Subject: Mvd Development Status
Posted by Ralf Kliemt on Mon, 14 May 2007 15:05:34 GMT

View Forum Message <> Reply to Message

Hello everyone,
This thread will cover the development communication for the PANDA Mvd in PandaRoot. I'll try to update this first post to have a summary what is done and what is next. Comments on the code itself shall be the central part of the discussion here.

Here is a list of items [09/2008], which have to be done:

Digitization

  implement root based geometry file   [DONE]
  channel mapping   [DONE]
  frontend emulation   [DONE]
  parameter definitions & database   [DONE]
  noise emulation   [DONE]
  timestamp ordering for digis   [RK]
  multiple hit digi merging   [RK]

Cluster Reconstruction

  alignment   [-]
  channel back-mapping   [DONE]
  cluster finding   [DONE]
  output data structure   [partly done/additions needed] (genfit compatible)
  Ideal reconstruction   [RK (done) is it needed]
  timeordered clusterfinding   [RK]

Tracking/Pattern recognition

  ideal track finding   [DONE]
  conformal mapping?   [-]
  riemann algorithm   [DONE]
  local PID   [DONE]

Event mixing

  event time distance sampling   [-]
  electronics delay & clock   [-]

More has to come during the discussion.

The following macros are a bit old. They will be revised.

Mvd MC simulation (Geant Transport)
Toggle Spoiler
macro/mvd/runMvdSim.C

```
gSystem->Load("libMvd");
...
CbmDetector *Mvd = new MvdDetector("MVD", kTRUE);
Mvd->SetGeometryFileName("MVD14.root");
Mvd->SetVerboseLevel(0);
fRun->AddModule(Mvd);
```

The output data are MvdPoint objects (inherited from CbmPoint).
To access them manually you might want to have a look into macro/mvd/anaMvdSim.C.

Mvd Digitization
Toggle Spoiler
The Mvd Digitization is almost done. There is only a nice Parameter handling missing and the hard coded values have to be removed. To use the digitization with both Pixel an Strip detectors put the corresponding hitproducers into your macro.

```
  //
========================================================================
==
  // ======              Hit Producers              ======
  //
========================================================================
==

  // -----   MVD Strip hit producer   ----------------------------------------
  double  topPitch=0.004921,
       botPitch=0.0042969,
       orient=TMath::Pi()*(0.5),
       skew=TMath::Pi()*(-0.5),
       threshold=0., noise=0.;
  int topFE = 10, botFE = 4, nrFEChannels = 128;
  MvdStripHitProducer* mvdHitProd
    = new MvdStripHitProducer(topPitch, botPitch, orient, skew,
                   topFE, botFE, nrFEChannels, threshold, noise);
  mvdHitProd->SetVerbose(iVerbose);
  fRun->AddTask(mvdHitProd);

  // -----   MVD Pixel hit producer   ----------------------------------------
  Double_t  lx=0.01, ly=0.01, threshold=000, noise=00;
  MvdHybridHitProducer* mvdPixProd = new MvdHybridHitProducer(lx,ly,threshold,noise);
  mvdPixProd->SetVerbose(iVerbose);
  fRun->AddTask(mvdPixProd);

  //
========================================================================
```

```
==
 // =====              End of HitProducers                =====
 //
======================================================================
==
```

There will be later a combined task for everything, including the parameter readout from an external ascii file. The output are MvdDigiPixel and MvdDigiStrip data containing mainly Channel and Frontend numbers and Charge.

Mvd Reconstruction
Toggle Spoiler
There is a factory switching the different types of Reconstructions. So far there is only the "Simple" Clusterfinder which is under construction now.
However there is the Monte-Carlo-Clustering there which generates MvdClusters directly from the MvdPoints and which applies a gauss smearing. Here is how you shall use it:

```
  gSystem->Load("libgenfit");
  gSystem->Load("libtpc");
  gSystem->Load("libtpcreco");
  gSystem->Load("librecotasks");
  gSystem->Load("libMvd");
  gSystem->Load("libMvdReco");
  ...
  // -----  Ideal Cluster Producers   ---------------------------------------
  MvdIdealRecoTask* mvdirec = new
MvdIdealRecoTask(0.01,0.01,0.005);//(SigmaX,SigmaY,SigmaZ)
  mvdirec->SetVerbose(iVerbose);
  fRun->AddTask(mvdirec);
```

The libraries have to have this particular order until librecotasks doesn't depend on the libtpcreco anymore. Following the tpc example I made a second library, libMvdReco, which depends on libMvd, libgenfit and librecotasks.
The output are MvdCluster and the three parameters are the smearing in the local sensor system with z being the sensor thickness.

Mvd Tracking
Toggle Spoiler
The ideal tracking is started. But we need still some communication with the tracking group and we have to see how to put this together wit the GEANE stuff.

```
  gSystem->Load("libgenfit");
  gSystem->Load("libtpc");
  gSystem->Load("libtpcreco");
  gSystem->Load("librecotasks");

  gSystem->Load("libMvd");
  gSystem->Load("libMvdReco");
```

...

```
  MvdIdealTrackingTask* mvdmctrk = new MvdIdealTrackingTask();
  fRun->AddTask(mvdmctrk);
```

Thoughts on PID
Toggle Spoiler
Tobias Baldauf, a student at Dresden, made a task for doing some particle identification on dE/dx for only Mvd Monte-Carlo Points. It is like a MC truth based trackfinding combining the energyloss information.
The corresponding task is MvdIdealPidTask and the Output is MvdPidCand objects.

```
  gSystem->Load("libgenfit");
  gSystem->Load("libtpc");
  gSystem->Load("libtpcreco");
  gSystem->Load("librecotasks");

  gSystem->Load("libMvd");
  gSystem->Load("libMvdReco");
```
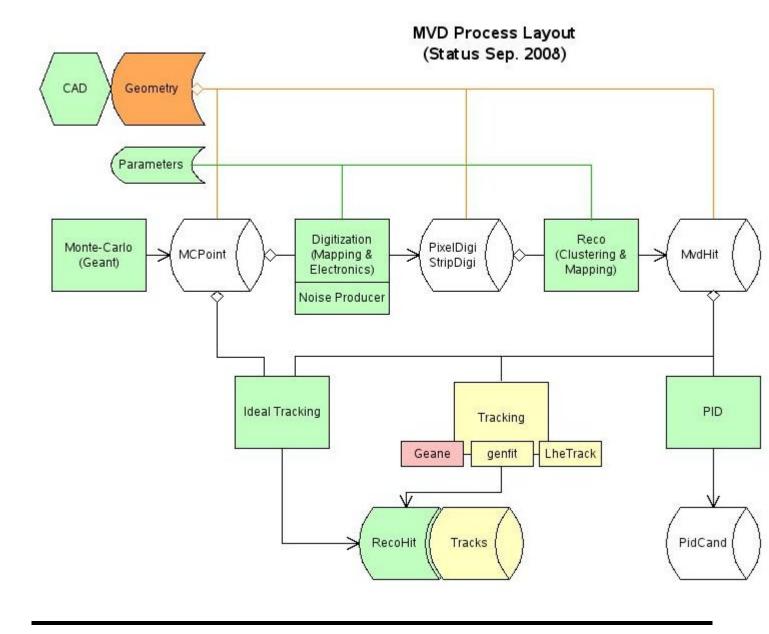
...

```
  Double_t trunc=0; //switch for different treatments of calculating the dE/dx mean for a Track
  MvdPidIdealTask* mvdpid = new MvdPidIdealTask(trunc);
  fRun->AddTask(mvdpid);
```

Have fun, Ralf.

File Attachments
1) Mvd_software_layoutstatus.jpg, downloaded 1553 times

MVD Process Layout
(Status Sep. 2008)