TRAIL Status Report

Matt Needham

- Quality Checking / Splitting of Sicb
- General status of the code
- Problems Encountered with the new version of Gaudi
- Remaining design issues
- To do....
- Simulation and Digitizations
- Summary

Quality Checking/ Sicb Splitting

In Sicb there exist two routines:

axthisto.F

tfmonitor.F

that produce histograms capable of monitoring tracking output $\Rightarrow$ use these as basis for a ‘standard’ set of $\sim$ 20 plots to look at (and what they should look like).

Started to check the split version of SICB (so far with little success).

Persons responsible for defining plots /checking: Matt Needham
Person responsible for checking SICB splitting: Marcel Merk

In Gaudi have ‘monitoring’ algorithm that produces similar plots $\Rightarrow$ intend to put sample output on the web.

Person responsible: Matt Needham

General Status in Gaudi

- Framework exists for general track reconstruction
- Track Fit has
  - $\sim$ same functionality as in SicB (and works)
  - Still need to implement use of Velo imformation
  - Documentation required
- Fitting software in a $\sim$ releasable form
- Track following studies/implementation well advanced but not releasable yet
- Track finding studies started
- Slight divergence in fitting/pattern recognition code

Problems Encountered with new version of Gaudi

LOTS !

Stupid Problems with:

- passing job Options to subalgorithms
- need to provide code for static linking of our library - not documented

More serious problem:

ObjectVector has serialize member $\Rightarrow$ requires default constructer.....but this doesn't work if you have an interface class (say for tracks) unless you have a dictionary as in ROOT.

(No solution - we use private copy of LHCbEvent package)

Horrible problems because HTL histograms not protected against NaN:

- Program dies with obscure messages
- Sometimes only after 100 s of events $\Rightarrow$ problem as can't easily skip to event xxx in Gaudi

Problems with the histogram service

- convert all our histograms $(30 - 40)$ to new style of booking
- Why do the histograms have to have both a unique string and number
- What does the histogram interface class gain you - The user only ever sees it in the header files and the call to fill
- Please remove the call to HPRNT at the end of the job

Horrible problems because HTL histograms not protected against NaN:

- Program dies with obscure messages
- Sometimes only after 100 s of events $\Rightarrow$ problem as can't easily skip to event xxx in Gaudi

Features needed in Gaudi

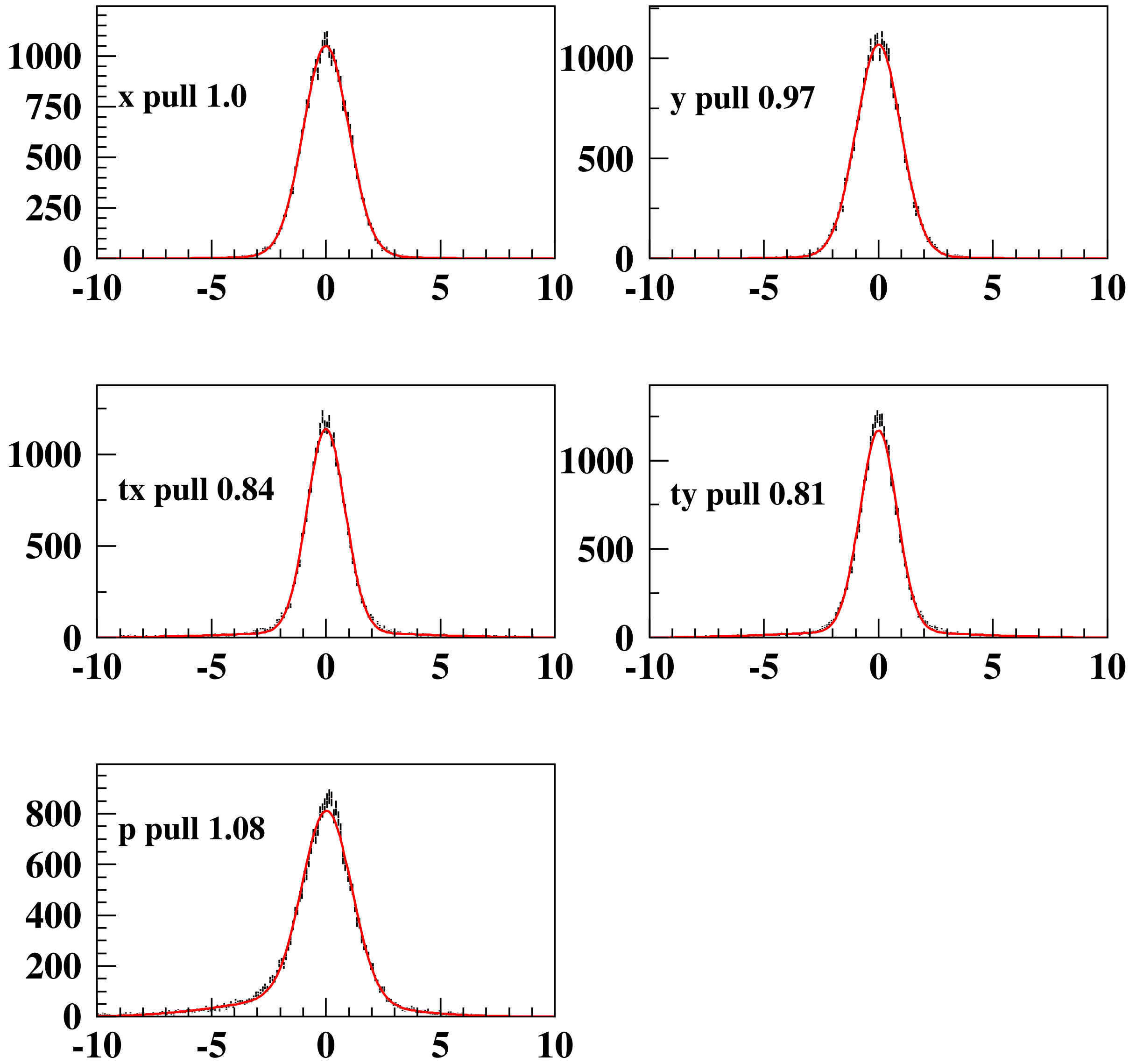
- Ability to run from event xxx to yyy
- Run over a list of tapes
- If you try to run over 1000 events and there are 998 events on the tape Gaudi goes to ZFATAL. Can this be protected against ?
- Some Algorithms count numbers during a job $\Rightarrow$ can we have a stream to send these to ?

Some design features

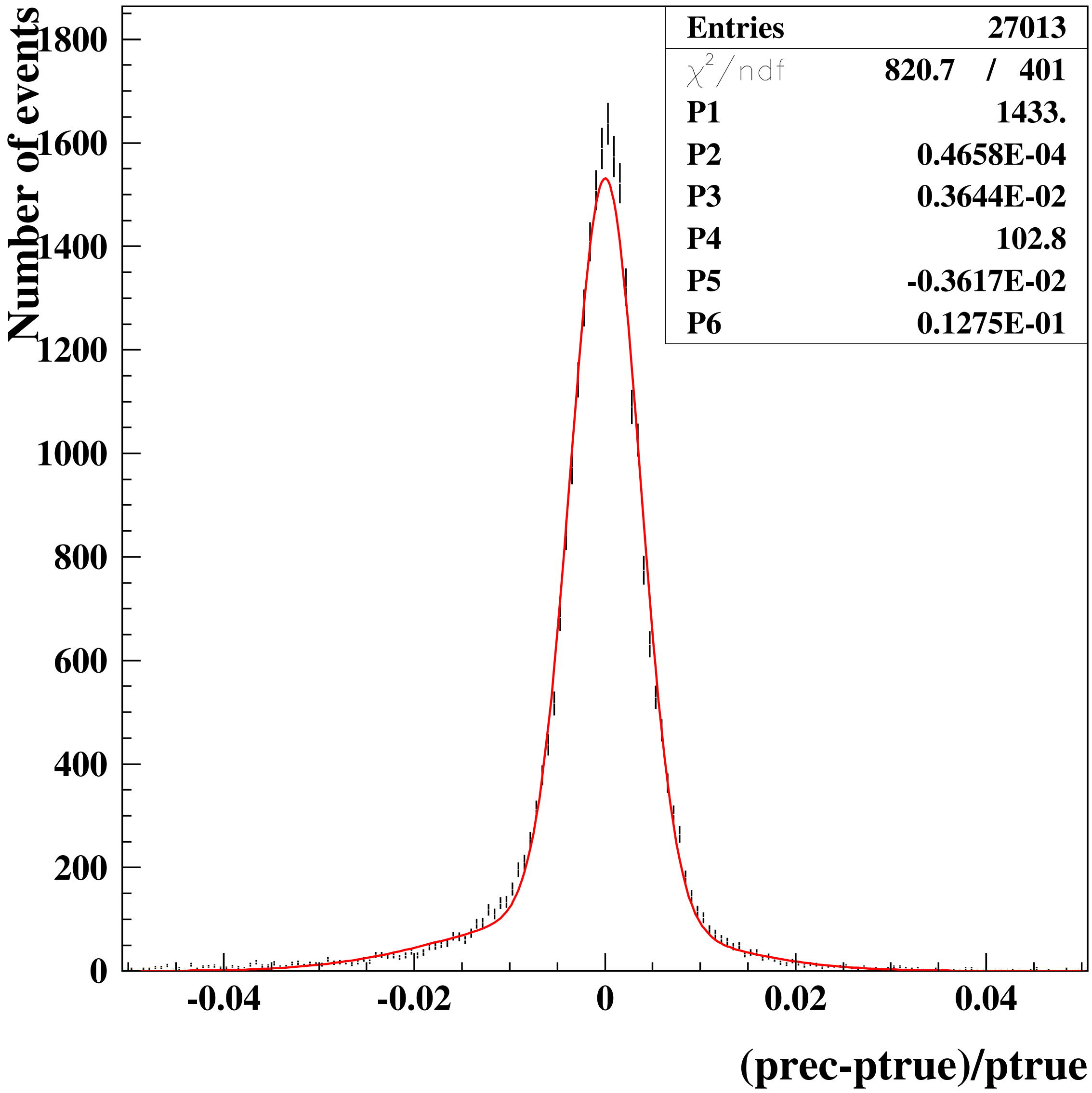
- Tools vs Algorithms
  - At the moment treat extrapolators, update step of kalman filter as subalgorithms
  - But e.g. Extrapolator needed by more than one algorithm (track fit/ pattern recognition/ RICH)
  - Tools rather than algorithms ?
- Store hits in Layers (Gaudi ObjectVectors)
  - Every event create/delete containers
  - Better(?) to create once at intialization time, empty every event
- Associators / Issue of dealing with Monte Carlo truth
  - Where do SmartRefTables fit into this ?
  - Should there be a common strategy ?
- At present if you want to group all Monte Carlo station entrance/exit points by Particle you have to do it yourself. Should this be provided as standard. We need the information like this alot.
- jobOptions not idiot proof

Pulls:

- 1000 inclusive b events
- Double Gaussian fit to each distribution
- Quoted pull is the smaller σ of the two Gaussian



Momentum Resolution



Core resolution 3.6×10^{-3}

To do (for the fit)

In order of priority:

- Produce documentation
- Tidy code up/ Resolve differences that have arisen with pattern recognition
- Move from ‘private wrapping’ of digitizations to the ‘standard’ one
- Use Velo information in the C++ fit
- Re-implement wrapped FORTRAN code in C++

Simulation and Digitization

The outer and inner tracker simulation/ digitization is very simple in SICB. For example:

- Simple box like geometry at simulation stage
- Digitizations
  - Stereo layer geometry not correctly treated
  - No proper r-t relation used
  - Propagation delay along wire not implemented

For the TDR we strongly feel we should improve both. We propose the following:

- Improve simulation
  - Implement at least the layers in simulation
  - use GEANT 3 ?
- Implement improved digitizations in C++

This solution works if

- SICBDST dies on the timescale of $\sim$ a few months.
- SICBMC's replacement is $\sim$ at least a year away.

Summary and Conclusions

- Design and inplementation well advanced
- Track fit in C++ works
- Need to provide some documentation
- With some tidying-up/solution of some problems could to be made publically available
- Pattern Recognition code not yet in a releasable state

- Time for a Design review ?


TRAIL