DyTER – Dynamic Track and Event Reconstruction

(Hyperon) Tracking for PANDA

Status Report

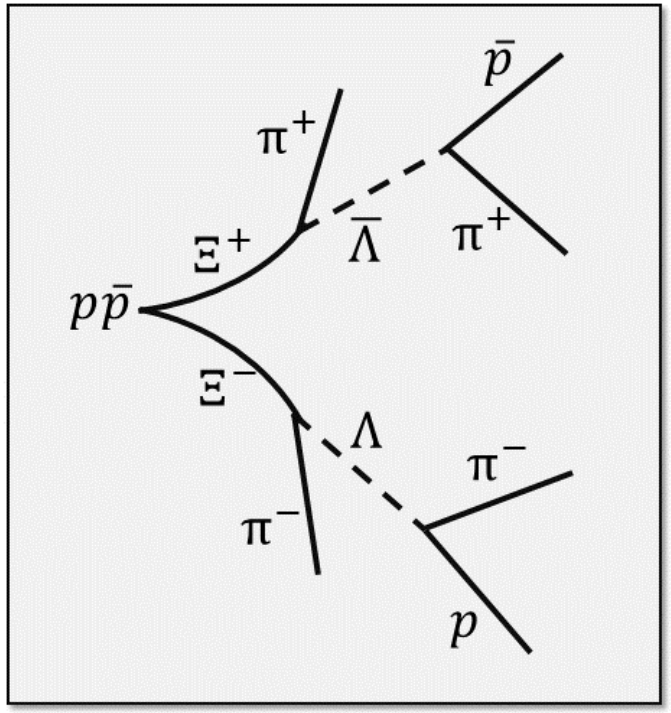
of Uppsala Hyperon Group

Content

- Motivation
- DyTER – Guidelines and Basic concept
- Current Issues
- Current focus and results
  - Event generation
  - T0 / event determination
  - IdealTrackFinder
  - EventCombiner
  - SttCellTrackFinder

Motivation

- Hyperon physics: Key to strong interaction
- Complex event topology
- Tracks originating far from collision point
- Sophisticated track reconstruction required



DyTER – **Dy**namic **T**rack and **E**vent **R**econstruction

Guidelines:

- Focus on hyperons (displaced vertices)
- Capitalize on PANDA's unique DAQ system

Basic concept:

- Use sophisticated informations after track reconstruction for event building
  - Generate tracks and vertices dynamically from continuous data stream
  - Group data to events based on track and vertex information
- Requires modular approach to code development

First simulation studies have been performed and presented

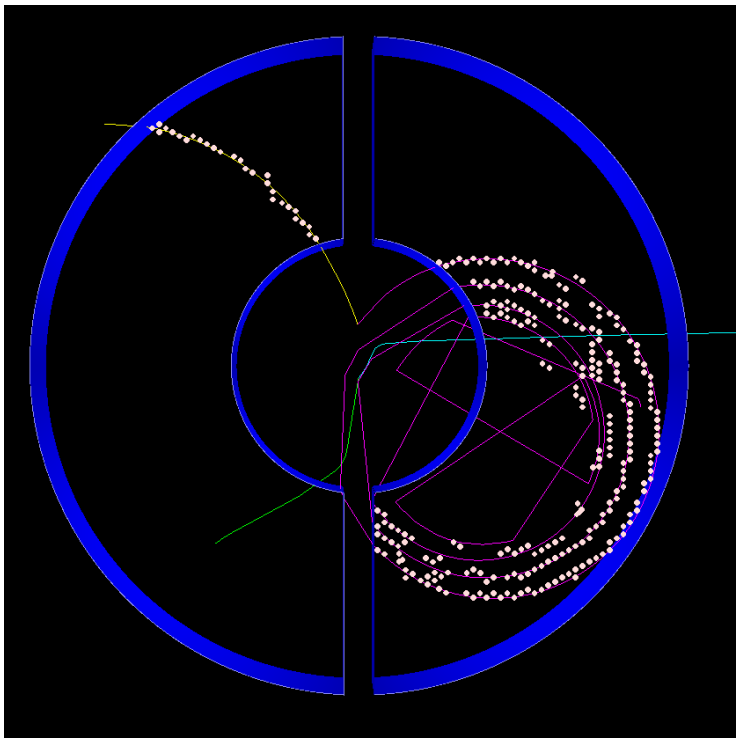
- "Prototype of an online track and event reconstruction scheme for the PANDA experiment at FAIR"
  - Collaboration Meeting 2017/1:

First steps: Evaluate Current status

- Modularized
  - Dynamical use/skip /repeat parts to adapt to input/requirements
  - Adapt / use existing algorithms
    - GabEventBuilder
- CellTrackFinder
  - Valuable option for DyTER / hyperons
  - event mixing
  - bunches of events
  - Ghost suppression
  - Performance for displaced vertices
- Implement missing modules
  - T0 determination
  - Event building

Studies on signal pattern in the tracking detectors

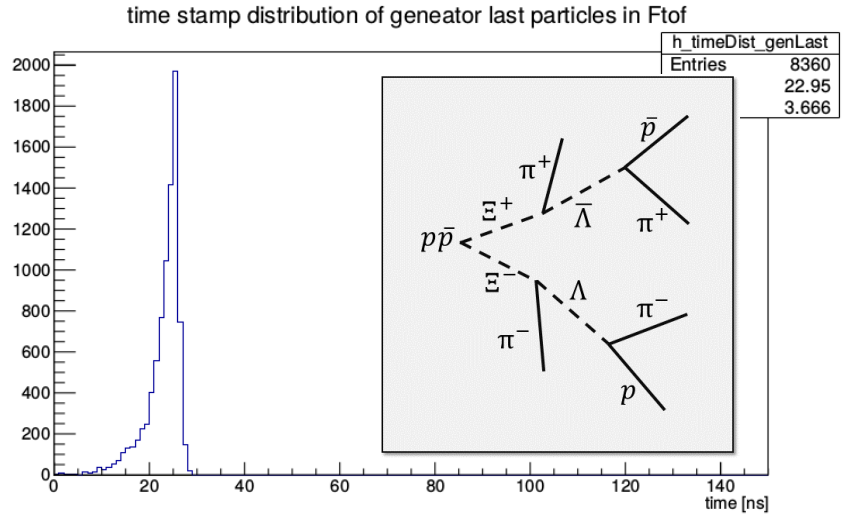
- Improve / implement reconstruction algorithms
  - Displaced/secondary tracks, hyperons
- "Investigations of detector signatures from $\Lambda\bar{\Lambda}$ and $\Xi\bar{\Xi}$ events"
  - By Jenny Regina

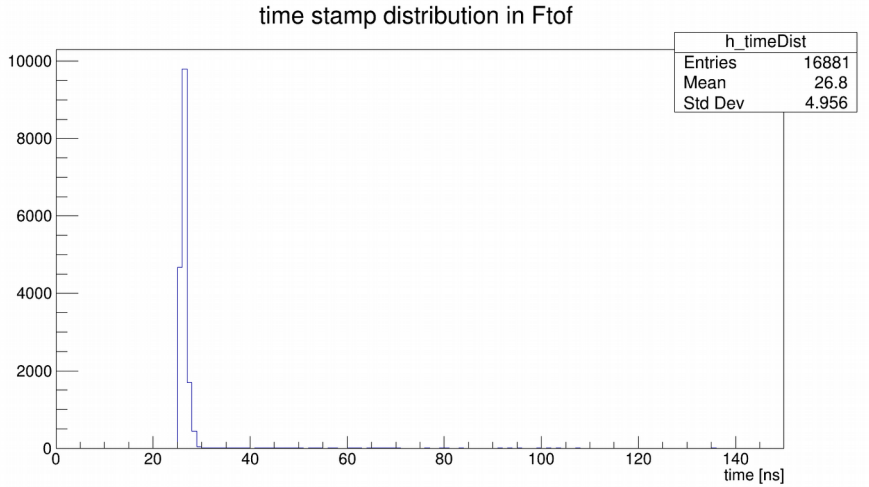


$p\bar{p} \rightarrow \Lambda\bar{\Lambda} \rightarrow p\pi^- \bar{p}\pi^+$

Event generation / EvtGen

- Time propagation bug
  - EvtGen → Geant3/4
    - Decay time of (long lived) particles not propagated
    - Especially problematic for hyperons
    - T0 determination and PID based on TOF Counters
  - Fixed in Trunk



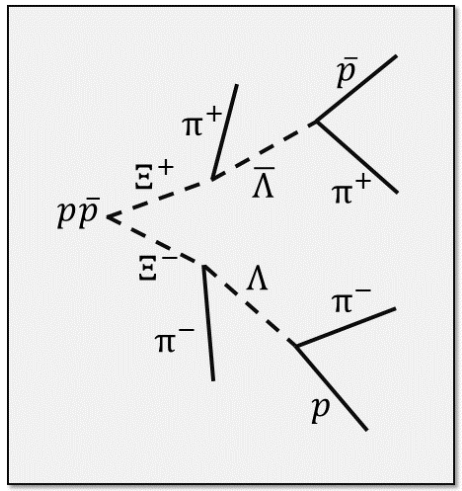


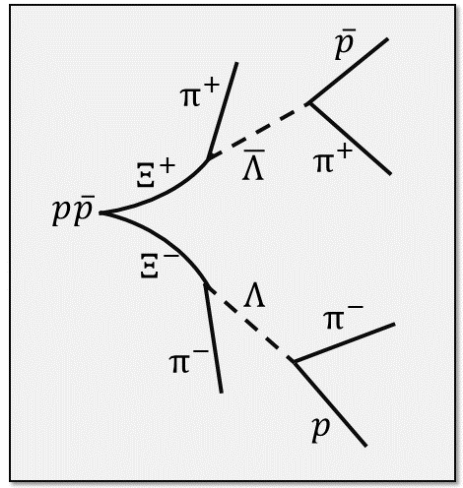
Event generation / EvtGen

- No realistic transportation in EvtGen
  - → Decay handled by Geant4
- → Geant 4: User defined Decay modes
  - For rare decay modes
  - Solution presented in an online Computing Meeting
    - https://panda-wiki.gsi.de/foswiki/pub/Computing/Minutes02May2017/2.5.2017_teammeeting.pdf

.

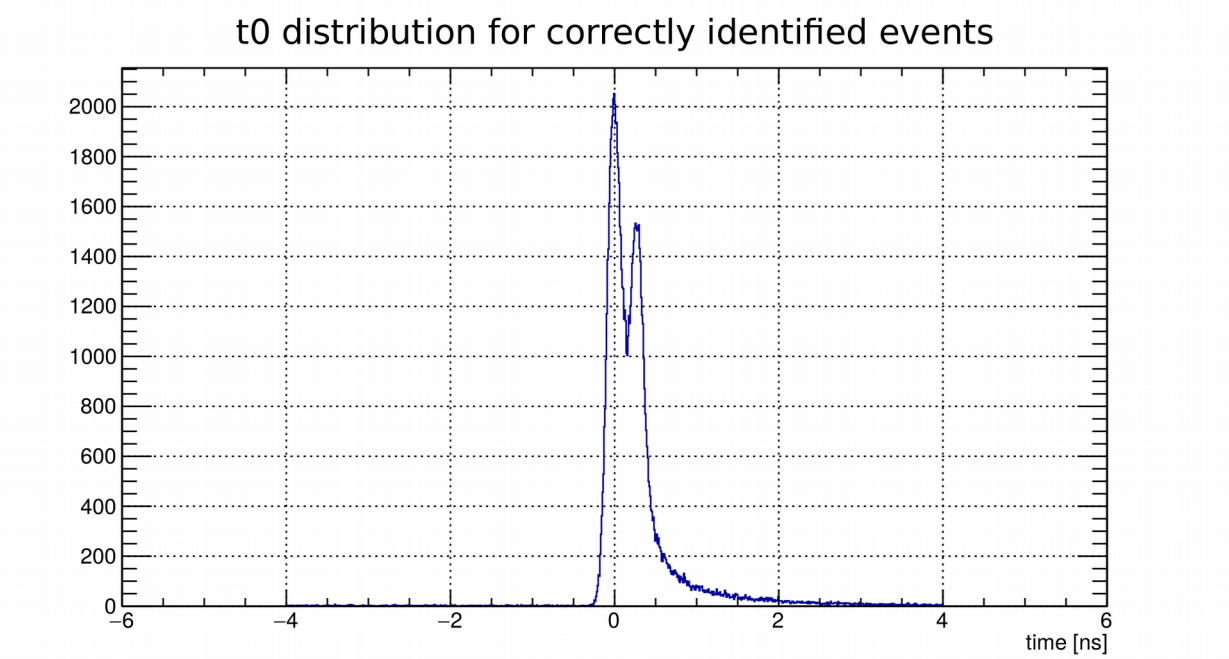
```
// simulation Master macro
int sim_master(Int_t nEvents = 10, TString  SimEngine ="TGeant4", Double_t BeamMomentum = 1.642)
{
// decay file for the Generator
TString inputGenerator = "llbar_fwp_1.642.DEC";
// Decay mode macro for the transport system
TString decayMode = "/ABS_PATH_TO/UserDecayConfig.C";  //Absolut path is needed!

 PndMasterRunSim *fRun = new PndMasterRunSim();
fRun->SetInput(inputGenerator);  // input file for the generator decays
fRun->SetName(SimEngine);
fRun->SetParamAsciiFile(parAsciiFile);
fRun->SetNumberOfEvents(nEvents);
fRun->SetBeamMom(BeamMomentum);
fRun→SetUserDecay(decayMode);
```





T0/event determination

- Basic algorithm implemented in trunk
  - Based on SciTil/ BTof and FTof
  - Usable for event and time based simulation
  - QA macros are provided
    - Trunk/macros/qa/eventDet
  - For more informations see presentations
    - Computing session, Panda LIX Collaboration Meeting, Dec 2016



- Also tested for $p\bar{p} \rightarrow \Lambda\bar{\Lambda} \rightarrow p\pi^- \bar{p}\pi^+$
  - Efficiency still quite low
    - due to forward peaking distribution only a few hits in Tof Counters
  - Work ongoing

IdealTrackFinder

- Fixed some Bugs
  - Multiple (wrong) FairLinks to MCTracks were stored
  - Default "NoFtsFunctor" accepted all Tracks in target spectrometer
    - Be aware if you use the release Version
    - May use the "OnlySttFunctor"

- Updated the interface
  - Now compatible with any input Branchname
  - Now Supports any Sub-detector
  - https://panda-wiki.gsi.de/foswiki/pub/Computing/Minutes03April2017/IdealTrackFinder_update.pdf
  - Supports semi time based data
    - No direct time based data

.

PndEventCombinerTask

- To simulate time based data
  - Create bunches of events
  - simulate a first "event building"
- Capable to work with any Branch at any stage
- Test ongoing
  - See SttCellTrackFinder
  - Upload to the trunk in the next weeks

```
//###################### Set all tasks #####################################
// ------------------------------------------------------------------------

PndEventCombinerTask* combi2 = new PndEventCombinerTask();  // a task to combine the STTHits of 2 Events
combi2->AddInputBranch("STTHit");    //Set all input Branch for which you want to generate combined Events
combi2->SetNEvents(2);               //Set the Number of Events Which should be combinded
combi2->SetPersistence(kTRUE);       // Persistence set to KTRUE if the combinded events should be stored in t
                                                          // if set to kFALSE they are just bu1
fRun->AddTask(combi2);              // the OutputBranchName will be comb_2_STTHit

PndSttCellTrackFinderTask *cellTrackFinder_2 = new PndSttCellTrackFinderTask(); // an other instance off the Tracker using the combined and stored Data
 cellTrackFinder_2->SetPersistence(kTRUE);
 cellTrackFinder_2->AddHitBranch("comb_2_STTHit");           // it should use the combined STT hit of 2 events
 cellTrackFinder_2->SetOutBranchNamePrefix("comb_2");        // the defoult output is already used by the Tracker above, so we can/must specify an pref
 fRun->AddTask(cellTrackFinder_2);
```

SttCellTrackFinder

- Updated interface
  - Works now with any input branch name
    - "PndSttHit"
  - You can specify an output prefix
  - Works now also with bunches of events ("semi time based")
- Evaluate the efficiency for (semi) time based data
  - Work ongoing

Tracking QA

- Update needed
  - for flexible input
    - Branch names
    - Event bunches
  - For time based data

General Interface problems

- Current Tasks
  - Input and Output Branch names are hard coded
    - Not compatible to new event builder tasks/ time based
  - Mandatory event based structure
    - can't even test event mixing
    - . . .
  - No use of FairLinks but old System
- Costs a lot of time adapting this later!

- We need common rules for the interface
  - Tasks
    - EventBuilder, Tracker ,QA Tasks ,PID
  - Stored data
    - FairLinks
    - Copy constructor, = operator
- Mandatory for new Tasks
- May update important Tasks
  - e.g. TrackingQA Task