

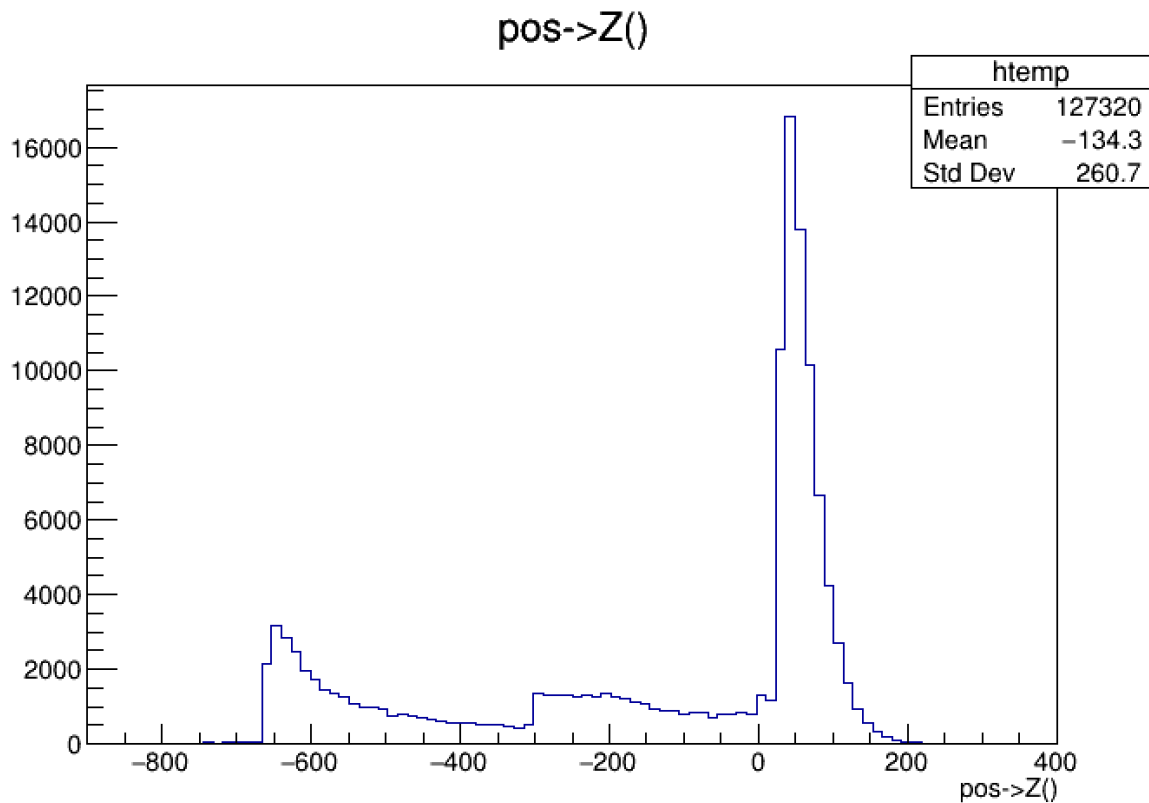
Monte Carlo Generation

The Monte Carlo is based on Geant4-produced mu+ and mu- tracks that are coming from vertex locations from around the target and dump. These Geant-generated single events are used to fill a hit matrix event (or embedded messy event) making a new event with some amount of reconstructable tracks (signal) and some amount of non-reconstructable partial tracks (noise). These hit matrix events are used as training data and so must contain a wide variety of noise. Some amount of the training data must be a variety of signal tracks with no noise at all while some have a little noise and some have more and more noise. The quality of data is very important and it is critical to try to incorporate as much information from experimental data as possible including occupancy patterns and density as well as random uncoupled hit frequency as a function of high voltage. The timing of hits must also be simulated in a way that the timing window cut can be tested. There are some amount of hits that are out of time with something that could have been a triggered muon from the target. The same basic timing cuts used in the Ktracker Fun4All analysis should also work for our events. We must also pay attention to dead channels and have the MC reflect that or have a way in the reconstruction package to turn off or black out channels.

After the filter is working well and it's possible to separate the no fully reconstructable track events from single and dimuon events then it's important to get the track finder and reconstruction part working very well. It is probably best to work on this part with clean (no noise) or cleaner (low noise) events until you see good accuracy and precision on the momentum and vertex information.

Use a mixture of DY, J/psi, and single muon tracks for background tracks.

The generated muons should be weighted based on the following Z distribution, which shows the approximate distribution of muons that will be generated in the events.



X-Y position information, and a wider variety of track angles

Make sure tracks are coming both from the target and elsewhere.

Don't use any timing information in the filter or track-finding part of the algorithm.

There should also be partial tracks that traverse part-way through the detector array. These tracks can go through anywhere between 1 and 5 planes of a station.

These should be generated to roughly match the occupancy that we see in [Seaquest docdb 9631-v3 in slides 26 and 27 \(DC occupancy\)](#). The plots show that the occupancy is lower at higher Z (station 1 has higher occupancy than station 3). This means that we should weigh the partial tracks so that there are more going through the first two stations. We are trying to match the occupancy after out-of-time cuts.

Once the full tracks and partial tracks are added, we can add sources of noise: Electronic Noise, Delta Rays, and Cell-Edge Hits. These are not as important for training, because they can mostly be removed by if statements.

We will also use E906 data to compare the generated MC to real experimental data. If the neural networks we train are able to perform equally well on E906 data and MC data, then we know we have a good agreement.

Make sure your generator runs fast. Use GPU if needed to accelerate. List here where the final generator is on Shannon.

Make sure that Drift information is in the data.

An Outline of what the Event-based MC should entail:

All the possible regions of the vertex should be generated, like more than 10 cm X and Y with uniform distribution and Z with the above distribution. The regions before the -300 are from the collimator and upstream components. Pay attention only to the blue line (after in-time cuts) from doc 9631.

The number of planes and stations should be randomized so that a partial track traverses resulting in 1, 2, and 3 plans of a station. Partial tracks should also be made to go through one full station but only 1, or 2 planes of the next station. There are more partial tracks in station 1, then fewer in station 2, then much fewer in station 3.

a.) Produce several spills (~100) each around 50K with 10% dimuon, 70% single muon, and 20% with no complete muon tracks. Make these as compressed NumPy arrays and RUS files.

b.) Run a demonstration showing the top performance of the filter

c.) With the same set of spills, run a demonstration showing the top performance of a single/di muon filter

d.) a.) and b.) should be done with 906 data and full MC events with kTracker/Q-tracking tests. What is the difference in accuracy? Are there differences in the patterns?

c.) Can a trigger emulator be used to make the MC more like the real trigger sets (trigger efficiency map)?