Tracker MC

Christopher Heidt
University of California Riverside

MICE CM 38
Feb 23 2014

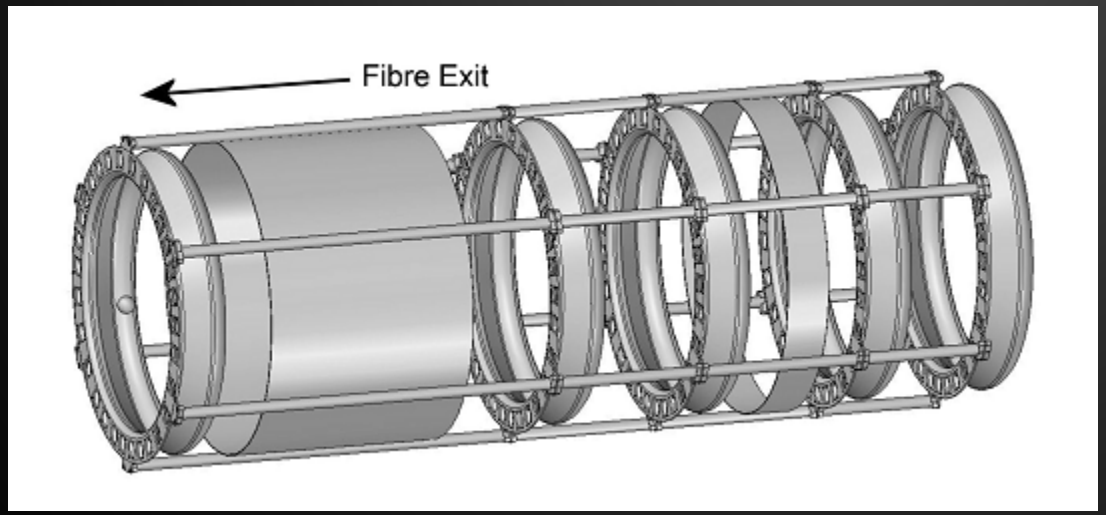
Outline

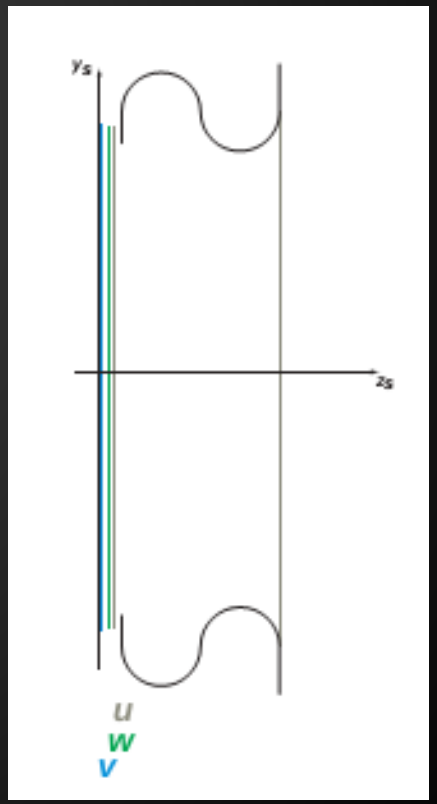
- Background
  - How do we model the tracker
  - How do we model hits
  - The digitization process
  - Results
- New Additions
  - Bridging MC to recon
  - Modeling VLPC noise
- New Idea
  - Individual channel configurations

Background

How do we model the tracker?

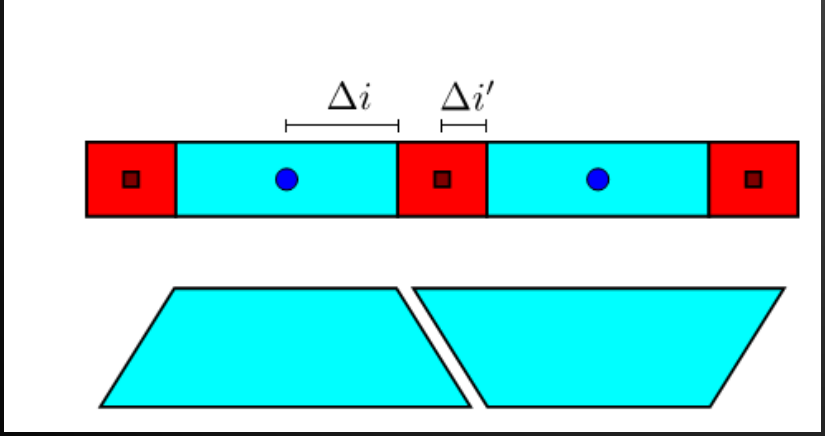
- Stations/Trackers
  - Station models are defined as an cylinder containing three empty planes, which will later be filled, separated by thin mylar layers
  - Looking into adding carbon fiber station bodies
  - No current plans to add scintillating fiber tails, connectors, light guides, or the rest of the station body

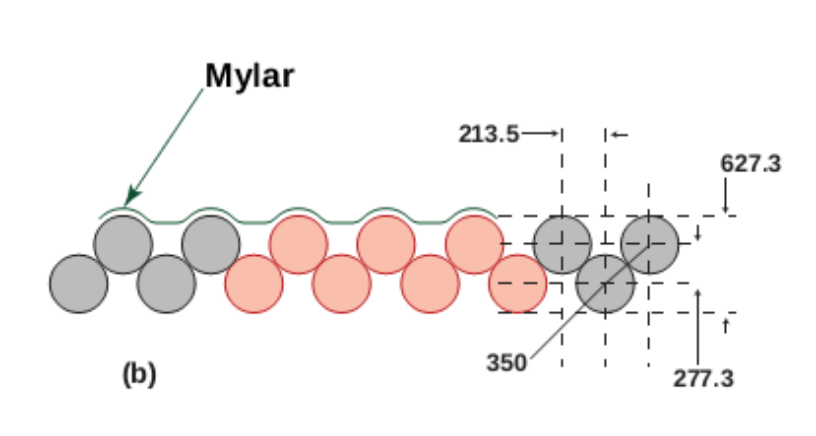




How do we model the tracker

- Planes
  - Initialized as thin cylinders with an active radius of 16cm
  - Fibers place one by one along diameter of plane
    - length of fiber based on placement of fiber
    - depth of fiber alternated between -0.5r and 0.5r
  - Contrasted to earlier G4Mice model which modeled fibers and overlapping/non-overlapping rectangles
- Both stations and planes can be rotated along x,y,z axis





How do we model hits

- Hit Generation
  - Beam (PID, energy, number of particles) generated outside of tracker MC
  - When particle interacts with scintillating fiber a certain amount of energy is deposited
    - Value determined by Geant4 libraries
    - Energy value is recorded along with position, momentum, and id (tracker,station,plane,fiber)

The digitization process

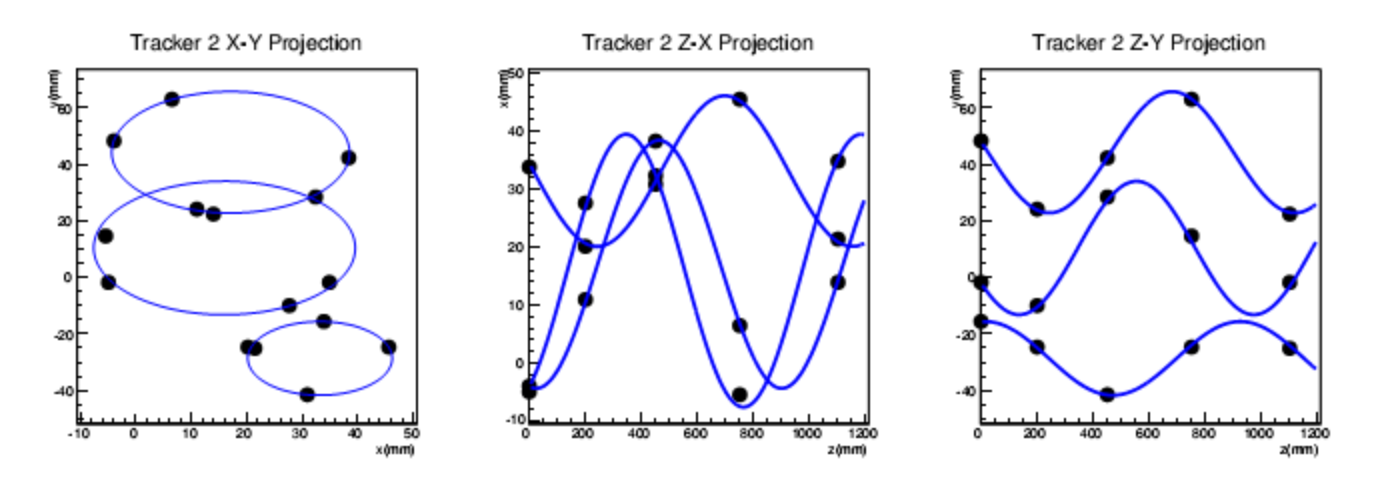
- The number of PE in a hit is determined by energy deposited
  - Energy is multiplied by a figure based upon physical factors of the fibers
  - Channel number is determined using fiber number
  - PEs from hits in the same tracker, station, plane, and channel in an event are summed

The digitization process

- ADC resolution simulated
  - Raw NPE checked against ADC saturation
  - Gaussian smearing based upon ADC resolution applied
- Results stored as digits to be passed to recon
  - MC digits should be identical to that produced from data

Results

- Used in pattern recognition and Kalman development and testing
  - Used to test code development
  - With field and without
- Will provide even more use with Trigger and RFCC MC modules



New Additions

Modeling noise from VLPC

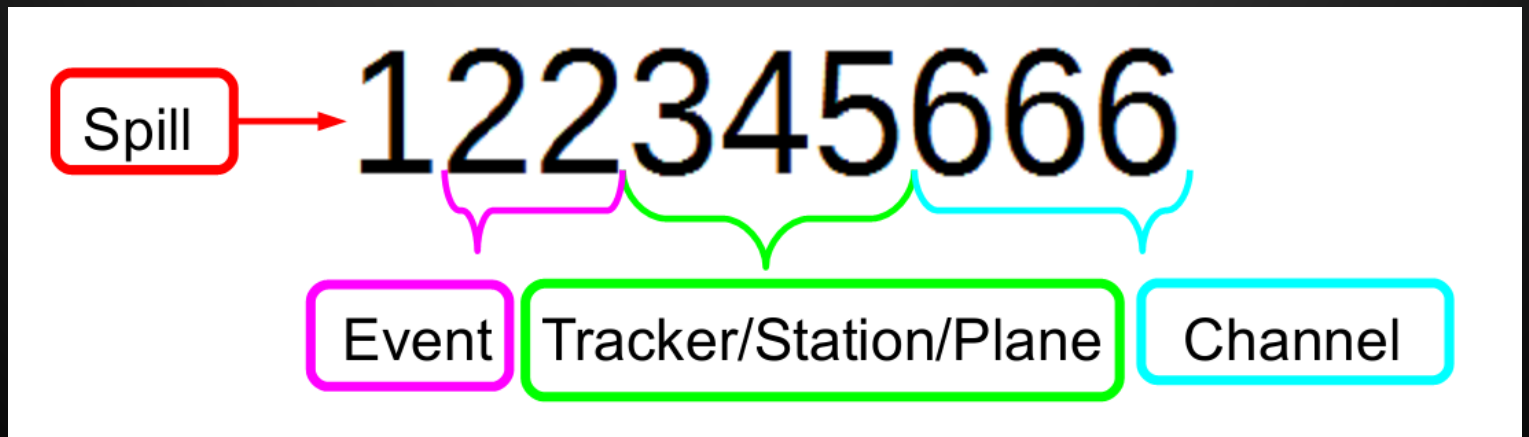
- Dark count present in VLPC
  - Originates from thermally excited electrons causing a cascade event
  - Present in ~1%/trigger/channel usually on the order of 1 PE
  - Described by Poisson distribution
- Modeled in the MC
  - At the start of each event channels are check for presence of noise
  - Random value fed to Poisson function to determine integer magnitude of noise
  - If present noise is saved in a format identical to digits but in its own branch

Modeling noise from VLPC

- Results combined with digits
  - MC digits compared to noise to find any possible overlap
  - PE of overlapping events are summed
  - Events that do not overlap are simply added to digit branch
  - Final results are sent for ADC simulation for gaussian smearing
  - End results produce digits that can not distinguish noise from MC
- Same framework can be used to add in crosstalk noise from ADC/TDC

Bridging MC to Recon

- Unique id number created that correlates hits with digits they create
  - Allows for us to follow chain hits -> digits -> clusters -> space points -> track points
  - Tool set built to run outside of MC, results stored in memory not to JSON
- Will be used in testing track finding algorithms
  - Work being done now in writing scripts to use this



New Idea

Individual Channel Configurations

- Individual calibrations
  - Write calibrations to separate file
  - Read in at start of run
  - Referenced at digitization level
- Allows:
  - Modeling dead fibers, different light yields
  - Channel noise, settings
  - ADC resolutions
- Does not Change:
  - Fiber definitions

Conclusion

- Current model gives us an accurate but general view of tracker
- Framework for noise in tracker electronics in place and VLPC noise in place
- MC is being used to aid in code development
- With more data we can be more realistic