Central Tracker Physics Channel Benchmark

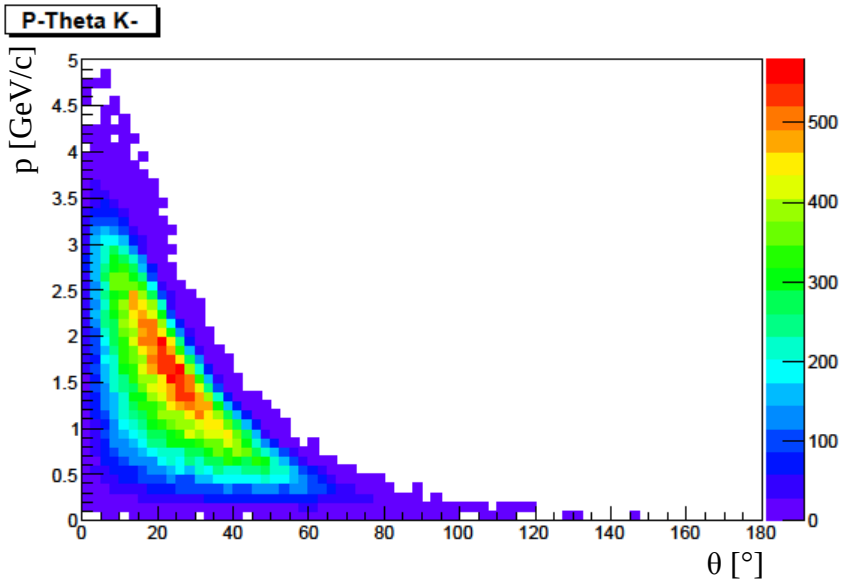
$\bar{p}p \rightarrow \Psi(3770) \rightarrow D^+D^- \rightarrow K^-\pi^+\pi^+ K^+\pi^-\pi^-$

Technical Aspects

Darmstadt, 5.9.2011 Marius C. Mertens

My apologies for the following text-heavy slides, but this topic is hard to be dealt with in pictures…

On Thursday there will be more pictures. Like this one:



Simulation Overview

- Channel: $\bar{p}p \rightarrow \Psi(3770) \rightarrow D^+D^- \rightarrow K^-\pi^+\pi^+ K^+\pi^-\pi^-$
- Beam momentum: 6.5788 GeV/c
- Simulated data on the PandaGrid
  - *Signal samples for STT (run934oldnocu, PandaRoot 12725)*
  - *Signal samples for TPC (run984, PandaRoot 12933)*
- Steps
  - *Simulation*
  - *Digitization*
  - *Reconstruction*
  - *PID*

  } Software experts, full data sets available on the Grid

  - ***Analysis (common for both CT options)***

Analysis Overview

0: **Total count** of signal events as input

1: Events within **geometric acceptance** of the central tracker. All six MC signal tracks must touch the CT volume. → **Detector acceptance**

2: **Reconstructable events**. The reconstructed tracks are cleaned by positive MC PID information. After that they must contain (at least) two oppositely charged Kaons and two oppositely charged pairs of two Pions.

3: **Events hitting the CT**. The STT reconstruction code does a global reconstruction MVD+STT+GEM which is not yet ready for the TPC. To allow comparison, all tracks without an STT hit are discarded (Same events as in (2) for TPC). After that they must contain (at least) two oppositely charged Kaons and two oppositely charged pairs of two Pions.

4: **Events with D+ and D- candidates**. Same events as in (3)

5: **Events with D+ and D- candidates within 1.5 GeV mass window**

6: **Events surviving vertex fit**. (Best candidate with GlobalChi2 < 18)
→ **Secondary vertex resolution**

7: **Events with D+ and D- candidates within 0.5 GeV mass window**

8: **Events with Psi candidates**. Same events as in (7)

9: **Events surviving 4C fit**. (Best candidate with GlobalChi2 < 18)
→ **Efficiency, mass resolution**

Analysis Aspects: Getting Started

- Getting started
  - *First source of information: Klaus‘ charmonium tutorial:* *http://panda-wiki.gsi.de/cgi-bin/view/Computing/PandaRootCharmoniumTutorial*
  - *Nice template to get something running*
  - *Nice set of the most important tools in the analysis: How to build lists, most important fitters, how to get the information out*
- At that stage I had something running which was already quite close to what I wanted
- Advancing from this point, I wanted to see which tools other are available, but as I didn‘t find an exhaustive list (at least not in a prominent place), I ended up searching in the source code
- **Suggestion to help people in an „after tutorial“ state: Reference lists of**
  - *How candidate lists can be filled an manipulated*
  - *Which selectors, fitters etc. exist and how they are used*
  - *In general:* ***A reference manual about the objects you encounter during analysis***

Tools used…

- **General remark**, applies to nearly all tools below: documentation seems to be sparse and/or not reflecting the current state of development. An exhaustive list of actions and a description of the designed behaviour would be very helpful.
- **Event Reader**
  - *Nice interface, efficient to use*
  - *Which option strings are available for selection?*
- **Track Candidates**
  - *At which point is the momentum calculated?*
  - *Any way to influence that?*
- **Candidate Lists**
  - *Which properties of the candidates within the list are preserved when building new lists?*
- **Selectors**
  - *Very efficient to select tracks within candidate lists*
  - *A list which selectors exist and how they are used would be nice*

Tools used… (continued)

- **Fitters**
  - *What happens when multiple fitters are applied?*
  - *What happens when fitting decay trees with several levels?*
  - *Vertex fitter: Use of mass constraint resulted in much longer processing time and many warnings about non-invertable matrices*
  - *4C fitter: Seems to compress a decay tree to only one level*
  - *A quick way to fit a whole candidate list and retrieve the best fit or a fitted candidate list would be nice*
- **Interface to Monte Carlo data**
  - *Method: Access of the MC data via indices, casting of TClonesArray members → not elegant, but worked in the end*
  - *There are more elegant solutions present, but I did not find good documentation*

Not directly analysis related, but funny:

```
TVector3 mymomentum = mymctrack->GetMomentum();
```

→ **Segfaults sometimes**

```
TVector3 mymomentum;
Double_t x;
Double_t y;
Double_t z;
std::stringstream strstrx;
std::stringstream strstry;
std::stringstream strstrz;
strstrx << mymctrack->GetMomentum().x();
strstrx >> x;
strstry << mymctrack->GetMomentum().y();
strstry >> y;
strstrz << mymctrack->GetMomentum().z();
strstrz >> z;
mymomentum.SetXYZ(x, y, z);
```

Only change → works!

Grid experience

- Mostly working as expected
- Large jobs usually more efficient than on a local machine

- alien interface stalls after some idle time
- Using less → „v“ → vi → „:wq“ → changes are not saved
- submit job.jdl parameter sometimes results in „Error getting the file parameter from the catalogue”
- Sometimes the Grid gets „hickups“ → Storage not accessable, many ERROR_IB/ERROR_SV, jobs „WAITING“ forever

- Testing parameter to have one split of a run processed very fast to see if it is working
- Error_v summary (currently the info is discarded)
- Automatically collecting data out of jobs