Incorporating the Kalman Filter into the MIPP Reconstruction

Jonathan M. Paley

Indiana University

February 16, 2006

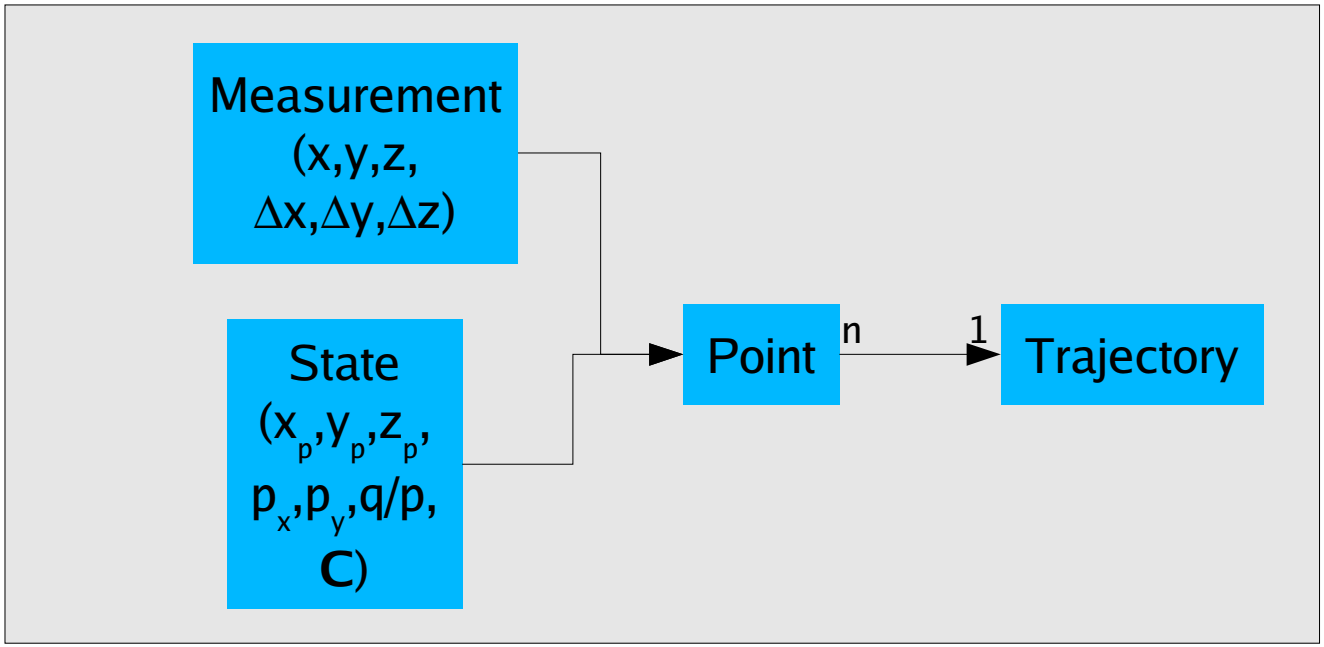
What Kalman Code?
Introduction to RecPack...

- Shortly after the last MIPP Software meeting at IU, Yagmur provided us a version of the RecPack (Kalman Filter) toolkit code used by the MICE experiment. The goal is to replace the current global fits with this.

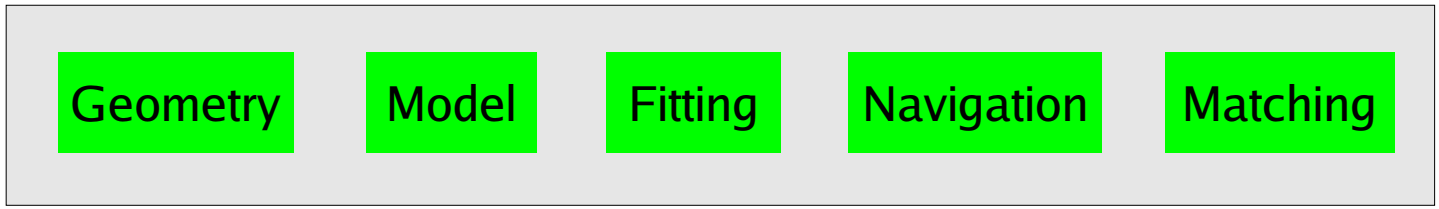
- Some notes:
  - Although RecPack has been used by several experiments (notably HARP & NOMAD), it is still under active development.
  - I have been in touch with the RecPack authors who have given us permission to use the code. Many many many thanks go to Malcolm Ellis (HARP) who has been very generous in providing me with some critical help in getting started...
  - RecPack does not use Root, therefore we cannot simply write “RecPack” points and tracks to the MIPP data stream.

RecPack Framework

“Data”



“Tools” - Manipulators of Data



Saving RecPack Fit Results in the MIPP Tree

- RecPack points and tracks cannot be *easily* written to the MIPP tree for a variety of reasons. Furthermore, these classes contain much more information than we really need.

- To get around this, we introduce new base classes and convert one type to another:
  - RBMeasurement--> base class with detector ID and dE information
  - RBViewHit:RBMeasurement --> eg, a 2D wire hit
  - RBSpacePoint:RBMeasurement--> eg, a TPC 3D hit (note, a SP can be made from 2-4 RBViewHits)
  - RBKPoint: holds a state vector & correlation matrix, and a pointer to a RBMeasurement (via TRef)
  - RBKTrack: holds a vector of RBKPoints

More About RBKTrack

- The interface to RBKTrack is nearly identical to RBTrack, so in analyses or in building the DST we should be able to simply replace “RBTrack” with “RBKTrack”.

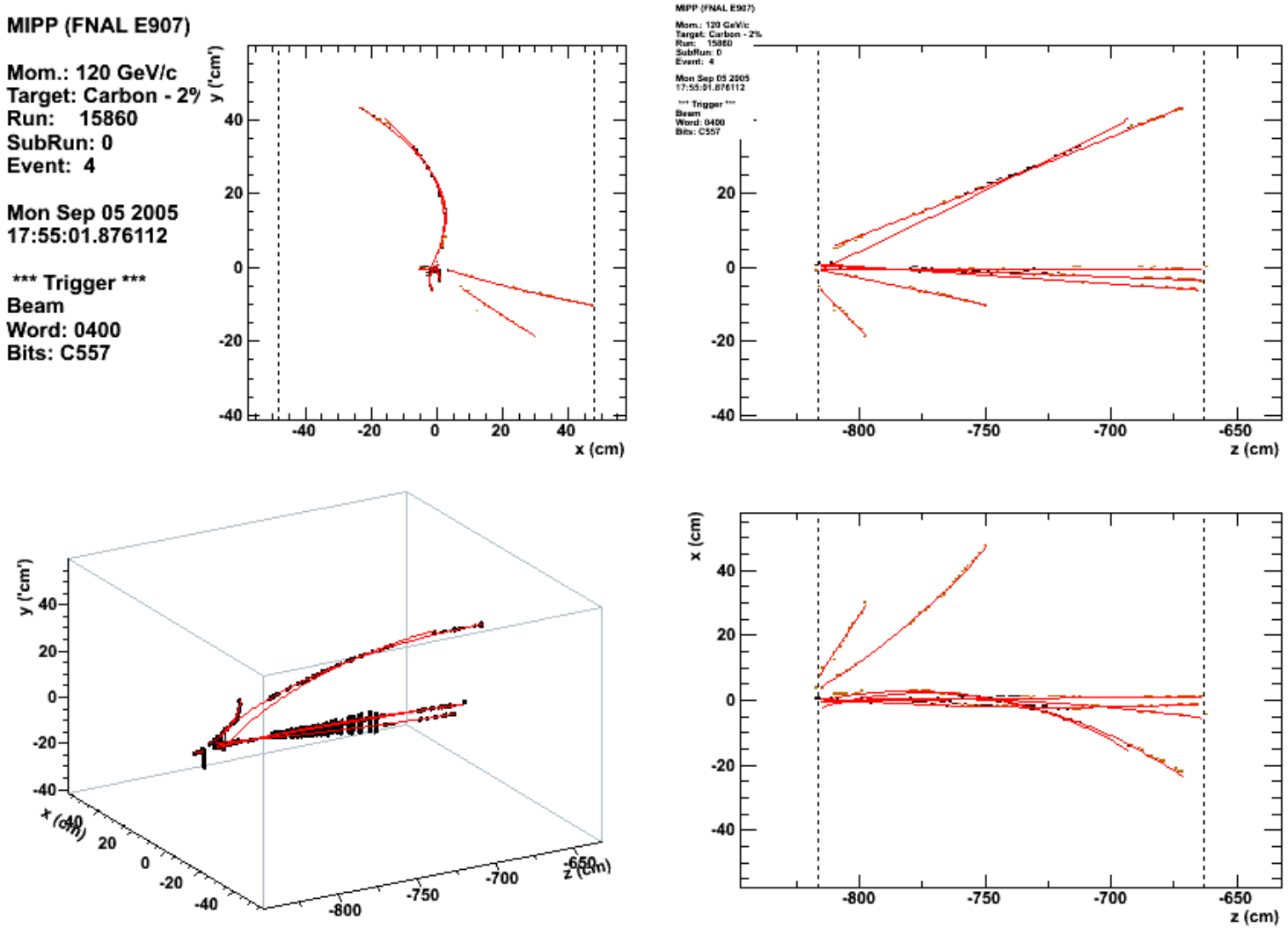
- As in the case of RBTrack, RBKTrack will hold more points than there are measurements for quicker and more accurate projections. However, since RBKTrack will hold pointers to actual measurements, we will be able to get these measurements directly from the track.

- Currently, points are only set to be sorted by z. We hope in the future to be able to fit by “s” in order to reconstruct tracks that turn around in the TPC.

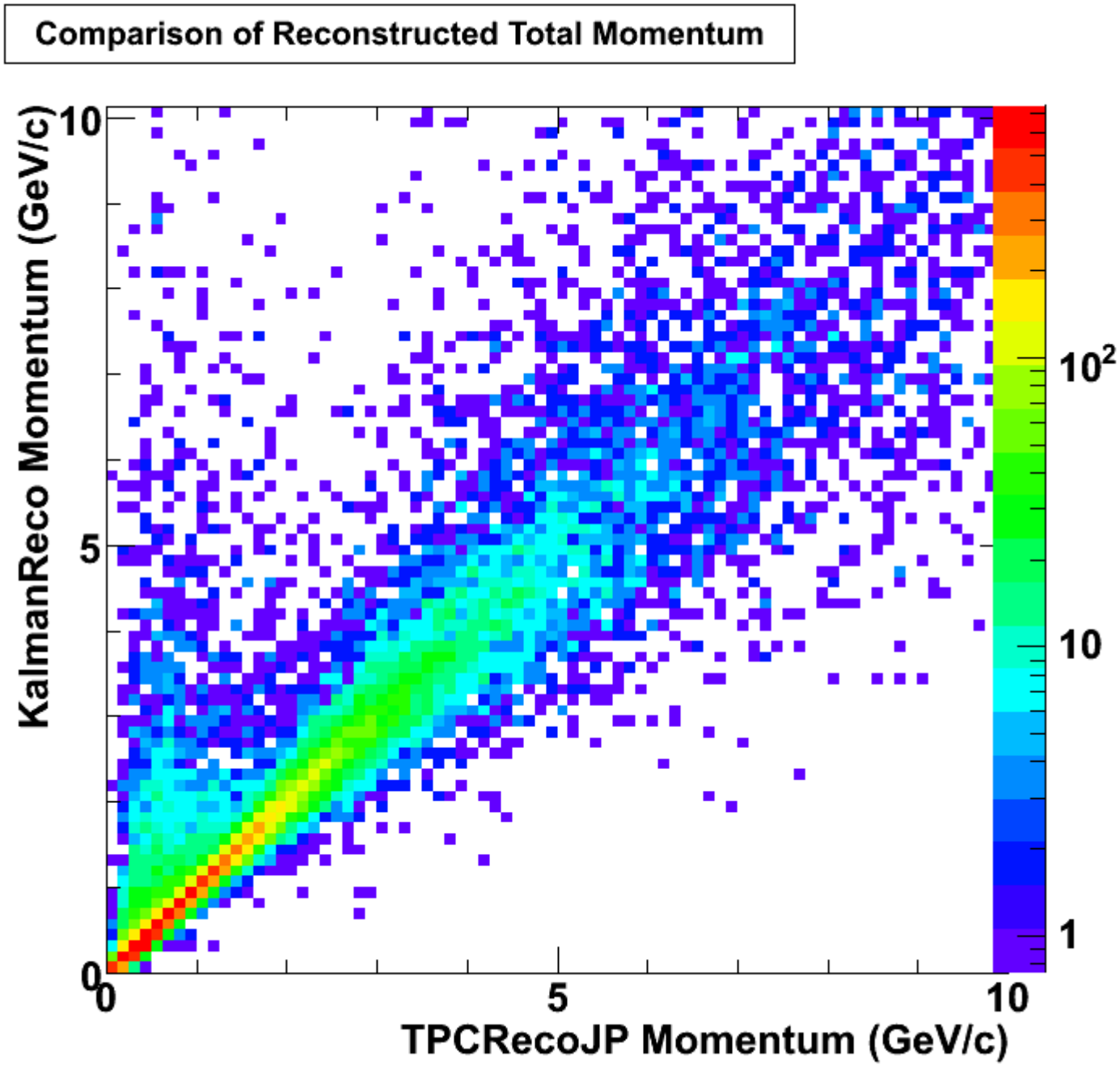
- The TPC event display now shows RBSpacePoints and RBKTracks.

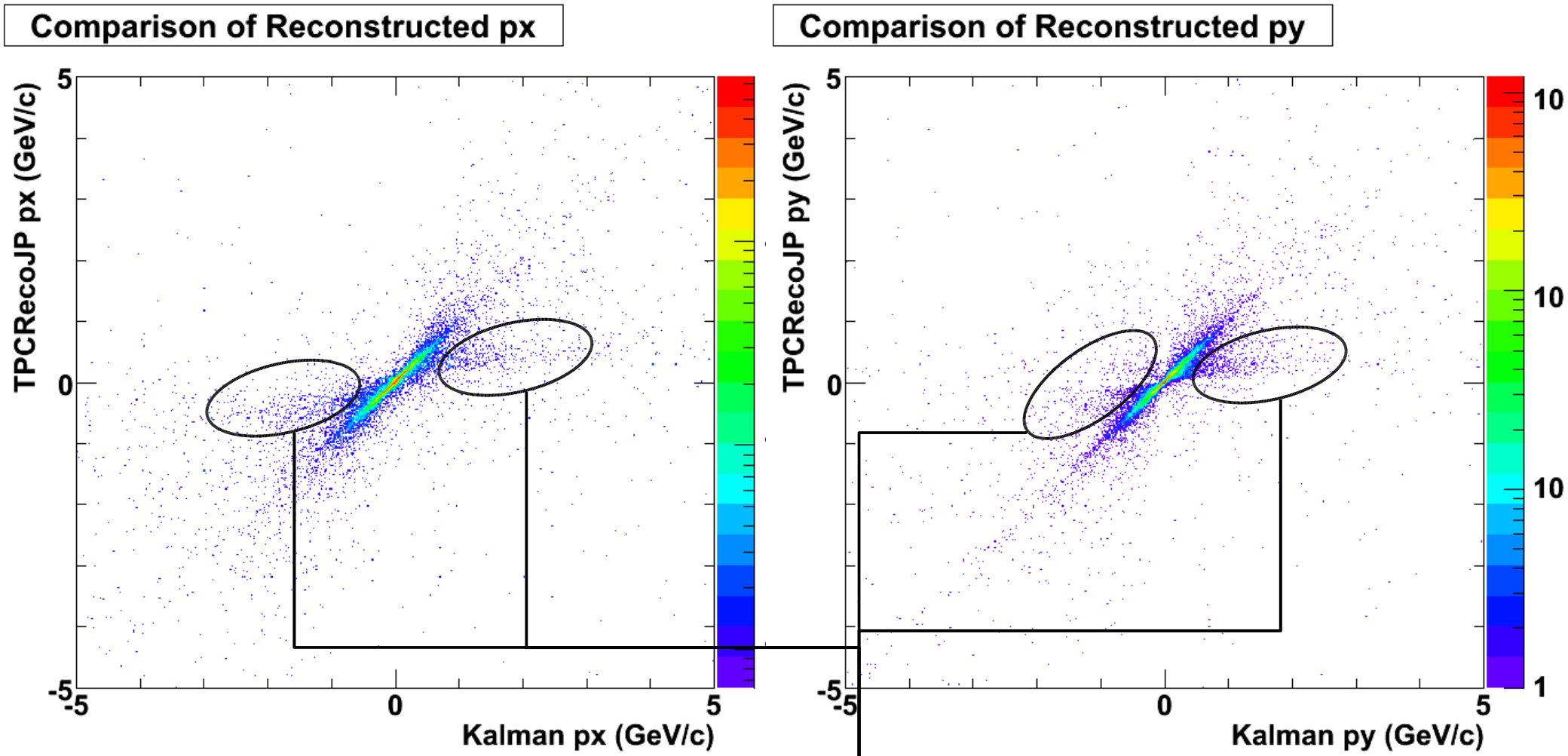
Status of Kalman Filter Reconstruction

- Short answer: it works! (kinda...)

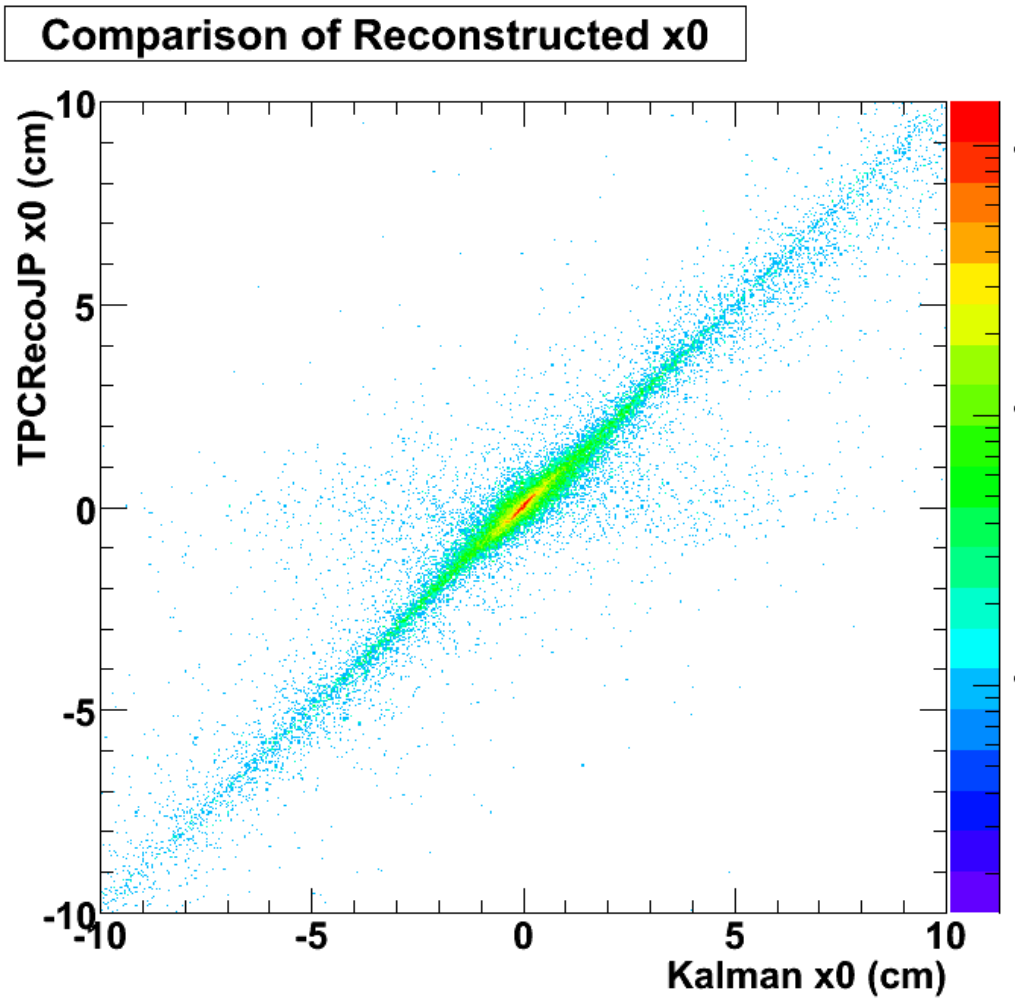


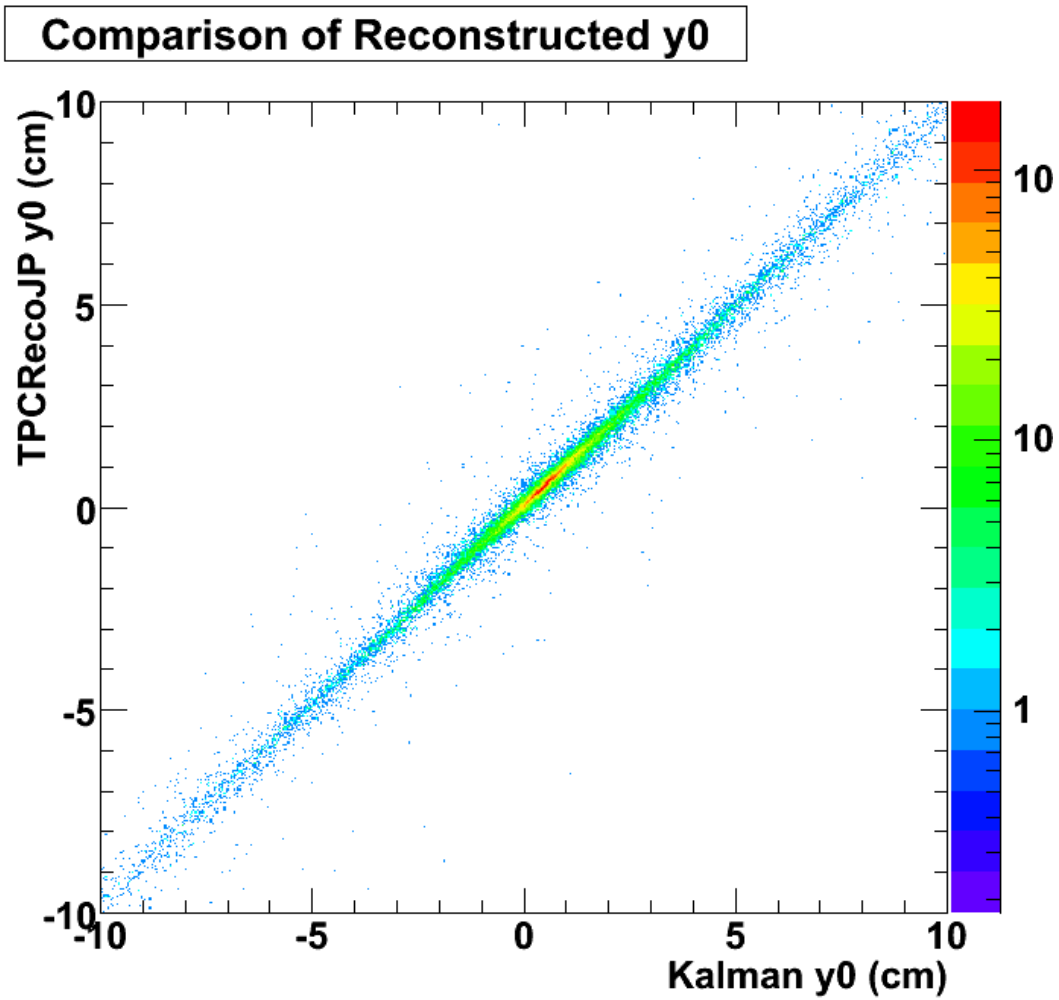
- Long answer: Using TPCReco tracks as seeds, I am able to refit TPC tracks using the Kalman Filter and assuming a constant B-field.


Comparison of Reconstructed Total Momentum
KalmanReco Momentum (GeV/c)
TPCRecoJP Momentum (GeV/c)
0
5
10
0
5
10
1
10
10^2

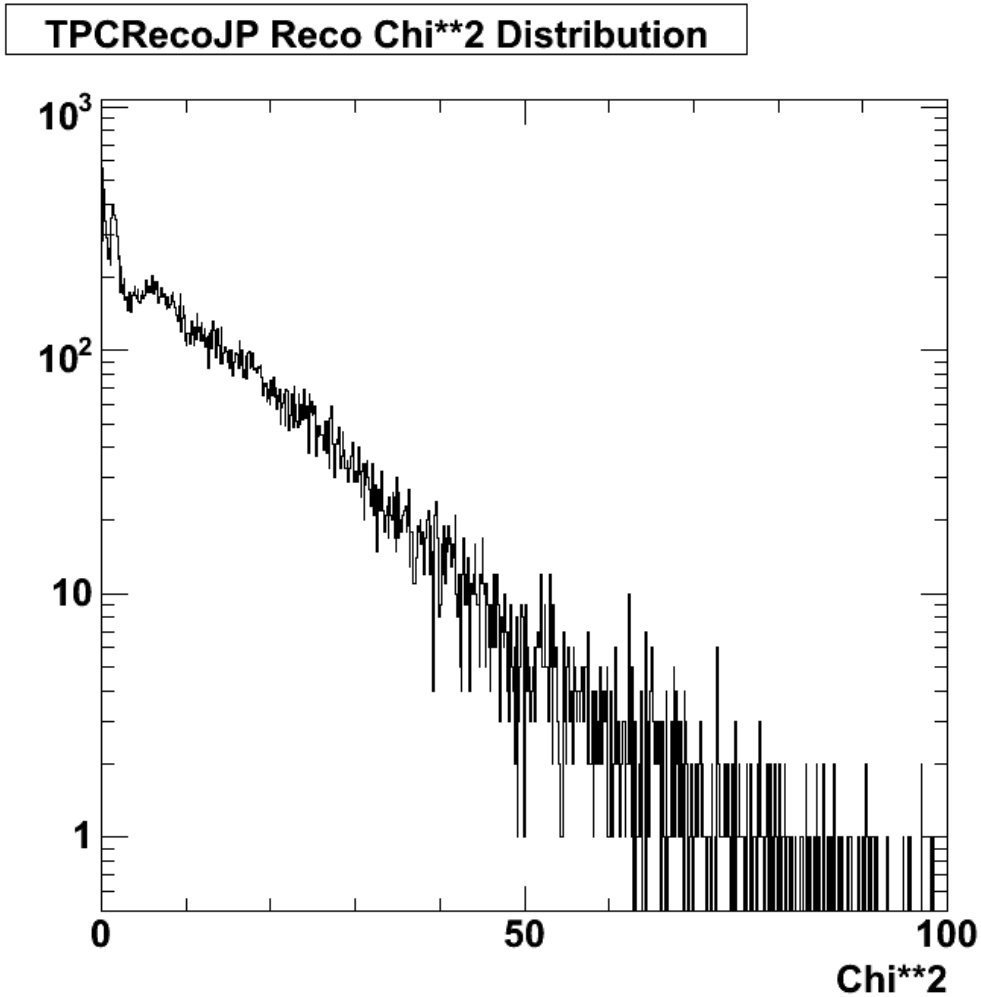


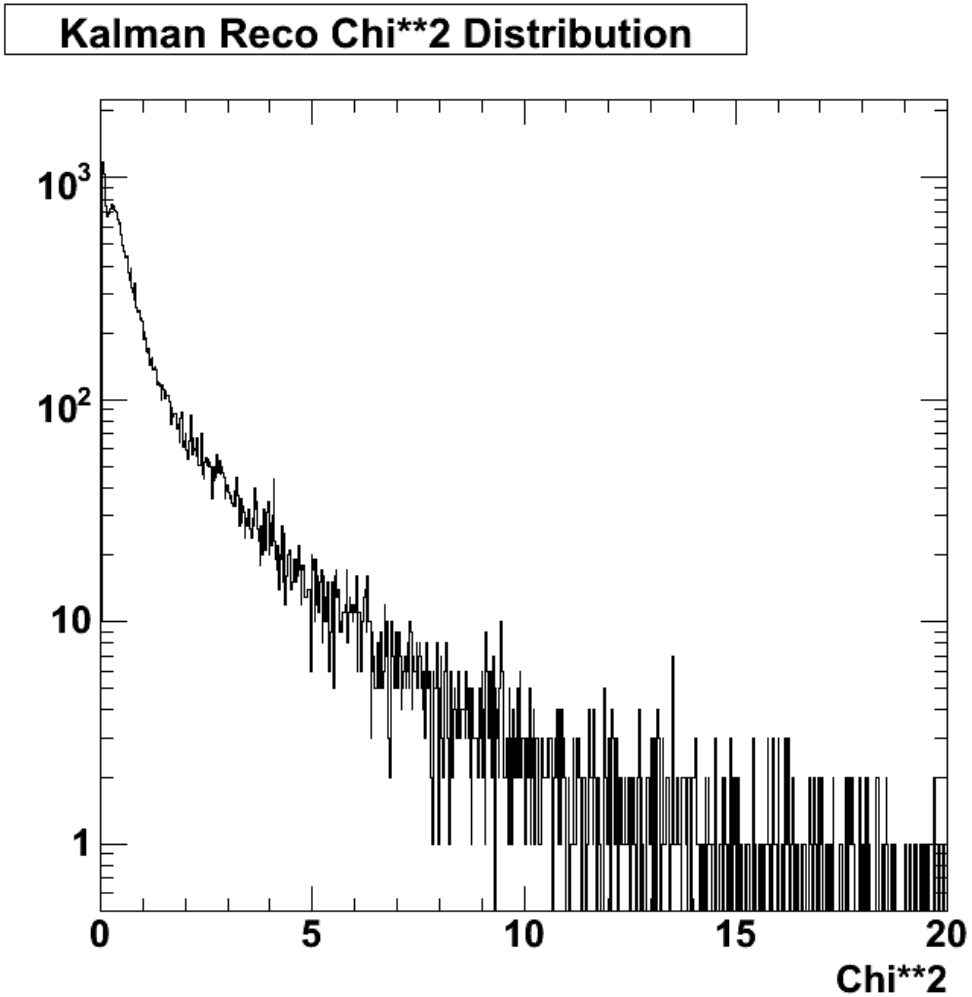
It's not clear which is right and which is wrong... only MC will tell!





We do better in predicting y, which is not surprising...





The KalmanReco χ^2 distributions looks “cleaner” to me, but still we will need to use MC to understand this better...

Some Performance Numbers...

- W/out creating RBKTracks from RecPack tracks, the re-fit of the TPC tracks is ~1.3x faster than TPCRTrackFind (which does both track fitting and finding).

- Creating RBKTracks from RecPack tracks makes the module run 2x *slower* than TPCRTrackFind. I think this can be improved once the detector reconstruction classes inherit from the new RecoBase classes (eg, TCRHit should inherit from RBSpacePoint).

Short-term “Roadmap” for Global Event Reconstruction

- Continue to use TPCRecoJP tracks as seeds for the Kalman fits.

- Get the vertex fitting to work with just the TPC tracks (in-progress... I'm close, I can almost taste it).

- Incorporate the B-field map into the Kalman helix fit (there may be delays due to an impending new release of the RecPack code).

- Incorporate the beam particle into the vertex fit.

- Incorporate the DC and MWPC wire hits into the track fits.

- Create interface between MIPP Root geometry classes and the RecPack geometry classes.