Λ^0 reconstruction

- aimed to a search for the H dibaryon -

FANTASTIC BEASTS AND WHERE TO FIND THEM

JG|U JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Belle II

Bianca Scavino, Uni Mainz

Tracking meeting, 9/06/17

Outline

- Analysis strategy
- Local modification implemented
- Definition of the variables
- Pull distributions
- Conclusions

Analysis Strategy

Goal: compare performances of lambda reconstruction with V0 objects and reconstructDecay

Strategy for the study: is to see what is delivered to *analysis level*

input: 2 particle lists

Lambda0:V0

```
fillParticleList('Lambda0:V0', '0.9 < M < 1.3', True)
vertexKFit('Lambda0:V0', 0.0)
applyCuts('Lambda0:V0', '1. < M < 1.2')
```

Lambda0:dec

```
reconstructDecay('Lambda0:dec -> pi-:all p+:all', '0.9 < M < 1.3')
vertexKFit('Lambda0:dec', 0.0)
applyCuts('Lambda0:dec', '1. < M < 1.2')
```

- selection: MC matching (mother & daughters)
- creates and fills 4 new particle lists

Lambda0:V0only | Lambda0:deconly | Lambda0:bothV0 | Lambda0:bothdec

- define variables and fill histograms

output: root file

Local Modification

interest: pull d_0^{V0}, pull d_0^{dec}

- d_0->TrackFitResult

d_0 from a particle p1:

- p1->getTrack()->getTrackFitResult()->getD0();

when a v0 object is created:

```
m_v0s.appendNew(std::make_pair(trackPlus, tfrPlusVtx),
                std::make_pair(trackMinus, tfrMinusVtx));
```

- tracks: original
- TrackFitResults: updated

→ $d_0^{V0} = d_0^{dec}$

default basf2

We modified locally Particle class:

add TFR as a member -> direct access

```
const TrackFitResult* getTFR() const
{
  return m_trackFitResult;
}
```

Variables Definition

```
Particle p1 = particleList->getParticle(i);
MCParticle mc_p1 = p1->getRelated<MCParticle>();
MCParticleInfo mcInfo_p1(mc_p1, magField);

TrackFitResult* TFRdec = p1->getTrack()->getTrackFitResult(type);

TrackFitResult* TFRV0 = p1->getTFR();
```

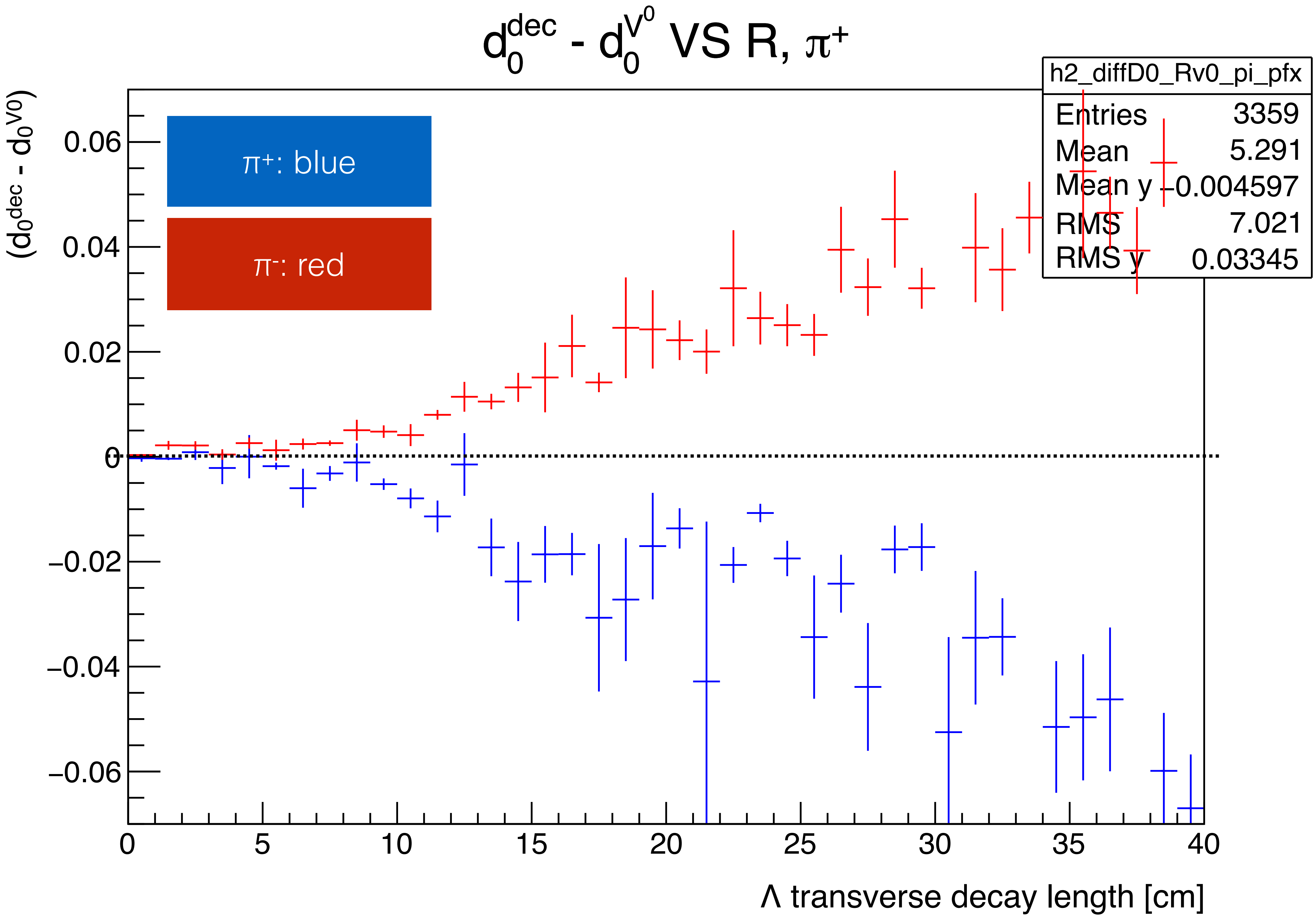
d_0^{dec} = TFRdec->getD0();
d_0^{V0} = TFRV0->getD0();
d_0^{gen} = mcInfo_p1->getD0();

σ_{d0}^{dec} = TMath::Sqrt(TFRdec->getCovariance5()[0][0]);
σ_{d0}^{V0} = TMath::Sqrt(TFRV0->getCovariance5()[0][0]);

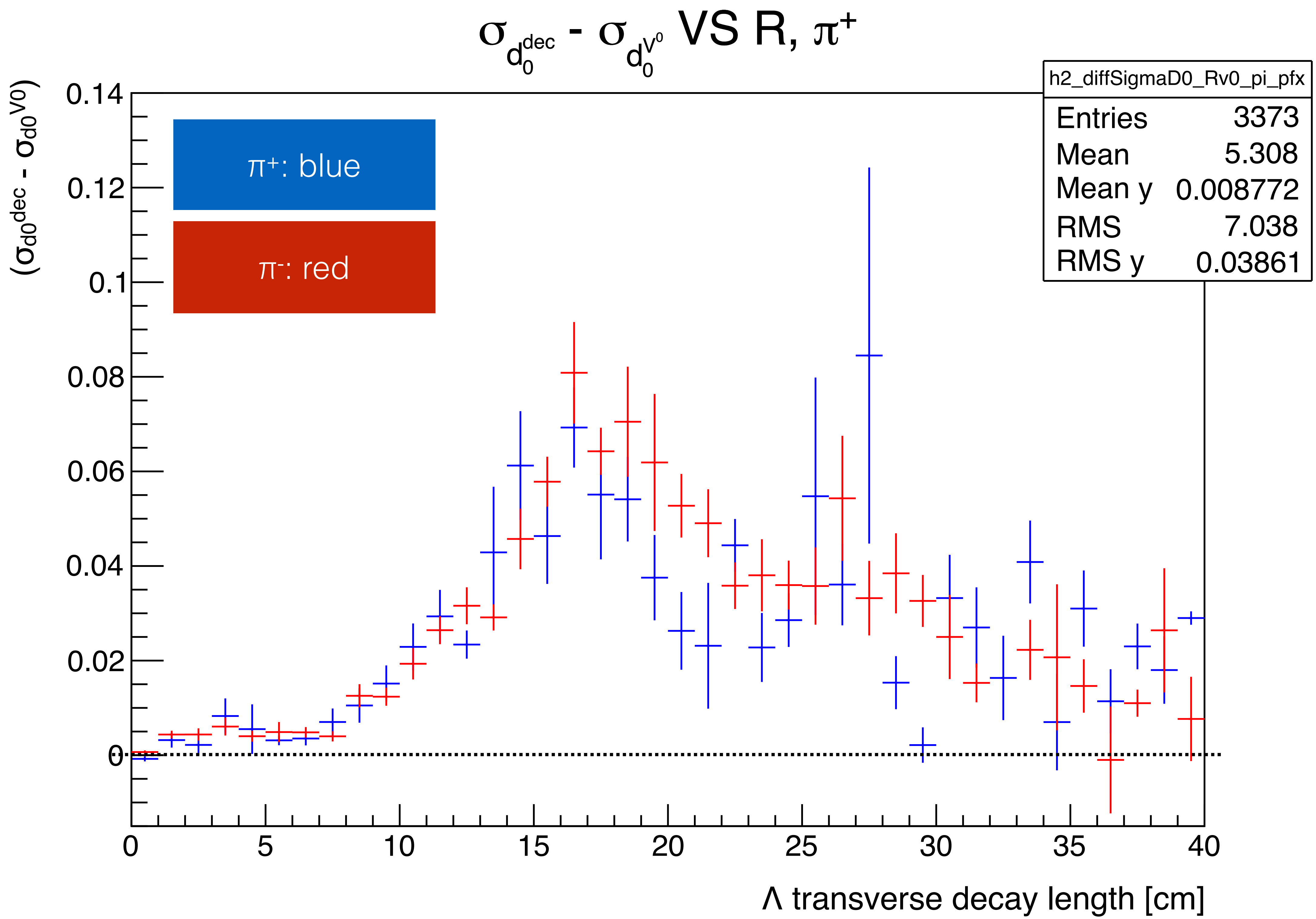
$pull_{dec} = (d_0^{dec} - d_0^{gen}) / \sigma_{d0}^{dec}$

$pull_{V0} = (d_0^{V0} - d_0^{gen}) / \sigma_{d0}^{V0}$

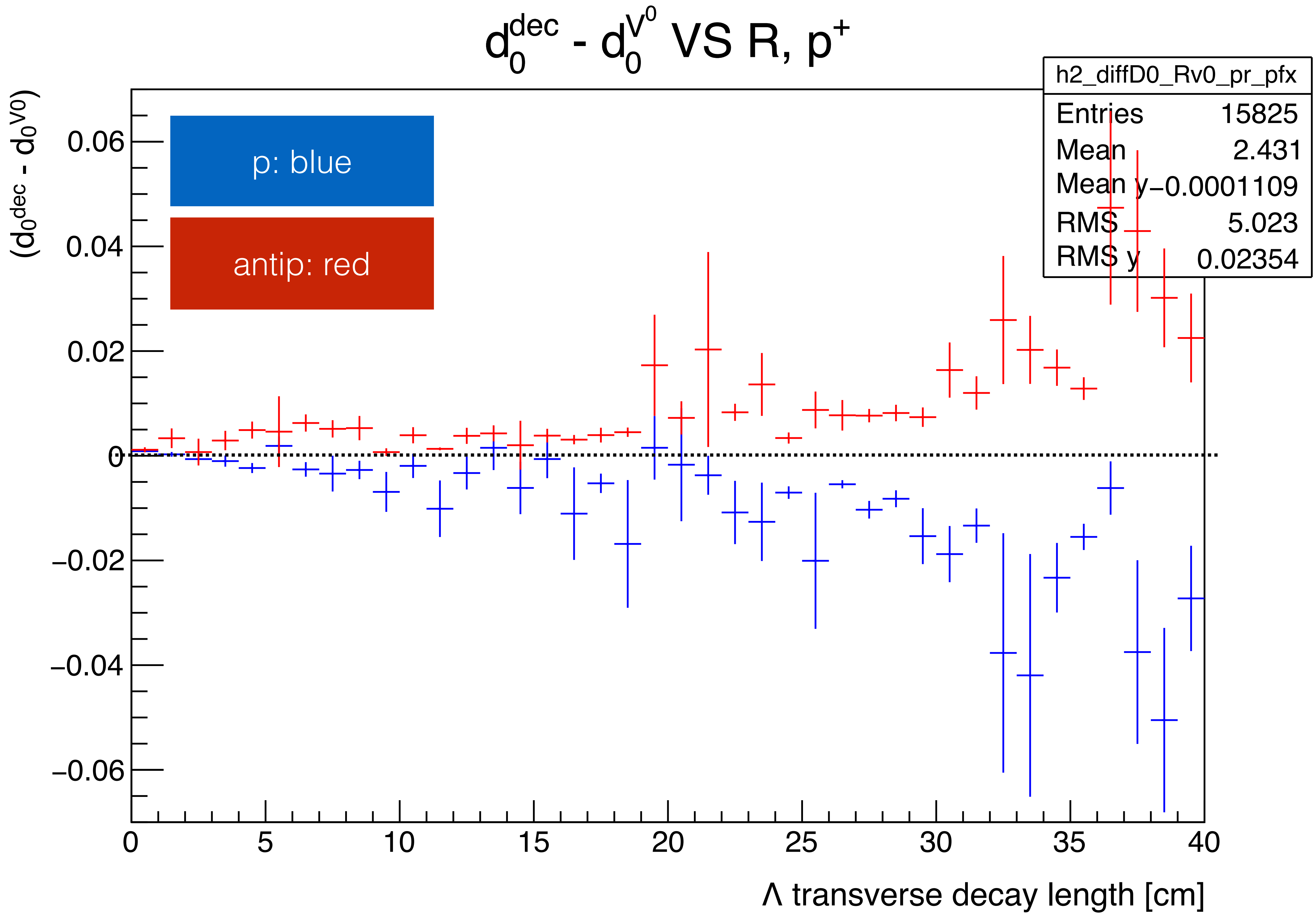
$(d_0^{dec} - d_0^{V0})$ VS R, π



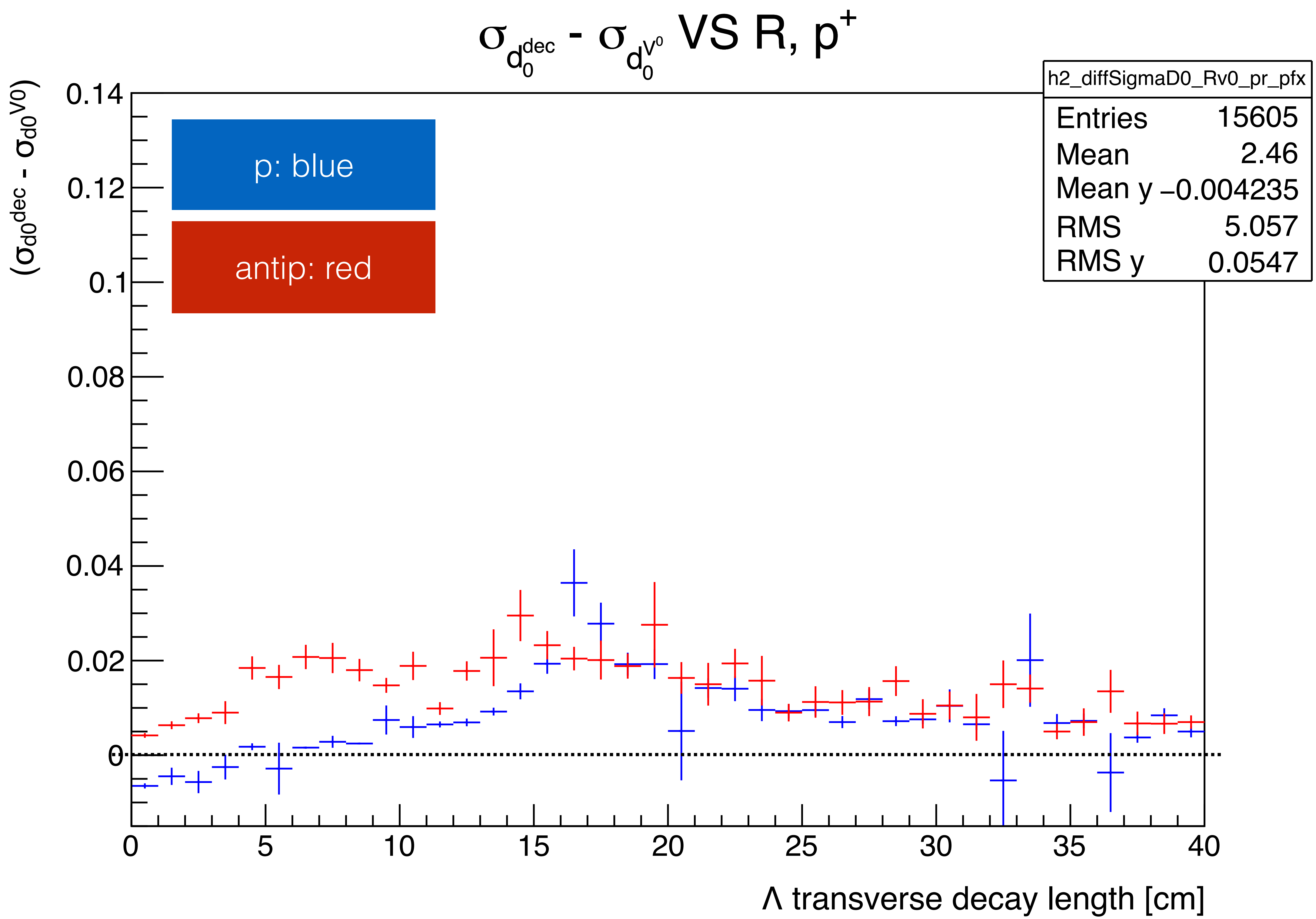
$(\sigma_{d0}^{dec} - \sigma_{d0}^{V0})$ VS R, π



$(d_0^{dec} - d_0^{V0})$ VS R, p

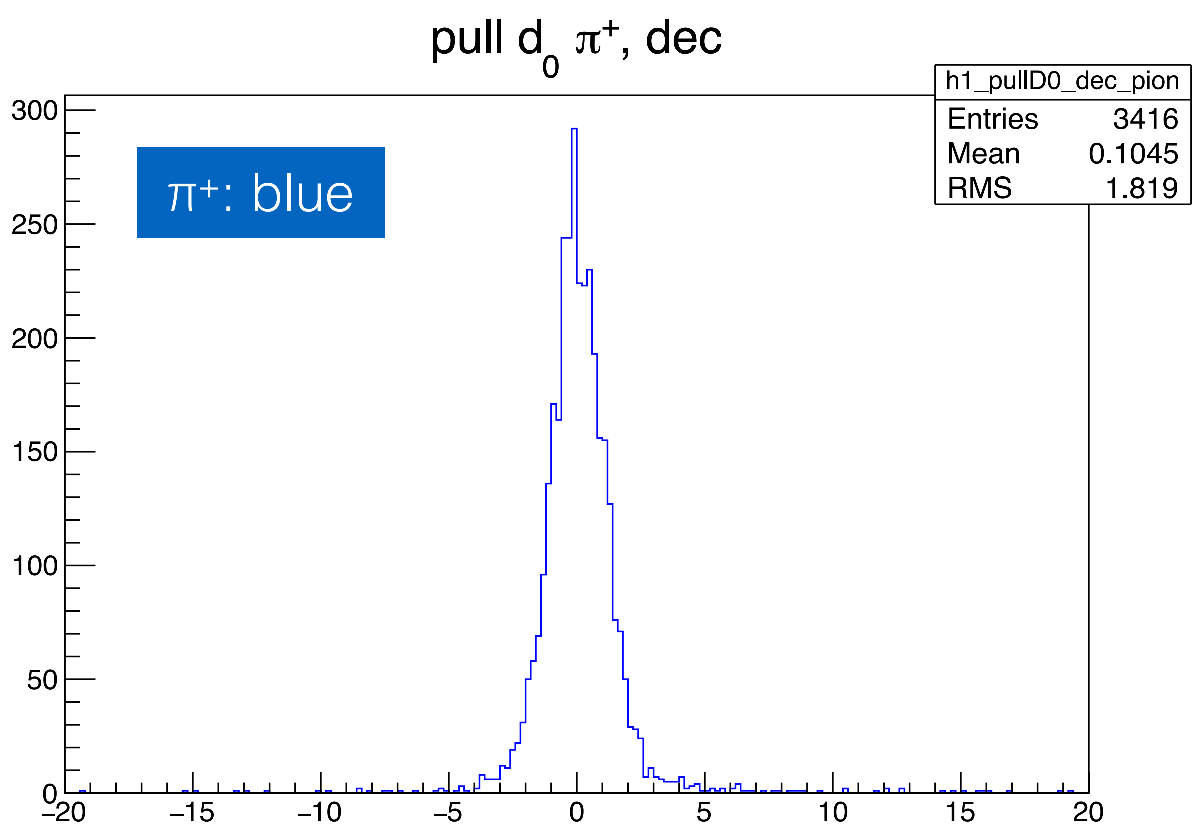


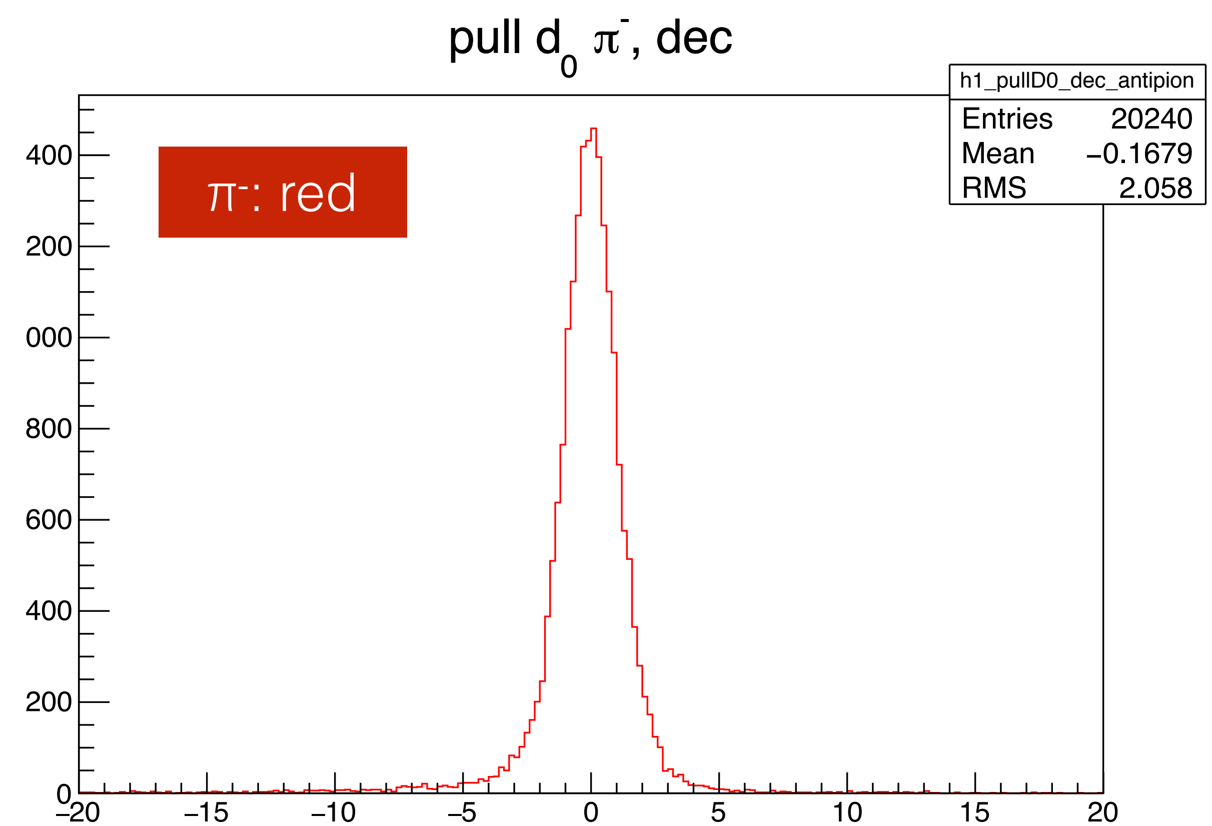
(σ_{d0}^{dec} - σ_{d0}^{V0}) VS R, p



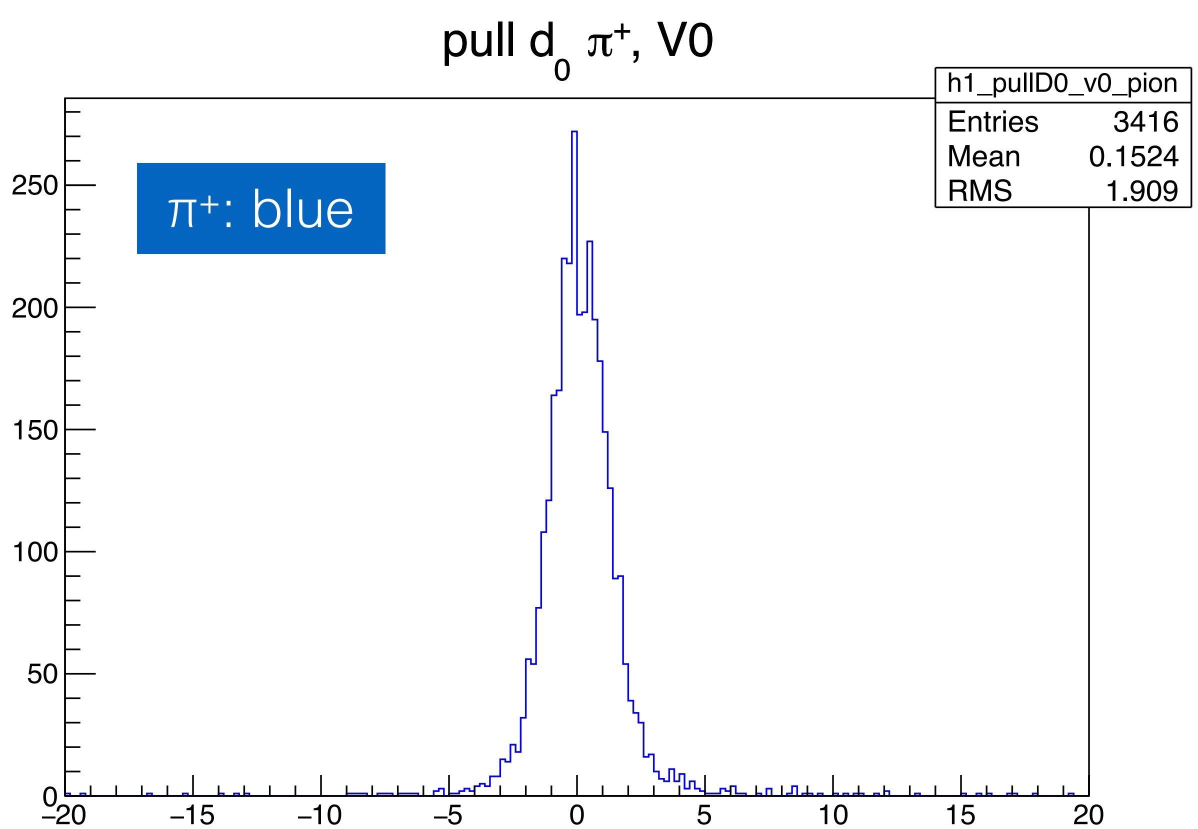
pull distribution, π

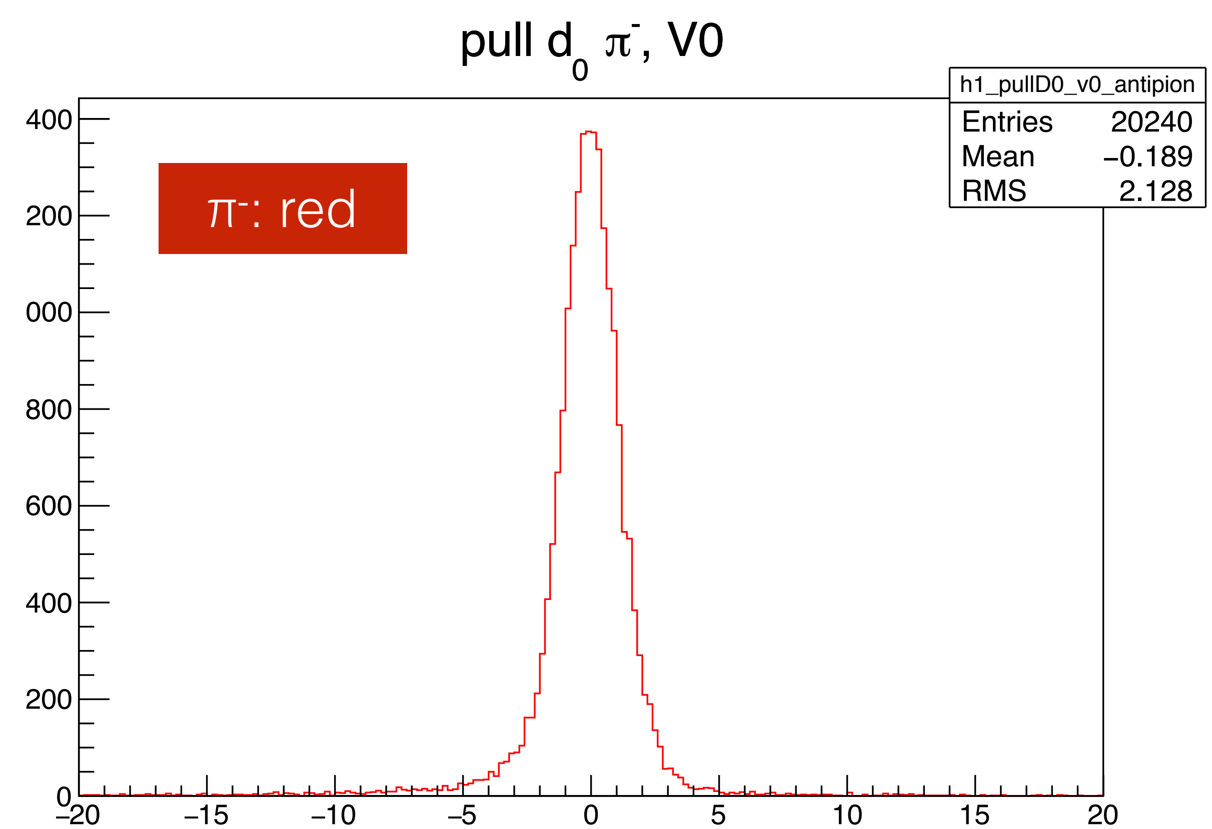
dec





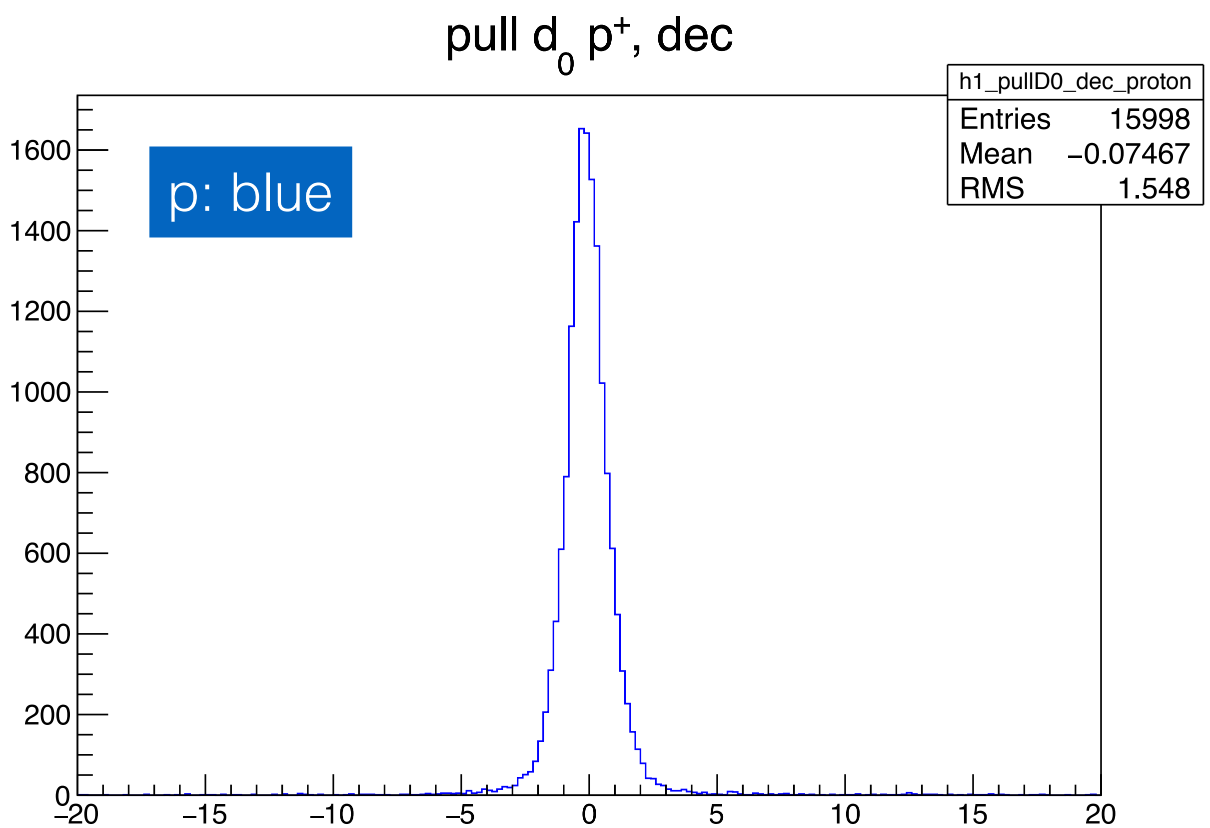
V0

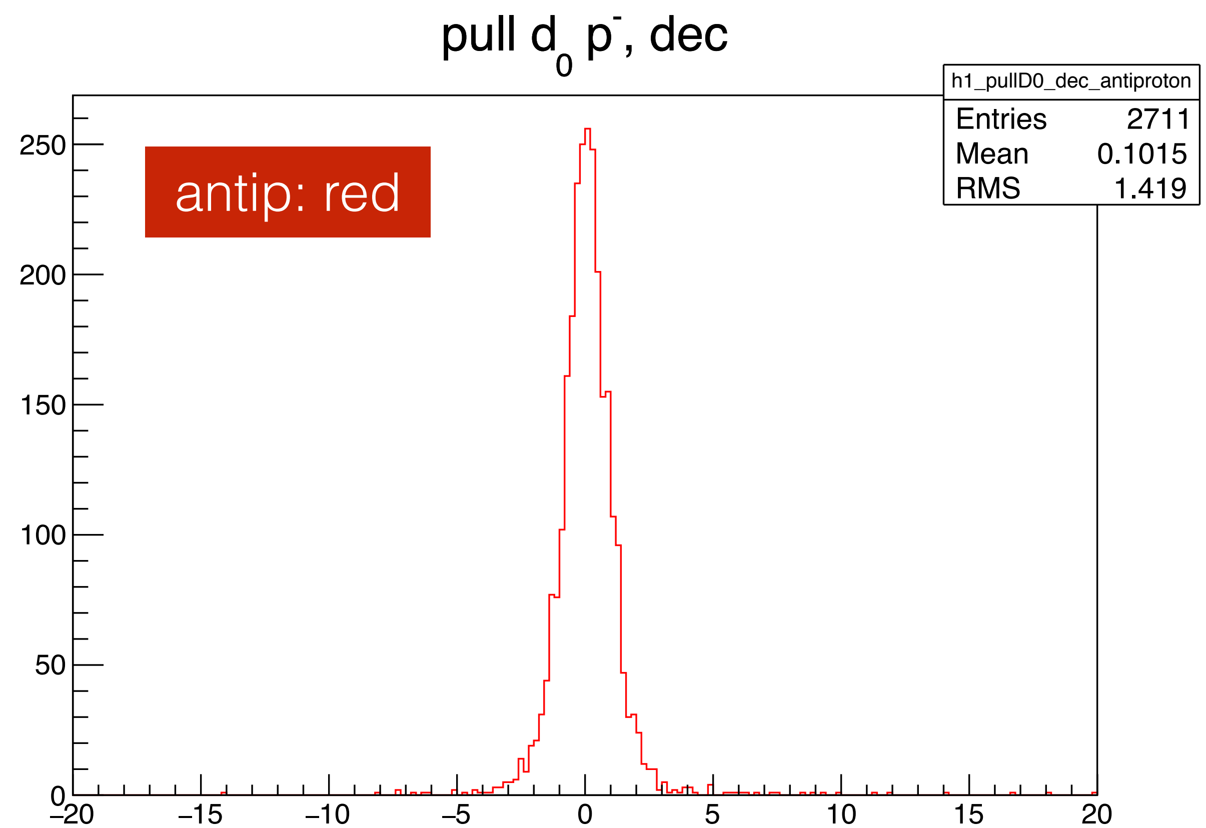




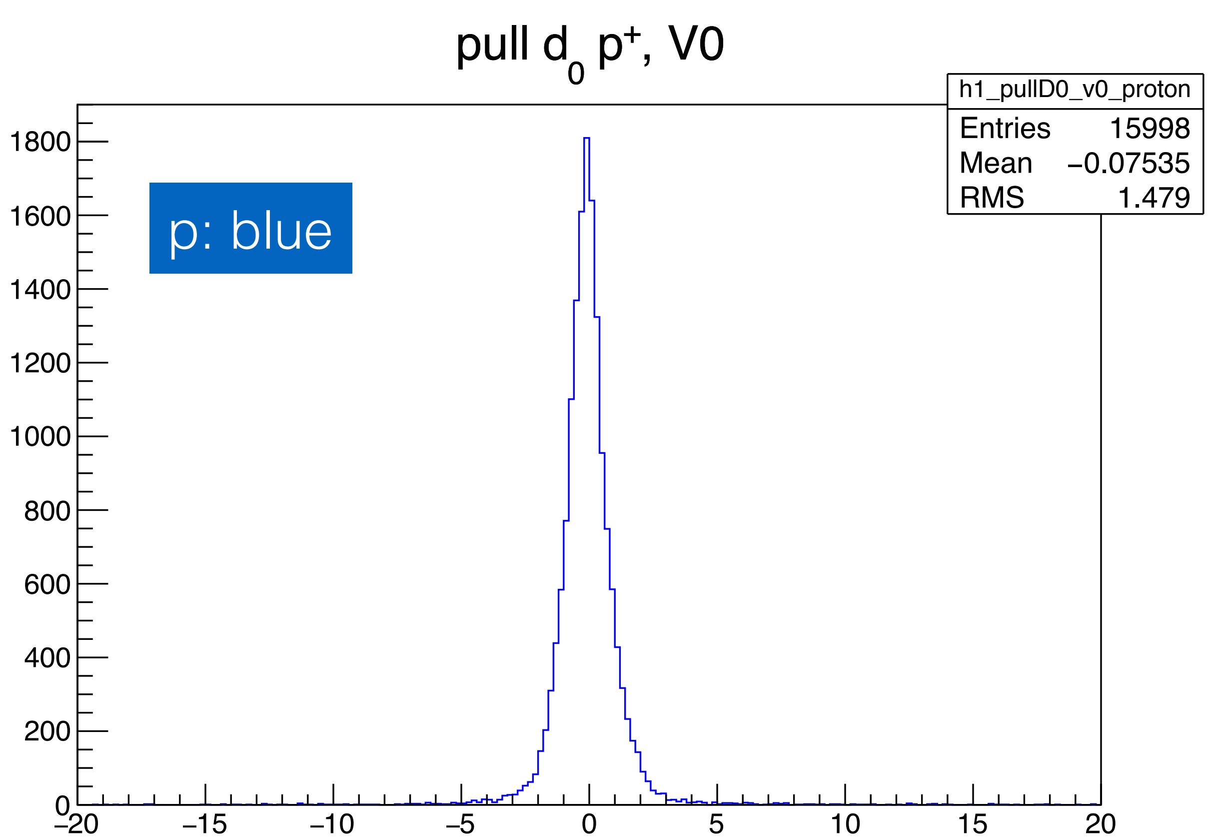
pull distribution, p

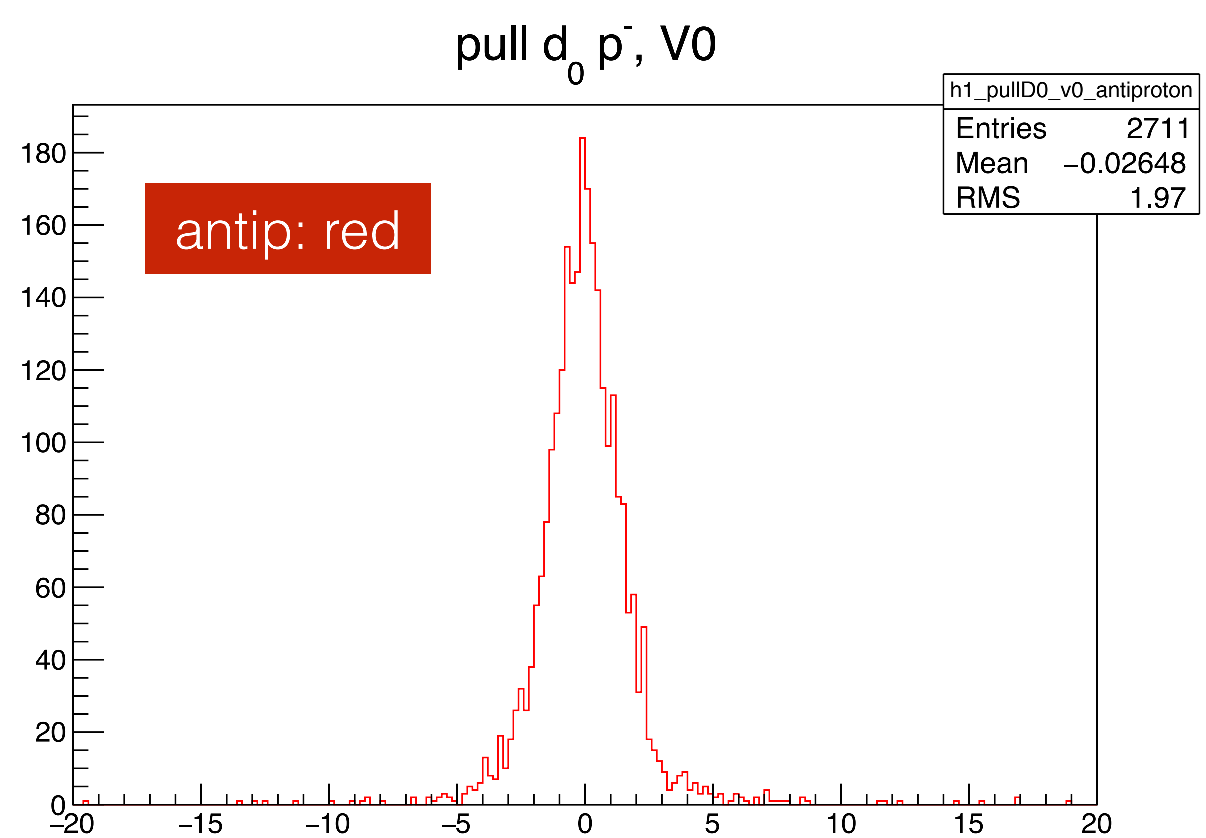
dec





V0





Conclusions

- I wrote a module that allows to compare the performances of 2 different Λ reconstruction methods at analysis level (starting with particle lists)
- problems in the interface between reconstruction and analysis with V0 objects
- pull distribution:
  - doesn't seem there is a big difference between dec and V0
  - π: sign of mean is opposite for π^+ and π^-

further considerations:

- quite low statistics: 110K events, ~2 Λ per event, BUT very low efficiency
- need to check if #Λ and #anti-Λ is so different already in the MCsample
- momentum distribution of π of the sample different from the momentum distribution of π coming from K_s (not directly comparable)