Tracker Noise and Hits to Digit Lookup in MAUS MC

- The TL;DR
  - Added a MAUS mapper to produce simple noise simulation described by Poisson distribution
  - Integrated noise into MAUS MC digitization functions so that it shows up in MC recon
  - Developed a process to generate ID that are stored in MAUS data structure to form a bridge between MC and Recon
  - Tools are in place to begin studying PR efficiencies, some further development would be beneficial

Noise Hits

- Noise is added to MAUS as a new mapper
  - Add MapCppTrackerMCNoise before MapCppTrackerMCDigitization
  - MC Digitization has been altered to look for presence of MC noise before running digitization and if present, add in new functionality

```
# This input generates empty spills, to be filled by the beam maker later on
my_input = MAUS.InputPySpillGenerator()

# Create an empty array of mappers, then populate it
# with the functionality you want to use.
my_map = MAUS.MapPyGroup()
my_map.append(MAUS.MapPyBeamMaker()) # beam construction
my_map.append(MAUS.MapCppSimulation())  #  geant4 simulation
my_map.append(MAUS.MapCppTrackerMCNoise())
my_map.append(MAUS.MapCppTrackerMCDigitization())  # SciFi electronics model
# my_map.append(MAUS.MapCppTrackerRecon())  # SciFi recon
# my_map.append(MAUS.MapCppTrackerReconTest())  # SciFi recon
# can specify datacards here or by using appropriate command line calls
datacards = io.StringIO(u"")

# reducer = MAUS.ReduceCppPatternRecognition()
reducer = MAUS.ReducePyDoNothing()
```

Noise Hits

- Function searches each tracker channel for noise.
  - Possibility of noise in each channel determined by global variable (Poisson Mean)
    - Possible to change in datacard
    - Best option would be to allow a per channel setting, but this may be a lot of work for little gain
  - Poisson Mean determined by probability of sinlge NPE dark count event (1.7% or is this the integrated dark count probability?)
    - Poisson mean feed into CLHEP library to determine number of hits
    - Noise is currently stored as integer number of NPE (should this have some amount of Gaussian smearing?)
  - Time may be an issue with this method
    - Something near 50 -60% increase in run time
    - If we are running the MC often it might be worth the trouble to rewrite how we handle the distribution

Noise Hits

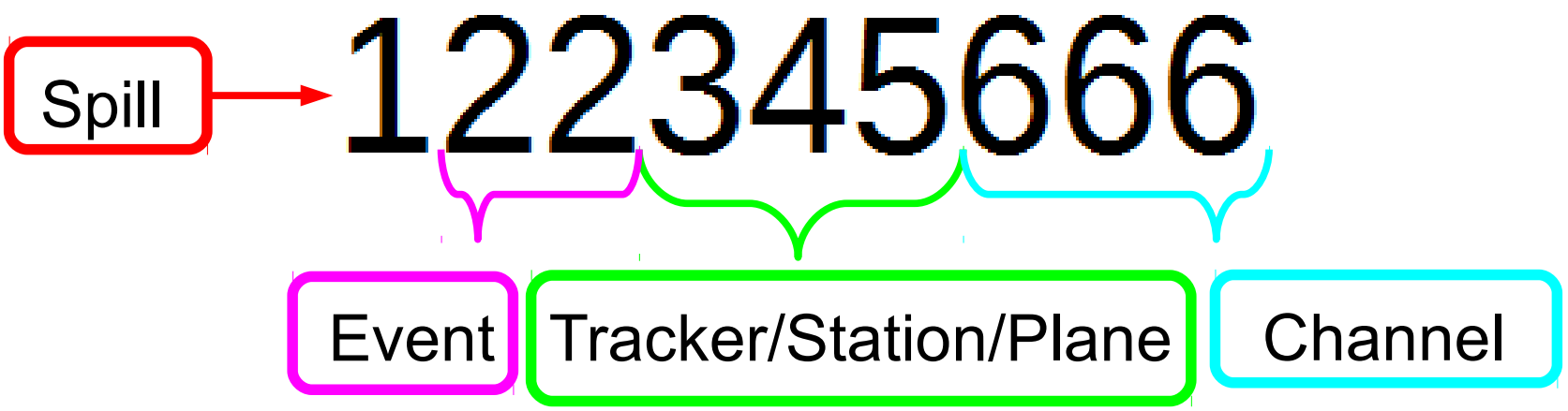
- Noise hits are digitized in one of three ways:
  - If directly on a MC hit the noise hit is digitized with the MC hits
  - If the noise hit is within one channel of a MC hit the noise hit is digitized like it was a hit, regardless of NPE
    - This way it can be used for cluster finding
  - If the noise hit is 2 NPE or over it is digitized as a hit, since our digitization cut off is 2 NPE
    - Cutoff is technically in clusters, but this helps cut down on clutter somewhat since already account for clusters

MC Hits to Digit Lookup

- Now a feature of MapCppTrackerMCDigitization
  - If you use the MAUS MC to create tracker digits this will now automatically work
- The MC Hits to Digit writes in three separate branches of MAUS data structure
  - MC Hits
  - Noise Hits (optionally)
  - SciFiMCLookup (new)
- **Never** touches the recon side of the data structure

MC Hits to Digit Lookup

- How it works
  - When MC hits are collected to form digit or noise hits meet the criteria for digitization
    - Digit ID is created and written into MC hits and noise hits data structure
    - A new entry is created in SciFiMCLookup data structure and the Digit ID is entered
  - Digit ID is created from the following information:



MC Hits to Digit Lookup

- TODO: not necessary, but may be helpful
  - Store the digit in the Lookup branch
    - Make it easier to do a comparison between raw digits and those found in seed points
  - Store information about noise in Lookup
    - Was any noise used in the creation of this digit? This is a good place to store this piece of information
  - *Write tests!*
    - This needs to be done, along with some code clean up and some more comments