ISBN 981-270-527-9, chapter 8.4.1

I find that the method is retrieved for a linear function of the free parameters. (see also German Wikipedia page about linear regression. On the English page I could not find it).

I don't know how well a linear approximation is fulfilled for our helix fit.

However, the n -DoF correction can be turned on in the BiasedResidualsProcessor and we can compare it to the geometric mean method. It uses $\text{DoF} = 3$ for a helix in the $xy/r\phi$ plane (2 for a straight line), and 2 in the sz plane.

Cheers

Martin

Subject: Re: Fitting scheme (part 2/2 for Martin)
Posted by [killenberg](#) on Wed, 02 Jun 2010 13:23:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Christoph,

according the residuals I found on Wikipedia that in general it means "distance of a measured point to the function", in our case the track. In the name residual it is not determined how the function is defined.

I propose to explicitly state which residuals we refer to:

r_n : residual with all hits in the track fit
r_(n-1) : residual with the test hit excluded in track fit
r_geo : geometric mean of r_n and r_(n-1)
r_MC : residual to Monte Carlo truth as track
r_hod : residual to track from hodoscope

Cheers

Martin

Subject: Re: Fitting scheme (part 2/2 for Martin)
Posted by [ralf](#) on Wed, 02 Jun 2010 16:22:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi all

killenberg wrote on Wed, 02 June 2010 06:23
I propose to explicitly state which residuals we refer to:

r_n : residual with all hits in the track fit
r_(n-1) : residual with the test hit excluded in track fit
r_geo : geometric mean of r_n and r_(n-1)
r_MC : residual to Monte Carlo truth as track
r_hod : residual to track from hodoscope

For the residual including and excluding the fit, I don't have much of an opinion. It is just definition how to call it.

I only used the names "distance" and "residual" in my thesis since its easy to write/read and to enunciate - at least a bit easier than with some kind of indices. And this definition spread in the DESY group.

But as I said, it was just a definition...

I would not call the geometric mean "r_geo", since it is the geometric mean of the widths of the

two corresponding distributions. So IMHO, "sigma" would be more appropriate here to avoid confusion.

Or does anyone need the geometric mean of two residuals? At the moment, I don't see for what this could be used.

The rest of proposed naming is good.

CU, Ralf.

Subject: Re: Conditons Data for MarlinTPC
Posted by [ralf](#) on Wed, 02 Jun 2010 16:46:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

jabernathy wrote on Fri, 14 May 2010 17:47:47 What about adding a parameter for gain in the ChannelCorrection class?

This would allow a consistent method for doing channel-by-channel gain correction in the GainCorrectorProcessor.

When checking the class, I also noticed, that it does not seem to be multi-module compatible. If someone needs the amplification gain separately -which I could imagine is the case in the simulation- it would be good to store it there. For the normal combined amplification/electronics gain correction, the calibration factor should be enough. Or am I missing something?

Proposal: I would implement it in a way, that if it isn't set (you don't always know this value), the object returns 1.0, so a multiplication wouldn't change anything. But you could still set the other correction values and use them.

Similar to the pedestals, where the handler returns 0.0 when there is no value, so that the loop can run even when there is no value for a particular channel...

Since I am -slowly- going through the conditions objects anyway, I can look after this.

CU, Ralf.

Subject: Hit errors in MarlinTPC
Posted by [killenberg](#) on Fri, 04 Jun 2010 17:12:57 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

I would like to discuss the implementation of hit errors in MarlinTPC.

Currently the RowBasedHitFinderProcessor is the only one which fills the covariance matrix at all. It uses the variance of the charge distribution, which for my understanding is not the value which should be in there (LCIO documentation: covariance of the position, not of the charge distribution).

If I understand it correctly the covariance matrix is supposed to hold the variances of the residual distributions, which are the errors of the hit position. These errors are definitely required in track fitting.

However, these values cannot so easily be computed from the charge distribution itself. The statistical answer for the error of the mean (σ/\sqrt{n}) does not work because the entries in the bins are not statistically independent measurements of the same value but ADC counts of the measurement of a suppositionally Gaussian shaped charge cloud.

If I assume no gas gain variations, electronics noise and infinite electronics resolution, the only error is the systematic error due to the binning onto the pads (pad response) when using centre of gravity to calculate the coordinates. After applying pad response correction the error is zero.

So the errors are all due to gas gain fluctuations, electronics noise and systematics which require additional parameters that are not in the hit itself. I see two solution to calculate the errors:

Try to evaluate the error from the hit

$$x_{\text{bar}} = \text{sum}(x_i q_i) / \text{sum}(q_i) ;$$

$$\sigma^2 = \text{sum}((d x_{\text{bar}} / d q_i * \Delta q_i)^2)$$

Here everything depends on the knowledge of the errors of the charge measurement on the pads.

Advantages:

Errors are hit dependent, actually measured charge is considered.

Disadvantages:

Knowledge of the errors Δq_i strongly depends on gas mixture, gas gain, amplification structure, electronics etc.

Requires extensive studies to determine the dependencies.

Does not include errors due to pad response (systematics of the CoG method).

Apply a mean error by parameterising the residual distributions as

$$\sigma = \sqrt{\sigma_0^2 + d^2 \cdot z}$$

where d is the diffusion dependent contribution of the resolution. This is the way the resolution usually is parameterised in dependency on the drift distance.

Advantages:

Includes also systematic errors due to pad response

Parameters easy to determine

Disadvantages:

Only mean error per hit is used, charge and shape of charge distribution are not considered

A posteriori application of the error. First the track has to be fitted to calculate the residuals, only then the errors can be determined. (Two runs: First track fit with errors set to 1, second track fit with errors. Or errors determined externally by another run or MC simulation)

I am not sure which version I should implement, probably we want both to compare. Or make a more complicated parametrisation of the residual distribution in dependence on the charge, width of the hit etc.

On the technical implementation of the calculation (class design etc.) I will make another thread because I also have some questions on that.

Cheers

Martin

P.S. Does anyone know how to get formulas into the forum? Not even inserting individual images works since one can only upload one file per message. I attached one image with the two definitions. Not even forcing a line break is possible. Sorry

File Attachments

1) [formulas.png](#), downloaded 547 times

Subject: Re: Hit errors in MarlinTPC

Posted by [tonyj](#) on Fri, 04 Jun 2010 17:49:45 GMT

[View Forum Message](#) <> [Reply to Message](#)

You can embed images anywhere you like in a posting using the [img] tag:

Google has a service which can generate equations from tex on the fly, so combining the two you can put equations anywhere you like:

or

You can even use the same technique to embed charts:

For more information see here:

http://code.google.com/apis/chart/docs/chart_playground.html

and here:

<http://code.google.com/apis/chart/docs/gallery/formulas.html>

Subject: Re: Hit errors in MarlinTPC

Posted by [killenberg](#) on Fri, 04 Jun 2010 19:14:37 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Tony,

thank you very much! This is extremely useful and one can also use this technique in other forums.

Cheers

Martin

Subject: LCTPC Conditions Database
Posted by [ralf](#) on Wed, 11 Aug 2010 14:04:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

Just to inform everyone:

After a bit of delay, the conditions database for MarlinTPC is partly online: so far a database for developers is online on the server. Using it you can test code needing conditions data or store conditions data used in your test cases.

This database also serves as kind of a test case for the server setup in general, so please make use of it :)

The Large Prototype database will go online a bit later.

If you want access or want a separate database for your own small prototype on the central server, please contact me.

Cheers, Ralf.

Subject: Recent Changes in MarlinTPC trunk
Posted by [ralf](#) on Wed, 11 Aug 2010 15:51:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello everyone,

following the latest changes of the MarlinTPC trunk, I have found some problems and some suggestions/discussion topics.

In the latest trunk version (r2207), there seems to be an error in the tools/Field/FindCGA.cmake file. The path to the CGAPack library seems to look in the Geant4 lib directory. This kept this version from compiling at my place. After I changed the line

```
PATHS $ENV{G4LIB}/$ENV{G4SYSTEM}/
```

to

```
PATHS ${Mokka_HOME}/lib/$ENV{G4SYSTEM}/
```

it works for me.

On my computer, using latest trunk (r2207), a simple chain using the ConditionsProcessor, the TrackerRawDataToDataConverterProcessor and the PedestalSubtractorProcessor does always stop with:

```
*** glibc detected *** Marlin: double free or corruption  
and then some more messages.
```

This worked in the last trunk revision before (r2198). Haven't yet figured out why this happens.

In the PedestalSubtractor, I think the renaming of the override parameter from "MaximumADCValue" to "MaximumADCValueOverride" is a good idea. Maybe we should make this a convention to append to optional parameters like this the word "Override"?

Regarding the use of the ConditionsMap for e.g. the Pedestals instead of the PedestalHandler: To my surprise, this works even faster than the use of the two nested maps that were used before in the handler (although speed was one of the ideas behind that approach). And I like the idea with the default key_type types.

Nevertheless, I would still vote to use the PedestalHandler (or other "Handlers", which should probably all be renamed into sth. with ChangeListener, since they are derived from the IConditionsChangeListener); probably using the ConditionsMap approach inside. The reason for this being that you can then handle in a consistent way missing conditions data for single channel in an event or even missing conditions for a whole time span.

For example: in the Pedestal"Handler", if a pedestal for a channel is missing, you get the value 0 for the value and -1 for the width. So your processor goes on processing without doing basically anything to the data. With the map solution, an exception will be thrown.

Of course, we would have to discuss what behavior is wanted in such a case (only a warning or really stopping the processing). This would of course also depend on the condition in question.

Also, using the latest versions of LCCD/CondDBMySQL, there is finally a solution how to treat times when a certain condition(s) is missing in the database. As far as I understood it, the proposed solution is to use a class derived from the ChangeListener and implement the handling of this case there.

With these changes, you can implement if to use in such a case a default value, the last valid conditions collection or stop processing. Also checks for which time span a condition is allowed to be missing (and probably the last valid one would be used) should be possible. From experience of other groups using real data, the case, that a condition missing for a short period of time, will happen.

Related to this and referring to this older thread:

I would still vote to use only one approach throughout MarlinTPC to access conditions data. In my opinion, using the ConditionsProcessor would be the cleanest way.

Using two ways makes making errors easier and finding errors harder. Also, the ConditionsProcessor does in principle the same thing as is done in the backup solution.

So, after going on for probably too long: tell me what you think :)

Cheers, Ralf.

Subject: Re: Recent Changes in MarlinTPC trunk
Posted by [ralf](#) on Wed, 11 Aug 2010 16:06:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

Another question:

Shouldn't we get rid of this "#ifdef USE_LCCD" tests?

IMHO MarlinTPC depends on a conditions system and there is only LCCD available if we stick to the ILC software.

Cheers, Ralf.

Subject: Re: Recent Changes in MarlinTPC trunk
Posted by [jabernathy](#) on Wed, 11 Aug 2010 19:16:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Ralf,

ralf wrote on Wed, 11 August 2010 08:51

In the latest trunk version (r2207), there seems to be an error in the tools/Field/FindCGA.cmake file. The path to the CGAPack library seems to look in the Geant4 lib directory. This kept this version from compiling at my place. After I changed the line

```
PATHS $ENV{G4LIB}/$ENV{G4SYSTEM}/
```

to

```
PATHS ${Mokka_HOME}/lib/$ENV{G4SYSTEM}/
```

it works for me.

In the GNUMakefile.common the target directory for the CGAPack library is set as

```
LIBPACKNAME := libCGAPack.so  
TARGETDIR := $(G4WORKDIR)/lib/$(G4SYSTEM)  
LIBPACK := $(TARGETDIR)/$(LIBPACKNAME)
```

The problem occurs because G4WORKDIR is set differently when Mokka is installed by the ilctool scripts. In the new version of the file written by Jan both directories are used.

ralf wrote on Wed, 11 August 2010 08:51 On my computer, using latest trunk (r2207), a simple chain using the ConditionsProcessor, the TrackerRawDataToDataConverterProcessor and the PedestalSubtractorProcessor does always stop with:

```
*** glibc detected *** Marlin: double free or corruption  
and then some more messages.
```

This worked in the last trunk revision before (r2198). Haven't yet figured out why this happens.

I don't have the same error using those processors. Can you send me your files so I can test it locally?

ralf wrote on Wed, 11 August 2010 08:51 In the PedestalSubtractor, I think the renaming of the override parameter from "MaximumADCValue" to "MaximumADCValueOverride" is a good idea. Maybe we should make this a convention to append to optional parameters like this the word "Override"?

In the latest version of the trunk it is named correctly.

ralf wrote on Wed, 11 August 2010 08:51
Nevertheless, I would still vote to use the PedestalHandler (or other "Handlers", which should probably all be renamed into sth. with ChangeListener, since they are derived from the IConditionsChangeListener); probably using the ConditionsMap approach inside.

Regardless of the solution used I suggest that it is handled globally.

One disadvantage of creating ConditionMaps / ConditionHandlers is that they are created locally: ie one per processor. This leaves the possibility of two processors using different default values.

I'll respond to your other suggestion in the older thread to revive it :)

- Jason

Subject: Re: Recent Changes in MarlinTPC trunk
Posted by [killenberg](#) on Thu, 12 Aug 2010 08:54:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Ralf,

yes, I think we should remove the "#ifdef USE_LCCD" tests. The same hold for "#ifdef USE_GEAR" and "#ifdef USE_AIDA", which I think are also still in in some places.

MarlinTPC cannot run or compile without these packages, they are defined as dependencies in the CMakeLists.txt.

Cheers

Martin

Subject: Re: Recent Changes in MarlinTPC trunk
Posted by [ralf](#) on Thu, 12 Aug 2010 17:22:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi

thanks for the fast replies :)

jabernathy wrote on Wed, 11 August 2010 12:16
ralf wrote on Wed, 11 August 2010 08:51 On my computer, using latest trunk (r2207), a simple chain using the ConditionsProcessor, the TrackerRawDataToDataConverterProcessor and the PedestalSubtractorProcessor does always stop with:
*** glibc detected *** Marlin: double free or corruption
and then some more messages.
This worked in the last trunk revision before (r2198). Haven't yet figured out why this happens.

I don't have the same error using those processors. Can you send me your files so I can test it locally?

I figured out that it happens after all the processing of the events is done (took some hours to run it in DEBUG mode).
Do you have access to AFS? Tomorrow, I could place the input files somewhere there.

jabernathy wrote on Wed, 11 August 2010 12:16
ralf wrote on Wed, 11 August 2010 08:51 In the PedestalSubtractor, I think the renaming of the override parameter from "MaximumADCValue" to "MaximumADCValueOverride" is a good idea. Maybe we should make this a convention to append to optional parameters like this the word "Override"?

In the latest version of the trunk it is named correctly.

I know. I just wanted to point out, that I liked your idea and that one might do it like this in the other processors :)

CU, Ralf.

Subject: Re: Recent Changes in MarlinTPC trunk
Posted by [jabernathy](#) on Thu, 12 Aug 2010 19:34:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

ralf wrote on Thu, 12 August 2010 10:22

I figured out that it happens after all the processing of the events is done (took some hours to run it in DEBUG mode).

Do you have access to AFS? Tomorrow, I could place the input files somewhere there.

I have access to AFS but I don't think I have a DESY login.

Another thing that might work is running it inside a debugger (if you haven't already).

```
> gdb Marlin
> set arg steering_file.xml
> run
> *** glibc detected *** Marlin: double free or corruption
> backtrace
```

ralf wrote on Thu, 12 August 2010 10:22

I know. I just wanted to point out, that I liked your idea and that one might do it like this in the other processors :)

Thanks, it is always wise to keep `variable_names_as_verbose_as_possible`. :)

Subject: Re: Recent Changes in MarlinTPC trunk
Posted by [ralf](#) on Fri, 13 Aug 2010 17:24:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

I have put the files of the now failing example here:
/afs/desy.de/group/flc/pool/rdiener/public

There is:

TPCRawData.slcio : file with 100 events (7 modules with 5000 channels each)
condDB_pedestal_HEAD_12623040000000000000.slcio : contains the pedestals as simple file
pedestal_substraction_trunk.xml : MarlinTPC steering file; the MaximumADCValueOverride parameter is set wrong (should be 255), but this shouldn't be a problem. I have now set that it processes 3 events, since it doesn't matter for the error how many events it processes.
generateRawData.cpp : a small program with which I created the raw data - not programmed too clean (more copy&pasted from other programs I had). The pedestals are calculated from the RawData using the PedestalCalculator.

I hope there are no problems with access rights. They should be set in the right way now.

I tried the debugger, but at least for me it didn't provide much useful information.
I hope it isn't some embarrassing, simple error I overlooked :)

Subject: Re: Recent Changes in MarlinTPC trunk
Posted by [sailer](#) on Mon, 16 Aug 2010 08:09:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

ralf wrote on Wed, 11 August 2010 08:51
On my computer, using latest trunk (r2207), a simple chain using the ConditionsProcessor, the TrackerRawDataToDataConverterProcessor and the PedestalSubtractorProcessor does always stop with:
*** glibc detected *** Marlin: double free or corruption
and then some more messages.
This worked in the last trunk revision before (r2198). Haven't yet figured out why this happens.

Hi,
this looks very similar to some problem I also encountered, it had to do with the fact that some libraries were loaded twice, due to the fact that one they showed up in the wrong place of the MARLIN_DLL. If one of the libraries loaded first are linked to one of the libraries loaded later they are "unloaded" twice at the end, which is not possible and this error shows up.

If that is the reason, the error should also show up if just running

Marlin -c steering.xml

Cheers,
AndrÃ©

Subject: Re: Recent Changes in MarlinTPC trunk
Posted by [ralf](#) on Tue, 17 Aug 2010 16:12:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

sorry for the late reply, but I wasn't at work yesterday.

Thanks for the tip AndrÃ©, although I have to admit I don't fully understand your comment. Actually, it also happens when I only start Marlin (without any steering file or other command line options), which I tried after the Marlin -c test.

When comparing the two MarlinTPC libraries, the difference was mainly the inclusion of the CGAPack lib. After I compiled the newer version without Mokka, the error was gone. I used Geant4 9.3 with Mokka 07-04 from the latest ILCSoft release (v01-09).

CU, Ralf.

Subject: Re: Recent Changes in MarlinTPC trunk
Posted by [sailer](#) on Tue, 17 Aug 2010 16:36:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,
I'll try to explain more clearly. :)

I encountered this problem, when my MARLIN_DLL looked something like this

```
MARLIN_DLL=libMarlinPandora.so:libPandoraPFANew.so
```

Now libMarlinPandora depends on PandoraPFA, so when MarlinPandora was loaded by Marlin, it also loads libPandoraPFANew, which is then loaded again by Marlin, because it is in the

MARLIN_DLL.

(PandoraPFANew is now no longer added to the MARLIN_DLL.)

At the end the library is tried to be "unloaded" twice or what ever happens at the end of a program, which causes this "double free".

The CGAPack lib contains a lot of libraries, maybe one of them was loaded twice, like for me with the PandoraPFANew?

Cheers,
Andre

Subject: Re: Recent Changes in MarlinTPC trunk
Posted by [ralf](#) on Tue, 17 Aug 2010 16:50:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks for the detailed explanation :)

I will check this...

CU, Ralf.

Subject: Re: Recent Changes in MarlinTPC trunk
Posted by [ralf](#) on Wed, 18 Aug 2010 12:23:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

Just to stay up-to-date:

I looked at the libraries but couldn't spot anything obvious. Jan said he would try to reproduce the problem. I hope he spots something :)

Subject: Re: Recent Changes in MarlinTPC trunk
Posted by [engels](#) on Thu, 19 Aug 2010 14:02:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Ralf,

i get 'permission denied' when trying to access:
/afs/desy.de/group/flc/pool/rdiener/public

Cheers,
Jan

Subject: Re: Recent Changes in MarlinTPC trunk
Posted by [engels](#) on Thu, 19 Aug 2010 15:54:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

ok, now it's working ;)

even running with the files in your public in an SL5 machine I could not reproduce the problem :(

anyway, here the output from ldd in my MarlinTPC library:

```
$ ldd $MARLIN_DLL
linux-gate.so.1 => (0xffffe000)
libMarlin.so.0.12 =>
/scratch/engels/nbuilds/2010-08-19/nb-release/Marlin/HEAD/lib/libMarlin.so.0.12 (0xf7ec4000)
liblcio.so.1.51 => /scratch/engels/nbuilds/2010-08-19/nb-release/lcio/HEAD/lib/liblcio.so.1.51
(0xf7df1000)
libsio.so.1.51 => /scratch/engels/nbuilds/2010-08-19/nb-release/lcio/HEAD/lib/libsio.so.1.51
(0xf7de0000)
libdcap.so => /scratch/engels/nbuilds/2010-08-19/nb-release/dcap/1.9.5-5/lib/libdcap.so
(0xf7dc5000)
libgearxml.so.0.14 =>
/scratch/engels/nbuilds/2010-08-19/nb-release/gear/HEAD/lib/libgearxml.so.0.14 (0xf7d84000)
libgear.so.0.14 => /scratch/engels/nbuilds/2010-08-19/nb-release/gear/HEAD/lib/libgear.so.0.14
(0xf7d2b000)
libMarlinTPCAnalysis.so.0.0 =>
/scratch/engels/nbuilds/2010-08-19/nb-release/MarlinTPC/HEAD/lib/libMarlinTPCAnalysis.so.0.0
(0xf7cba000)
libMarlinTPCValidation.so.0.0 =>
/scratch/engels/nbuilds/2010-08-19/nb-release/MarlinTPC/HEAD/lib/libMarlinTPCValidation.so.0.0
(0xf7c8f000)
libMarlinTPCDigitisation.so.0.0 =>
/scratch/engels/nbuilds/2010-08-19/nb-release/MarlinTPC/HEAD/lib/libMarlinTPCDigitisation.so.0.
0 (0xf7c26000)
libMarlinTPCCalibration.so.0.0 =>
```

/scratch/engels/nbuilds/2010-08-19/nb-release/MarlinTPC/HEAD/lib/libMarlinTPCCalibration.so.0.0 (0xf7c06000)
libMarlinTPCSimulation.so.0.0 =>
/scratch/engels/nbuilds/2010-08-19/nb-release/MarlinTPC/HEAD/lib/libMarlinTPCSimulation.so.0.0 (0xf7bd8000)
libMarlinTPCPhotoelectricSimulation.so.0.0 =>
/scratch/engels/nbuilds/2010-08-19/nb-release/MarlinTPC/HEAD/lib/libMarlinTPCPhotoelectricSimulation.so.0.0 (0xf7bcb000)
libMarlinTPCCloudSimulation.so.0.0 =>
/scratch/engels/nbuilds/2010-08-19/nb-release/MarlinTPC/HEAD/lib/libMarlinTPCCloudSimulation.so.0.0 (0xf7b91000)
libMarlinTPCReconstruction.so.0.0 =>
/scratch/engels/nbuilds/2010-08-19/nb-release/MarlinTPC/HEAD/lib/libMarlinTPCReconstruction.so.0.0 (0xf7a67000)
libMarlinTPCPhotoelectricReconstruction.so.0.0 =>
/scratch/engels/nbuilds/2010-08-19/nb-release/MarlinTPC/HEAD/lib/libMarlinTPCPhotoelectricReconstruction.so.0.0 (0xf7a29000)
libtpcconddata.so.0.0 =>
/scratch/engels/nbuilds/2010-08-19/nb-release/MarlinTPC/HEAD/lib/libtpcconddata.so.0.0 (0xf7a02000)
libMarlinTPC_tools_Position.so.0.0 =>
/scratch/engels/nbuilds/2010-08-19/nb-release/MarlinTPC/HEAD/lib/libMarlinTPC_tools_Position.so.0.0 (0xf79fa000)
libPadGeometryChecker.so.0.0 =>
/scratch/engels/nbuilds/2010-08-19/nb-release/MarlinTPC/HEAD/lib/libPadGeometryChecker.so.0.0 (0xf79f1000)
libHepRepOutput.so.0.0 =>
/scratch/engels/nbuilds/2010-08-19/nb-release/MarlinTPC/HEAD/lib/libHepRepOutput.so.0.0 (0xf79da000)
libMarlinTPCField.so.0.0 =>
/scratch/engels/nbuilds/2010-08-19/nb-release/MarlinTPC/HEAD/lib/libMarlinTPCField.so.0.0 (0xf79b7000)
libMarlinTPCToolProcessors.so.0.0 =>
/scratch/engels/nbuilds/2010-08-19/nb-release/MarlinTPC/HEAD/lib/libMarlinTPCToolProcessors.so.0.0 (0xf7985000)
libIntersectionCalculator.so.0.0 =>
/scratch/engels/nbuilds/2010-08-19/nb-release/MarlinTPC/HEAD/lib/libIntersectionCalculator.so.0.0 (0xf7981000)
libLCObjectCopier.so.0.0 =>
/scratch/engels/nbuilds/2010-08-19/nb-release/MarlinTPC/HEAD/lib/libLCObjectCopier.so.0.0 (0xf797d000)
libgsl.so.0 => /afs/desy.de/group/it/ilcsoft/gsl/1.8/lib/libgsl.so.0 (0xf781e000)

libgslcblas.so.0 => /afs/desy.de/group/it/ilcsoft/gsl/1.8/lib/libgslcblas.so.0 (0xf77f0000)
liblccd.so.1.0 => /scratch/engels/nbuilds/2010-08-19/nb-release/lccd/HEAD/lib/liblccd.so.1.0
(0xf77c9000)
libCore.so => /afs/desy.de/group/it/ilcsoft/root/5.26.00b/lib/libCore.so (0xf7115000)
libCint.so => /afs/desy.de/group/it/ilcsoft/root/5.26.00b/lib/libCint.so (0xf6b61000)
libRIO.so => /afs/desy.de/group/it/ilcsoft/root/5.26.00b/lib/libRIO.so (0xf69e5000)
libNet.so => /afs/desy.de/group/it/ilcsoft/root/5.26.00b/lib/libNet.so (0xf68e7000)
libHist.so => /afs/desy.de/group/it/ilcsoft/root/5.26.00b/lib/libHist.so (0xf6566000)
libGraf.so => /afs/desy.de/group/it/ilcsoft/root/5.26.00b/lib/libGraf.so (0xf63bc000)
libGraf3d.so => /afs/desy.de/group/it/ilcsoft/root/5.26.00b/lib/libGraf3d.so (0xf62d9000)
libGpad.so => /afs/desy.de/group/it/ilcsoft/root/5.26.00b/lib/libGpad.so (0xf621e000)
libTree.so => /afs/desy.de/group/it/ilcsoft/root/5.26.00b/lib/libTree.so (0xf6074000)
libRint.so => /afs/desy.de/group/it/ilcsoft/root/5.26.00b/lib/libRint.so (0xf604a000)
libPostscript.so => /afs/desy.de/group/it/ilcsoft/root/5.26.00b/lib/libPostscript.so (0xf600a000)
libMatrix.so => /afs/desy.de/group/it/ilcsoft/root/5.26.00b/lib/libMatrix.so (0xf5e04000)
libPhysics.so => /afs/desy.de/group/it/ilcsoft/root/5.26.00b/lib/libPhysics.so (0xf5d7f000)
libMathCore.so => /afs/desy.de/group/it/ilcsoft/root/5.26.00b/lib/libMathCore.so (0xf5bd8000)
libThread.so => /afs/desy.de/group/it/ilcsoft/root/5.26.00b/lib/libThread.so (0xf5b95000)
libMinuit2.so => /afs/desy.de/group/it/ilcsoft/root/5.26.00b/lib/libMinuit2.so (0xf5a77000)
libCLHEP-2.0.4.2.so => /afs/desy.de/group/it/ilcsoft/CLHEP/2.0.4.2/lib/libCLHEP-2.0.4.2.so
(0xf5925000)
libRAIDA.so.1.4 =>
/scratch/engels/nbuilds/2010-08-19/nb-release/RAIDA/HEAD/lib/libRAIDA.so.1.4 (0xf587c000)
libstdc++.so.6 => /usr/lib/libstdc++.so.6 (0xf5767000)
libm.so.6 => /lib/libm.so.6 (0xf5740000)
libgcc_s.so.1 => /lib/libgcc_s.so.1 (0xf5734000)
libc.so.6 => /lib/libc.so.6 (0xf55f0000)
libstreamlog.so.0.1 =>
/scratch/engels/nbuilds/2010-08-19/nb-release/Marlin/HEAD/lib/libstreamlog.so.0.1 (0xf55eb000)
libz.so.1 => /usr/lib/libz.so.1 (0xf55d7000)
libdl.so.2 => /lib/libdl.so.2 (0xf55d3000)
libpthread.so.0 => /lib/libpthread.so.0 (0xf55bc000)
libconddb.so.0 =>
/scratch/engels/nbuilds/2010-08-19/nb-release/CondDBMySQL/HEAD/lib/libconddb.so.0
(0xf553a000)
libncurses.so.5 => /usr/lib/libncurses.so.5 (0xf54f1000)
libcrypt.so.1 => /lib/libcrypt.so.1 (0xf54be000)
/lib/ld-linux.so.2 (0x00b77000)
libmysqlclient.so.15 => /afs/desy.de/group/it/ilcsoft/mysql/5.0.45/lib/mysql/libmysqlclient.so.15
(0xf5460000)
libnsl.so.1 => /lib/libnsl.so.1 (0xf5448000)

Cheers,
Jan

Subject: Problem on new trunk version r2253
Posted by [katoy](#) on Wed, 08 Sep 2010 16:28:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,
Martin updated the MarlinTPC trunk, r2253.
I got and build them, but the building error has been
happened. The error message are in below,

```
/afs/desy.de/user/k/katoy/MarlinTPC/r2253/reconstruction/src /TrackFitterKalman.cc:128: error:  
prototype for `std::string marlintpc::TrackFitterKalman::getRevision()' does not match any in class  
'marlintpc::TrackFitterKalman'  
/afs/desy.de/user/k/katoy/MarlinTPC/r2253/reconstruction/inc lude/TrackFitterKalman.h:58: error:  
candidate is: virtual std::string marlintpc::TrackFitterKalman::getRevision() const  
/afs/desy.de/user/k/katoy/MarlinTPC/r2253/reconstruction/src /TrackFitterKalman.cc:138: error:  
prototype for `IMPL::TrackImpl* marlintpc::TrackFitterKalman::fitTrack(const EVENT::Track*)'  
does not match any in class `marlintpc::TrackFitterKalman'  
/afs/desy.de/user/k/katoy/MarlinTPC/r2253/reconstruction/inc lude/TrackFitterKalman.h:94: error:  
candidate is: virtual IMPL::TrackImpl* marlintpc::TrackFitterKalman::fitTrack(const  
EVENT::Track*) const  
make[2]: *** [reconstruction/CMakeFiles/lib_MarlinTPCReconstruction.dir/s  
rc/TrackFitterKalman.o] Error 1  
make[1]: *** [reconstruction/CMakeFiles/lib_MarlinTPCReconstruction.dir/a ll] Error 2  
make: *** [all] Error 2
```

I think Martin checked r2253 without Kalman filter. Because
TrackFitterKalman requires to have KalTest and KalDet.
I need help to solve the above problem. Could someone help me?

Yukihiro Kato

Subject: Re: Problem on new trunk version r2253
Posted by [killenberg](#) on Wed, 08 Sep 2010 16:34:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

Dear Kato-san,

yes, I did not check with Kaltest and Kaldet. I have it on my disk but have to recompile with the latest ILCSoft.

I will provide a fix as soon as possible.

Cheers

Martin

Subject: Re: Problem on new trunk version r2253
Posted by [kato](#) on Wed, 08 Sep 2010 16:42:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi, Martin

I changed the two lines in TrackFitterKalman.cc and confirmed it to be built. The lines I changed are,

```
std::string TrackFitterKalman::getRevision() const  
IMPL::TrackImpl * TrackFitterKalman::fitTrack(EVENT::Track const * seedTrack) const
```

Could you update these modifications to trunk?

Yukihiro kato
Kinki Univ.

Subject: Re: Problem on new trunk version r2253
Posted by [killenberg](#) on Wed, 08 Sep 2010 21:38:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

The trunk is now up to date and compiles with KalTest/KalDet.
The changes in TrackFitterKalman.cc had somehow slipped the merge into the trunk. I had tested with my branch where it was already correct.

I also turned on the automatic revision logging for this file. Now the revision in the code is automatically updated when a new version is committed.

Cheers

Martin

Subject: Changes in PulseFinder

Posted by [rosemann](#) on Wed, 22 Sep 2010 16:00:20 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi,

after only a couple of weeks I finally get to spend some time on updating/implementing some stuff into MarlinTPC again. So I pick up on the last meeting and Ralfs initial posting.

For now I concentrate on bringing the PulseFinderProcessor to version 1.0.

ralf wrote on Wed, 11 August 2010 08:51[...]

In the PedestalSubtractor, I think the renaming of the override parameter from "MaximumADCValue" to "MaximumADCValueOverride" is a good idea. Maybe we should make this a convention to append to optional parameters like this the word "Override"?[...]

Yes, as discussed in the last meeting. I will try to prepare the PulseFinderProcessor as the first prototype with this scheme.

ralf wrote on Wed, 11 August 2010 08:51[...]
Regarding the use of the ConditionsMap for e.g. the Pedestals instead of the PedestalHandler:

To my surprise, this works even faster than the use of the two nested maps that were used before in the handler (although speed was one of the ideas behind that approach). And I like the idea with the default key_type types.

Nevertheless, I would still vote to use the PedestalHandler (or other "Handlers", which should probably all be renamed into sth. with ChangeListener, since they are derived from the IConditionsChangeListener); probably using the ConditionsMap approach inside.

The reason for this being that you can then handle in a consistent way missing conditions data for single channel in an event or even missing conditions for a whole time span.

I would separate the two aspects of design and implementation. The implementation (for speed, simplicity, ...) is the thing that needs to follow the design.

As also proposed in the meeting, and I hope I get it right this time:

1) The default way of providing "conditions data" to the event stream is via the Conditions

Processor. How this special processor accesses the data is of no direct concern to us (e.g. database, some file).

2) A processor (downstream in the chain) accesses the conditions data by a "Listener" (or "Handler"), the naming is a bit unfortunate. Maybe we can agree on a single naming scheme here. The processor in question then always has a listener/handler object that can access (with all the overhead) the relevant information.

3) For overriding this access method, one specifies the "OverrideValues". If these are present, they will precede over the values from the Handler/Listener. Best practice might be that the Handler/Listener isn't even instantiated in the processor; I'll see if I can come up with a halfway generic solution for the PulseFinderProcessor.

The second part is then the implementation. From what I have seen in the current (revision 2256) version of the PulseFinderProcessor, there is a mixture of two access methods to the conditions data, namely the Pedestals. One is the "PedestalHandler", the other one uses the ConditionsMap.

Since the Handler adds some functionality we might (or already do) want to have, I'm in favour of using this. Especially when the new and long debated features of LCCD/CondDBMySQL can be incorporated.

Either way, I am for a uniform way of doing this access.

Regards,
Christoph

Subject: building LCFIVertex with AIDAJNI
Posted by [voutsi](#) on Wed, 29 Sep 2010 21:15:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

I want to build LCFIVertex package with AIDAJNI. I have ilcsoft version 01-08-01 so trying to install AIDAJNI 3.2.3 there using ilcsoft-install. It says that build is successful

BUILD SUCCESSFUL

Total time: 7 seconds

sh: gmake: command not found

sh: gmake: command not found

Traceback (most recent call last):

File "/var/autofs/nfs/rawcmos11/voutsi/v01-08-01/ilcsoft-install", line 72, in ?
ilcsoft.makeinstall()

```
File "/var/autofs/nfs/rawcmos11/voutsi/v01-08-01/ilcsoft/ilcsoft.p y ", line 423, in makeinstall
  mod.install([])
File "/var/autofs/nfs/rawcmos11/voutsi/v01-08-01/ilcsoft/baseilc.p y ", line 758, in install
  self.compile()
File "/var/autofs/nfs/rawcmos11/voutsi/v01-08-01/ilcsoft/aidajni.p y ", line 71, in compile
  os.system( 'tar -xzf %s-%s-'+self.os_ver.type+'-g+++.tar.gz' % (self.alias, self.version) )
TypeError: not all arguments converted during string formatting
```

So I don't really understand if is successful or not. Then trying to build LCFIVertex with AIDAJNI and

```
-- Check for AIDAJNI: /rawcmos11/voutsi/ilcsoft/AIDAJNI/3.2.3
-- Java version 1.6.0 configured successfully!
-- Check for JAIDA: /rawcmos11/voutsi/ilcsoft/JAIDA/3.2.3 -- works
-- Check for AIDAJNI: /rawcmos11/voutsi/ilcsoft/AIDAJNI/3.2.3 -- failed to find AIDAJNI AIDAJNI
library!!
-- Check for AIDAJNI: /rawcmos11/voutsi/ilcsoft/AIDAJNI/3.2.3 -- failed to find AIDAJNI FHJNI
library!!
CMake Error at /rawcmos11/voutsi/ilcsoft/CMakeModules/v01-08-01/FindAIDAJNI .cmake:234
(MESSAGE):
  Check for AIDAJNI: /rawcmos11/voutsi/ilcsoft/AIDAJNI/3.2.3 -- failed!!
Call Stack (most recent call first):
  /rawcmos11/voutsi/ilcsoft/CMakeModules/v01-08-01/MacroLoadPa ckage.cmake:103
(FIND_PACKAGE)
  /rawcmos11/voutsi/ilcsoft/CMakeModules/v01-08-01/MacroCheckD eps.cmake:36
(LOAD_PACKAGE)
  CMakeLists.txt:267 (CHECK_DEPS)
```

So seems like something went wrong at AIDAJNI install.. Have already install java and JAIDA. The path to the AIDAJNI library is also defined correctly.

Any comment is very welcomed,

Thanks a lot

Yorgos

Subject: Measurement of Impact parameter
Posted by [Ryan_Page](#) on Tue, 05 Oct 2010 11:51:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

I have been doing some simple simulation and track reconstruction using muons generated by the particle gun to test a vertex geometry that I have built in Mokka. To get a benchmark I used the vxd05 and the vxd03 vertex detectors in the ILD00 geometry. When I plotted the sigma of a fit to the impact parameter residuals I noticed that the D0 impact parameter resolution was worse than the Z0 as a function of momentum (1-10GeV). I then looked at the impact parameter resolution as a function of phi and noticed that at two point in phi I was getting a gaussian like rise in the resolution, which was confirmed when I checked the covariance matrix of D0; this effect is not seen in Z0. The effect occurs in both the vxd05 and vxd03 detectors, but oddly not in my design, although I still see a slightly better Z0 resolution.

The gun direction was set to 89 degrees and was set to step in phi by 1 degree each event and I ran for 10,000 events. I then did another test where I aimed at one of the points in phi where I was seeing this effect and smeared uniformly over it. For the reconstruction I used a marlin steer file based on the LOI steer file.

I am checking the other covariance matrices at the moment and checking the pulls again, both mean and sigma, but if anyone else has any ideas why I am seeing this effect please let me know. I have attached a couple of plots to illustrate. Also I have checked the other track parameters both LDC and Si and I dont see anything odd as a function of phi.

If you need more information then please let me know.

Thanks

Ryan

File Attachments

1) [Plots.pdf](#), downloaded 440 times

Subject: Re: Measurement of Impact parameter
Posted by [harderk](#) on Tue, 05 Oct 2010 12:03:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Ryan,

Just an almost blind guess: could this be coming from the beam spot size? This would affect d0, but not z0. Linear collider beams are very flat and wide, as far as I remember.

(I have no idea why you wouldn't see this effect in your own vertex detector design, though -

unless you somehow use different beam parameters when simulating it.)

Cheers,
Kristian

Subject: Re: Measurement of Impact parameter
Posted by [Ryan_Page](#) on Tue, 05 Oct 2010 12:15:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

I am only simulating single muons with the particle gun not actual collisions yet.

Cheers

Ryan

Subject: Re: Measurement of Impact parameter
Posted by [harderk](#) on Tue, 05 Oct 2010 12:22:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

Sure, but the d0 measurement must be with respect to *something*, and if you are using some standard track reconstruction code it might take a beamspot uncertainty into account even if what you simulate does not strictly speaking have a beamspot at all.
But as I said, it was just a guess anyway.

Subject: Re: Measurement of Impact parameter
Posted by [Ryan_Page](#) on Tue, 05 Oct 2010 13:12:41 GMT
[View Forum Message](#) <> [Reply to Message](#)

I see what you mean I will take a look at the documentation again to see if there is something like that in there.

Thanks

Subject: Re: Measurement of Impact parameter
Posted by [aplin](#) on Tue, 05 Oct 2010 17:03:44 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Ryan

we very recently found a bug in the vxd digitiser, which means that the smearing is not being done correctly along the ladders. I want to tidy up the digitiser and submit it to svn in the next couple of days.

The problem in d0 vs phi you observe is identical to what I have also seen.

By the way does your design have ladders? Or is it cylinders?

Cheers,

Steve.

Subject: Re: Measurement of Impact parameter
Posted by [Ryan_Page](#) on Tue, 05 Oct 2010 17:13:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

Mine is ladders but the way they are positioned is different, whether this would effect the outcome i dont know, you would not have thought so. But it still maybe effecting the performance of mine as I see better Z0 resolution than D0. I will check when the next release comes out.

Thanks for the help,

Ryan

Subject: Re: Recent Changes in MarlinTPC trunk
Posted by [ralf](#) on Fri, 15 Oct 2010 15:41:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

Late solution, but I thought, I anyway keep this thread up-to-date:

The

*** glibc detected *** Marlin: double free or corruption

problem is now solved. I used the latest release of ILCSoft v01-09-03 and since then it has vanished (it was still present with v01-09-02). Could be that the cause is removed since this is the first time SL5/32bit is especially considered in the ilcinstaller.

Subject: MarlinTPC flagwords

Posted by [rosemann](#) on Thu, 21 Oct 2010 12:44:52 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi everyone,

I implemented a header file, which currently resides in my branch under reconstruction/include/FlagwordDefinitions.h

Please comment on the layout and the completeness of the implemented flagwords.

Please note that I slightly changed Martins proposal and extended the basic static int definitions with functions that are global to that namespace; hopefully this will make it easier to use/read.

Here's the code:

```
#ifndef FLAGWORDDEFINITIONS_H
#define FLAGWORDDEFINITIONS_H
```

```
/** a collection of the quality flagwords that are stored in the respective LCIO classes:
    these are collected into three different namespaces (for pulses, hits and tracks),
    and equipped with "global" functions (within this namespace) to ease the usage
```

```
the flags were discussed in the forum
```

```
example usage:
```

```
EVENT::TrackerPulse* aPulse;
if(isUnderflow(aPulse->getQuality())
{...}
```

```
int quality = 0;
quality = setMultiplePulseCandidate(quality);
quality = setOverflow
**/
```

```
namespace marlintpc
{
    namespace pulseflag
```

```

{
static int const MULTIPLE_PULSE_CANDIDATE = 1;
static int const ANOMALOUS_SHAPE      = 1 << 1;
static int const OVERFLOW_BIN         = 1 << 2;
static int const UNDERFLOW_BIN        = 1 << 3;
static int const WAS_SPLIT             = 1 << 4;

int setMultiplePulseCandidate(int i)
{
return i |= MULTIPLE_PULSE_CANDIDATE;
}

int setAnomalousShape(int j)
{
return j |= ANOMALOUS_SHAPE;
}

int setOverflow(int k)
{
return k |= OVERFLOW_BIN;
}

int setUnderflow(int m)
{
return m |= UNDERFLOW_BIN;
}

int setSplit(int n)
{
return n |= WAS_SPLIT;
}

bool isMultiplePulseCandidate(int i)
{
return i & MULTIPLE_PULSE_CANDIDATE;
}

bool isAnomalousShape(int j)
{
return j & ANOMALOUS_SHAPE;
}

```



```

bool isOverflow(int k)
{
    return k & OVERFLOW_BIN;
}

bool isUnderflow(int m)
{
    return m & UNDERFLOW_BIN;
}

bool isSplit(int n)
{
    return n & WAS_SPLIT;
}

}

namespace hitflag
{
    static int const CONTAINS_BROKEN_CHANNEL = 1;
    static int const CONTAINS_NOISY_CHANNEL = 1 << 1;
    static int const HIT_AT_BORDER = 1 << 2;
    static int const CONTAINS_OVERFLOW_PULSE = 1 << 3;
    static int const CONTAINS_SPLIT_PULSE = 1 << 4;
    static int const CONTAINS_ANOMALOUS_PULSE = 1 << 5;
    static int const CONTAINS_EMPTY_CHANNEL = 1 << 6;
    static int const MULTIPLE_HIT_CANDIDATE = 1 << 7;

    int setBrokenChannel(int i)
    {
        return i |= CONTAINS_BROKEN_CHANNEL;
    }

    int setNoisyChannel(int j)
    {
        return j |= CONTAINS_NOISY_CHANNEL;
    }

    int setAtBorder(int k)
    {
        return k |= HIT_AT_BORDER;
    }
}

```

```
int setOverflowPulse(int m)
{
    return m |= CONTAINS_OVERFLOW_PULSE;
}

int setSplitPulse(int n)
{
    return n |= CONTAINS_SPLIT_PULSE;
}

int setAnomalousPulseShape(int p)
{
    return p |= CONTAINS_ANOMALOUS_PULSE;
}

int setEmptyChannel(int q)
{
    return q |= CONTAINS_EMPTY_CHANNEL;
}

int setMultipleHitCandidate(int r)
{
    return r |= MULTIPLE_HIT_CANDIDATE;
}

bool isBrokenChannel(int i)
{
    return i & CONTAINS_BROKEN_CHANNEL;
}

bool isNoisyChannel(int j)
{
    return j & CONTAINS_NOISY_CHANNEL;
}

bool isAtBorder(int k)
{
    return k & HIT_AT_BORDER;
}

bool isOverflowPulse(int m)
```

```

{
    return m & CONTAINS_OVERFLOW_PULSE;
}

bool isSplitPulse(int n)
{
    return n & CONTAINS_SPLIT_PULSE;
}

bool isAnomalousPulseShape(int p)
{
    return p & CONTAINS_ANOMALOUS_PULSE;
}

bool isEmptyChannel(int q)
{
    return q & CONTAINS_EMPTY_CHANNEL;
}

bool isMultipleHitCandidate(int r)
{
    return r & MULTIPLE_HIT_CANDIDATE;
}
}

namespace trackflag
{
    static int const HELIX_TRACKMODEL      = 1;
    static int const STRAIGHT_LINE_TRACKMODEL = 1 << 1;
    static int const MULTIPLE_TRACK_CANDIDATE = 1 << 2;
    static int const CHISQUARE_FITTER      = 1 << 8;
    static int const KALMAN_FITTER        = 1 << 9;
    static int const LIKELIHOOD_FITTER    = 1 << 10;

    int setHelixTrackModel(int i)
    {
        return i |= HELIX_TRACKMODEL;
    }

    int setStraightLineTrackModel(int j)
    {
        return j |= STRAIGHT_LINE_TRACKMODEL;
    }
}

```

```
}

int setMultipleTrackCandidate(int j)
{
    return j |= MULTIPLE_TRACK_CANDIDATE;
}

int setChiSquareFitter(int j)
{
    return j |= CHISQUARE_FITTER;
}

int setKalmanFitter(int j)
{
    return j |= KALMAN_FITTER;
}

int setLikelihoodFitter(int j)
{
    return j |= LIKELIHOOD_FITTER;
}
}
}
#endif //FLAGWORDDEFINITIONS_H
```

Sorry that the post is so long.

Cheers,
Christoph

Subject: Inconsistencies with "Transience Flag"
Posted by [rosemann](#) on Fri, 19 Nov 2010 10:18:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi everyone,

I noticed some slight confusion and unfortunately also plain errors/bugs concerning the usually common processor parameter of whether the output collection is to be stored (e.g. in a file) or not.

The currently common name is "SetOutputTransient". Due to historic reasons these are sometimes still integers. Even worse, sometimes the logic is turned around, so only when you set

this parameter to true/non-zero, the collection will be saved.

So I will change this in every processor that I can find to the right logic. But besides this, it might be easier to understand -- in reading and writing -- to call this parameter "SetOutputPersistent" or even something like "CommitOutputToStorage".

This will of course lead to a change in the "interface", or rather a change in all the existing steering files.

Opinions? Objections?

Cheers,
Christoph

Subject: Re: Inconsistencies with "Transience Flag"
Posted by [killenberg](#) on Fri, 19 Nov 2010 10:44:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

In principle we could skip this flag at all. The LCIOOutputProcessor is able to drop or preserve collections, either by type or by name.

But I found it convenient to have this flag with the processor. I agree that the name is not very easy to understand, but I think the logic is good: Normally a collection is written, unless specified otherwise.

The parameters you propose have the inverse logic: The parameter has to be set for the collection to be written, and is true by default. Would this not cause more confusion, now that some users are already used to the current scheme?

How about changing the comment lines so people are aware what transient means: A collection is NOT written to disk if the transient flag is true (default: false. The collection is written to disk).

In terms of backward compatibility: A new parameter does not necessarily have to be in the steering file. We just have to make sure it has a reasonable default value. I don't know if a parameter (the old one) is in the steering file but was removed from the processor. We cannot leave both (for backward compatibility) because their use is mutually exclusive. And we cannot give precedence to one. This would lead to strange bugs if both or the 'wrong' one is used/modified in the steering file.

So we could change the parameter without causing too much trouble. But I vote for clarifying the

usage of SetOutputTransient and making it consistent.

Cheers

Martin

Subject: Re: Inconsistencies with "Transience Flag"
Posted by [rosemann](#) on Fri, 19 Nov 2010 11:11:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Martin,

I'm all in favour for simplicity. A comment needs to be read and understood, which is a complication. If the parameter is called "WriteOutputToStorage", then there is no misunderstanding about software concepts, and no double-negation ("i don't want to not store the output") is needed to achieve the same effect.

I've collected five votes so far, (hopefully without biasing the voters) and all are in favour of the inverted logic w.r.t. what we have now.

Cheers,
Christoph

If we are voting anyhow, the here's the preliminary count:
six votes for "WriteOutputToStorage" (default true)
one vote for "SetOutputTransient" (default false)

Subject: Re: Inconsistencies with "Transience Flag"
Posted by [killenberg](#) on Fri, 19 Nov 2010 12:19:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Christoph,

you are right. I always thought about "I set transient true in order not to write the output." But "I set transient false to prevent the data from not being written" is not so good. We should change this. I think it historically came from the fact that the lcio::collection has this transient flag, which we just

repeated for the processor.

So you have another vote for "WriteOutputToStorage"

Cheers

Martin

Subject: Re: Inconsistencies with "Transience Flag"
Posted by [jabernathy](#) on Fri, 19 Nov 2010 18:44:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

That sounds like a good change.

Subject: Problem in track reconstruction
Posted by [kiewfer](#) on Sun, 09 Jan 2011 11:41:57 GMT
[View Forum Message](#) <> [Reply to Message](#)

I run ilcsoft v01-09 on e+e->WW->qqqq event.

I Tried to activate All Tracker digitizers, and track

```
===== track digitization and tracking ===== -->
<processor name="MyTPCDigiProcessor"/>
<processor name="MyVTXDigiProcessor"/>
<processor name="MyFTDDigiProcessor"/>
<processor name="MyETDDigiProcessor"/>
<processor name="MyLEPTrackingProcessor"/>
<processor name="MySiliconTracking"/>
<processor name="MyFullLDCTracking"/>
<processor name="MyV0Finder"/>
```

But it print a long list of:

THITSRT error: too many TPC hits in one row
and finish by:

TPCOBIN: NMAXROW too small This is FATAL: TPC Tracking in event Skipped !Increase
NMAXROW in my_tctrl.inc and recompile
TRKFND:Error in binning of coordinates

I found the file, but if I recompile it in local, I will not be able to send easily my jobs on the grid.

Can you do something in the official release?

The file seems to be this one:

/v01-09/MarlinReco/v00-18/Tracking/BrahmsTracking/src/f77/include/my_tcrl.inc

If you need my .slcio file I can send you his location on the grid

Thanks a lot for your help.

Subject: Re: Problem in track reconstruction
Posted by [aplin](#) on Mon, 10 Jan 2011 10:12:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

unfortunately it will not be possible to increase this for the official release, but one option would be for you to create your own modified MarlinReco plugin, upload this to the Grid and then copy this during your Job.

Alternatively you could skip this event. It is almost surely being caused by a rouge curler, which is just a technical problem for the Fortran code and not a general problem, so I would assume it will not produce and bias to your result. I assume you are not trying to overlay background?

Could you send be the event in question?

Cheers,

Steve.

Subject: Re: Problem in track reconstruction
Posted by [kiewfer](#) on Wed, 12 Jan 2011 14:30:36 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Steve,

First I assure you that I did not put overlay background.

Actually this bug occurs on about 20% of my events, so I will try to fix the problem instead of dropping these events.

So, I tried to move NMAXROW from 2000 to 10000, recompile the libMarlinReco.so and run on this one, changing the MARLIN_DLL env..

I still have the same problem!

If you want to have a look using my test file, the location on the grid is:
/grid/calice/SDHCAL/kieffer/eeWW/

The name of the actual couple of (slcio/xml)file I tried is:
file_eeWW_0.5Tev_ISR_ON_FSR_ON_10000evts.stdhep_skip_000_ILD_00fw_Dhcal_50_v07-02_740687262.slcio
file_eeWW_0.5Tev_ISR_ON_FSR_ON_10000evts.stdhep_skip_000_ILD_00fw_Dhcal_50_v07-02_740687262.xml

This file contain 50 evts, and the list of bad events is:
4,10,11,16,17,20,22,23,32,46,47

This slcio file has been generated using the first 50 evts of the following stdhep file (on same grid path):
eeWW_0.5Tev_ISR_ON_FSR_ON_10000evts.stdhep

Thanks for helping me.

Cheers,

Robert

Subject: Re: Problem in track reconstruction
Posted by [aplin](#) on Thu, 13 Jan 2011 15:00:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Robert,

the first thing would be to upgrade to v01-10 as there are problems with the TPC smearing in v01-09, in that it is not done. Nevertheless this shouldn't cause the problem you are seeing.

I have run over the 50 events in your file and I don't get the error message concerning NMAXROWS. We haven't had problems with this for a long time so I am quite surprised it has turned up now.

I have looked at the gear file and every thing seems to be OK there.

Could you send me you steering file so that I can check that also.

Cheers,

Steve.

Subject: Re: Problem in track reconstruction
Posted by [kieffer](#) on Thu, 13 Jan 2011 22:33:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

I asked my computing administrators to install a v01-10 of the soft.

As soon as I can I will try with this version.

Any way, I join this post my steering file.

Did you ran my file with v01-09?

Cheers,

File Attachments

1) [TrackerDigiOnly.xml](#), downloaded 339 times

Subject: Re: Problem in track reconstruction
Posted by [aplin](#) on Fri, 14 Jan 2011 09:27:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Robert,

I have run over the file using your steering file and do not see any problems.

I am a little worried that the problem maybe coming from the mixed build C++/F77.

Where is your installation, and how was it installed.

Which operating system is it installed on and which version of gcc and gfortran are used.

One thing that we could do is to add a couple of debug statements to tcobin.F. I have attached a copy of tcobin.F, if you could recompile using this we could perhaps get an idea where things are going wrong.

Cheers,

Steve.

File Attachments

1) [tcobin.F](#), downloaded 603 times

Subject: Re: Problem in track reconstruction
Posted by [kieffer](#) on Fri, 14 Jan 2011 11:18:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

I put on ftp the out.log and err.log files obtained when running with the new tcobin.F compiled version.

I still have the same bug but with more detailed debug info.

You could find those files there:
<http://www.ipnl.in2p3.fr/perso/kieffer/>

I wait for the information about compiling the software on our server because that was done by the administrator.

I will had a post as soon as I get the info.

Subject: Re: Problem in track reconstruction
Posted by [aplin](#) on Fri, 14 Jan 2011 15:14:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Robert,

from you output it looks very much like you have a version of MarlinReco which has been compiled with a different versions of gcc for the fortran code and the c++ code. We have seen something similar before.

If you are using gcc 4.x you should make sure you set FC=gfortran when building ILCSOft.

If you are using gcc 3.x then you should use FC=g77.

This should be enforced in v1-10 so it would be good for you to start using that one, and make sure that the version v1-09 you have is removed.

Cheers,

Steve.

Subject: Re: Problem in track reconstruction
Posted by [kieffer](#) on Tue, 01 Feb 2011 20:15:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

I tried the last version but I have a new huge problem, in most of the event I have this kind of errors:

```
[ MESSAGE "MySiliconTracking"] SiliconTracking -> run = 0 event = 1
**** CERN G100 PROB ERROR G100.2: X = -8.7142536230E+17 < 0
**** CERN G100 PROB ERROR G100.2: X = -8.7142536230E+17 < 0
**** CERN G100 PROB ERROR G100.2: X = -8.7142536230E+17 < 0
**** CERN G100 PROB ERROR G100.2: X = -6.8392784344E+17 < 0
```

My version of the cernlib is the 2006 (last one)

So I tried to launch a simple job on the grid to check if it's a local problem or not, using ilcsoft tarball.

In attachment you could find the four files I used to launch this job on grid.

Indeed I tried with multiple files ee->WW->qqqq ee->ZZ->qqmunu which I produced with mokka.

I got every time the same error!

It seems to be due to a bad typedef.

Did you all ready add this problem?

Thank's for your help.

Robert

Subject: Re: Problem in track reconstruction
Posted by [kieffer](#) on Tue, 01 Feb 2011 20:17:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

Sorry, I forgot to had the tarball.

File Attachments

1) [Test_run_ilcsoft.tar](#), downloaded 407 times

Subject: Re: Problem in track reconstruction
Posted by [aplin](#) on Thu, 03 Feb 2011 16:47:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Robert,

I guess things have progressed with your installation.

I have seen this error several times before.

```
***** CERN G100 PROB ERROR G100.2: X = -8.7142536230E+17 < 0"
```

You say it is due to a typedef? Could point to it?

Cheers,

Steve.

Subject: Re: Problem in track reconstruction
Posted by [kieffer](#) on Fri, 04 Feb 2011 21:12:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

I'm not a programming expert but it look's like the SiliconTracking processor send a 64 bits value to the cernlib who wait for a 32 bit value or something like that...

Because it's every time a very big negative value, like if the "sign" bit and a few others are changed?

If you know wich person is in charge of this processor, let me know because I'm not capable of handling this kind of tricky bug.

And it could be an issue for every one who want to digitise a mokka output.

As I said I tried with my local version, AND the ilcsoft tarball the result is the same.

Cheers,

Robert

Subject: Re: Problem in track reconstruction
Posted by [aplin](#) on Wed, 16 Feb 2011 09:47:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Robert,

Thanks for the pointers, I'll have a look and see what I can find.

Cheers,

Steve.

Subject: Problem with PerEventIPFitterProcessor
Posted by [Hajrah Tabassam](#) on Thu, 24 Mar 2011 12:02:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Experts

There is a problem which I am getting while I am using Marlin. I am trying to run the LCFIVertex package on my samples but getting this error:

```
[ VERBOSE "MyPerEventIPFitterProcessor" ] A runtime error has occurred :  
Icio::ReadOnlyException: LCCollectionVec::addElement  
[ VERBOSE "MyPerEventIPFitterProcessor" ] the program will have to be terminated - sorry.  
I tried to use LCFIVertex package without this processor and it is working fine.
```

Is there any suggestion how to solve this problem and second thing what is difference between the default IPVertex and the one reconstructed by PerEventIPFitterProcessor?

Cheers,
Hajrah

Subject: Re: MarlinTPC flagwords
Posted by [rosemann](#) on Fri, 03 Feb 2012 11:39:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

pushing this topic up again. This happens, when one blindly implements something and never tries it as foreseen.

I noticed, that the implementation I supplied was not able to be included into different compilation units, if they are to be linked together. The reason is simply the definition of statics.

I tried something else, by encapsulating the flagword definition into classes; but this doesn't make much sense yet. I'll try to come up with something more sensible.

In principle I see the usage like this:

```
// define some quality word  
int qualityWord = whatever;  
// use a defined function to change the status  
pulseflag::setSomeBit(qualityWord):
```

```
// and another one to see, whether a bit is set
bool hasPropertyX = pulseflag::hasPropertyX(qualityWord);
```

For explicitness, I'd like to encapsulate the bit pattern definitions into namespaces.

I'll post the code, once I've written it.

Comments?

Cheers,
Christoph

Subject: Re: MarlinTPC flagwords
Posted by [rosemann](#) on Fri, 03 Feb 2012 19:40:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

I redid it, and it's similar to the original version. Here the example for the pulseflag (it's also in the marlintpc namespace):

```
namespace pulseflag
{
    int const MULTIPLE_PULSE_CANDIDATE_BIT = 1;
    int const ANOMALOUS_SHAPE_BIT        = 1 << 1;
    int const OVERFLOW_BIN_BIT           = 1 << 2;
    int const UNDERFLOW_BIN_BIT         = 1 << 3;
    int const WAS_SPLIT_BIT              = 1 << 4;

    inline void setMultiplePulseCandidate(int& i)
    {
        i |= MULTIPLE_PULSE_CANDIDATE_BIT;
    }
    inline void setAnomalousShape(int& j)
    {
        j |= ANOMALOUS_SHAPE_BIT;
    }
    inline void setOverflow(int& k)
    {
        k |= OVERFLOW_BIN_BIT;
    }
    inline void setUnderflow(int& m)
```



```

{
  m |= UNDERFLOW_BIN_BIT;
}
inline void setSplit(int& n)
{
  n |= WAS_SPLIT_BIT;
}

inline bool const isMultiplePulseCandidate(int i)
{
  return i & MULTIPLE_PULSE_CANDIDATE_BIT;
}
inline bool const isAnomalousShape(int j)
{
  return j & ANOMALOUS_SHAPE_BIT;
}
inline bool const isOverflow(int k)
{
  return k & OVERFLOW_BIN_BIT;
}
inline bool const isUnderflow(int m)
{
  return m & UNDERFLOW_BIN_BIT;
}
inline bool const isSplit(int n)
{
  return n & WAS_SPLIT_BIT;
}
}

```

BTW: I removed the mysterious and superfluous "hitflag::EMPTY_CHANNEL" bit (there is also a "dead" and a "noisy" channel flag) and introduced the helpful "hitflag::IS_NEXT_TO_DEAD_CHANNEL" bit.

Example code is as of rev2967 in the PulseFinder and the RowBasedHitFinder.

Cheers,
Christoph

Subject: Questions/remarks on pathfinder

Posted by [grenier](#) on Wed, 29 Feb 2012 16:00:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

I'm trying to use pathfinder to do tracking in the Hadronic Calorimeter.
For the moment, I'm looking for straight lines.

1) As it is done now, pathfinder can't really find straight lines that are parallel to the z-axis. I'm solving this for the moment by rotating the calorimeter. To get pathfinder able to find tracks parallel to the z-axis, would it need to change deeply the Hough transform computation or just change the fit for the output track parameter ?

2) I've observed that sometimes, the result of pathfinder depends on the way the input vector of basicHit is sorted. Is it a sign that I haven't set the parameters of the finder properly ? Is it an expected or an unexpected behaviour ?

Thanks for your answers

Subject: Re: Questions/remarks on pathfinder
Posted by [rosemann](#) on Wed, 29 Feb 2012 16:20:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Gerald,

the main author of pathfinder is Isa Heinze; I'm not sure if she is already registered here. This week she's away on a conference, so I'll try to answer your questions.

Quote: 1) As it is done now, pathfinder can't really find straight lines that are parallel to the z-axis.

Yes, right. There is no reason for this in terms of the technique. But pathfinder has a tracking device in mind, in first order a TPC. The actual finding is done in two steps. The first finds a projection of a track in the plane perpendicular to z. And a track parallel to z simply doesn't exist in this projection, since it's only a point.

I can't think of a simple solution from the top of my head -- the track parameters will simply be wrong if you start to change the coordinate system. But you can find the straight lines nevertheless; since each track has its own collection of hits attached to it.

But I'm not sure how you want to use pathfinder -- as track finder or as track fitter?

Quote:2) I've observed that sometimes, the result of pathfinder depends on the way the input vector of basicHit is sorted.

Hmm, this shouldn't be. Only one possibility comes to my mind; related to the technical implementation -- if there a two "equivalent" track candidates, always the first one will be chosen. (I could go into greater detail if wanted.) Which one the first is depends on the ordering of the input hits.

In principle the method is global, in the sense that every hit is taken into account with the same weight and at the same time. The choice of finder parameters shouldn't affect this.

cheers,
Christoph

Subject: wrong track direction...? PLEASE IGNORE!
Posted by [jeans](#) on Fri, 16 Jan 2015 08:22:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

Subject: calo TrackState along wrong track direction...?
Posted by [jeans](#) on Fri, 16 Jan 2015 09:26:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

I have a single 17GeV electron in ILD, which make a nice track with TPC, vtx, sit, etd hits.

I'm interested about it's state at the calorimeter, so I use `getTrackState(TrackState::AtCalorimeter)`.

This track state occasionally seems to be at the wrong intersection of the helix with the calorimeter, as if the particle goes the wrong way along the trajectory.

Using the TrackState AtLastHit seems OK.

I'm using MarlinReco v01-08

Are there anythings I can test to clarify the situation?

Thanks,
Daniel.

Subject: Re: calo TrackState along wrong track direction...?
Posted by [jeans](#) on Mon, 19 Jan 2015 09:21:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

I have done some more investigation, and I think I have found the source of the problem.

It seems that the "CalcXingPointWith" method in ILDParallelPlanarMeasLayer.cc does not take account of the requested "mode" (i.e. forwards or backwards along the track) when calculating the intersection of the track with the surface. This deficiency has even been helpfully noted in the code:

```
"//SJA:FIXME: do we need to consider the mode here",  
to which the answer is, in my opinion, "yes"!  
(KalDet/v01-12)
```

the "createTrackStateAtCaloFace" method of MarlinTrkUtils essentially looks for an intersection of the track in each of the 8 planes of the barrel ecal front surface. If it finds one of these, it accepts it (taking the "best" one if there are more than one); if not, it goes on to look at the endcaps. (MarlinTrk/v01-11)

The track I had problems with entered the ECAL just in the endcap; however a (wrong, backwards) intersection with the barrel had already been found, so it did not look for the correct endcap intersection.

Cheers,
Daniel.

Subject: Re: calo TrackState along wrong track direction...?
Posted by [gaede](#) on Thu, 25 Jun 2015 07:14:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Daniel,

thanks a lot for your detailed bug report. We will try and address the problem (for the next ilcsoft release).

Can you point me to an example LCIO file where this rare problem occurs ?

Cheers, Frank.

Subject: Re: calo TrackState along wrong track direction...?

Posted by [jeans](#) on Fri, 26 Jun 2015 01:51:34 GMT

[View Forum Message](#) <> [Reply to Message](#)

Dear Frank,

I attach an lcio file containing one event.

One track in the MarlinTrkTracks collection (the one with $\omega = -3.19071e-05$) has the following TrackState->getReferencePoints:

TrackState::AtLastHit : (1791.98, 319.048, -2273.11)

TrackState::AtCalorimeter: (-1847.35 -219.503 2335.41)

In this case, the AtCalorimeter one has been taken in the wrong direction along the helix.

Regards,
Daniel.

File Attachments

1) [event_with_wrong_trackstate.slcio](#), downloaded 682 times
