Track Selection in HCANA

M. K. Jones

May 23, 2020

Abstract

Explanation of tracking selection in HCANA. The tracking in HCANA is only partially described. For the purposes this document, I will just refer to the SHMS variable and parameter names, but the same applies for the HMS.

1 Introduction

The goal in HCANA is to find the track associated with the trigger or the golden track. HCANA is not written to find multiple tracks. To find multiple tracks, the hodoscope code would have to be rewritten so that one could find multiple times to associate with each track. Right now the hodoscope code focuses on finding the Start Time (also call the Focal Plane Time) of the track that caused the trigger.

The drift chamber code finds clusters of hits in each chamber (called space points). Each space point in a chamber is fit with a line to determine the dx/dz, x and y at the focal plane (the dy/dz is set to zero in the fit). The code loops through all the space points and matches the space points in the two chambers. For each space point pair, a possible track is found. So if chamber 1 has one space point and chamber 2 has two space points then two possible tracks will be found.

Selection criterion exists for eliminating matches of the space points between chambers. For example, one can eliminate matches between space points based on the difference between the dx/dz angle, x and y position of the space points. But presently most experiments use parameters which are very large and almost all possible matches between space points are accepted as a possible track.

The parameters for the selection of matches for possible tracks are located in *PARAM/SHMS/GEN/ptracking.param.* The main cuts that reject tracks are based on the number of hits in each of the chambers. The parameter called *p_max_pr_hits* is an two-dimension array with an element for each chamber. *p_max_pr_hits* is typical 25 or above. If either chamber has more than *p_max_pr_hits* (a DC hit has to be within the DC raw time min and max parameters) then no track will be found for that event. Another parameter is *pmin_hit* which is an two-dimension array with an element for each chamber.

pmin_hit is the minimum number of hits in a space point for a chamber. Typically it is set to either 4 or 5. With *pmin_hit = 4,4* , then the number of space points found per chamber will be higher than if *pmin_hit = 5,5* especially as the rates on the chambers increases. of course, the tracking efficiency becomes more sensitive to missing wires when the *pmin_hit* is larger.

Each track is a THaTrack object and information is stored and retrieved using the methods of the THaTrack class. In the CoarseProcess method of THcHodoscope class, all possible tracks are looped over and checked for matches between the hodoscope PMT hits and each track. Each track is projected to the hodoscope plane and the code loops through each hodoscope hit in that plane. If difference between the paddle center and the track position is less than half the paddle width plus *phodo_slop* then paddle is a good hit. The parameter *phodo_slop* is an array that is usually set in the *PARAM/SHMS/phodo_cuts.param* file. Typically *phodo_slop* $= 2, 2, 4, 4$ which is units of cm and one element for each plane 1X,1Y,2X and 2Y. The paddle width is set in the *PARAM/SHMS/phodo_geom.param* file. More than one paddle can be a good hit in a plane. If the 2X (2Y) plane has a good paddle hit then a flag is set true using the SetGoodPlane3 (SetGoodPlane4) THaTrack method. The total number of good PMTs hit by the track is stored using the SetNPMT method of THaTrack. If three or more planes have a good hodoscope hits, then the time-of-flight corrected times are averaged to give a focal plane time. If the condition is not met, then the focal plane is set to the hodoscope start time. The focal plane time is stored for the track with the SetFPTime method of THaTrack.

If there is a good hodoscope hit in planes 1X or 1Y and a good hodoscope hit in planes 2X or 2Y then beta is calculated by fitting a straight line to the hodoscope hits times as a function of their Z position. If this condition is not met, then beta is set to 0 and the chi-squared for the fit is set to -1. This is the tree variable *P.hod.beta* and the chi-squared is not saved in the tree. If the linear fit fails because the denominator is negative, then beta is set to 0 and the chi-squared for the fit is set to -2. The hodoscope beta and chi-squared are stored with the SetBeta and SetBetaChi2 methods of the THaTrack class.

Given a number of possible tracks are find in the space point matching, the code needs to determine which track is the one associated with the track that caused the trigger. The code needs to identify the "golden" track. This is done in the FindVertices method of the THcHallCSpectrometer class. A track is never rejected at this point in the code. Therefore the track selection does not effect the track efficiency. Hopefully, the track selection picks the track that caused the trigger and the associated reconstructed target quantities are for the event. Three methods are possible for the track selection.

2 Three types of Track Selection

The method of track selection is determined by parameters *psel_using_scin* and *psel_using_prune* in *PARAM/SHMS/GEN/ptracking.param*.

If *psel_using_scin=0* and *psel_using_prune=0*, then the FindVertices method calls the BestTrackSimple method. This method selects the best track as the one with the lowest chi-square per degree of freedom. This is simple, but probably not the best way.

If *psel_using_scin=1* and *psel_using_prune=0*, then a track selection method developed for the Fpi experiment during the 6 GeV era will be used.

If *psel_using_prune=1*, then a track selection method developed by Peter Bosted during the 6 GeV era will be used. It is an expansion on the Fpi method. It uses more criteria in the track selection. This is the pruning method.

3 Track selecton by the pruning method

In the pruning method, the code steps through the track selection criteria to eliminate tracks based on the successive track selection criteria. An logical vector, *keep*, is created which is the size of the total number of possible tracks. Initially, all elements of *keep* are set to kTRUE. For each criteria, the code loops through all tracks and determines number of tracks, NGOOD, which pass the current track selection and all the previous track selection criteria. If NGOOD >0, then loops sets *keep* to kFALSE for the track that does not passes the current track selection. If NGOOD $= 0$, then the present track selection is not used. If number of good tracks has not be narrowed down to one by all the previous criteria, the last track selection criteria is which track has the lowest chi-squared per degree of freedom which will select one track.

Below is a list of the track selection criteria in the order that they are applied in the code. The main idea is to give priority to selection based on track reconstruction. The next level of track selection is based on matching between the track and the hodoscope. The final track selection is based in the best chi-squared per degree of freedom. Most criteria are a cut on a track variable and have a parameter that can be set in *PARAM/SHMS/GEN/ptracking.param.* If the parameters are not set , then the default value is given in the description. The defaults are not set to be the best. Experiments need to determine the parameters to use.

Xptar Keep the track if abs(reconstructed dx/dz) < *pprune_xp*. *pprune_xp* is a parameter. If it is not set the default value is 0.2 rad.

Yptar Keep the track if abs(reconstructed dy/dz) < *pprune_yp*. *pprune_yp* is a parameter. If it is not set then the default value is 0.2 rad.

Ytar Keep the track if abs(reconstructed y) < *pprune_y*. *pprune_y* is a parameter. If it is not then set the default value is 20 cm.

Delta Keep the track if abs(reconstructed δ) < *pprune_delta*. *pprune_delta* is a parameter. If it is not set then the default value is 30%.

DipoleExit Keep the track if the track tracks back to be inside the dipole exit window. *pprune_DipoleExit* is a parameter. If it is not set then the default value is 0 (kFALSE) and the test is not used.

Beta Keep the track if abs(beta_hod - beta_p) < *pprune_beta*. *pprune_beta* is a parameter. If it is not set then the default value is 30. beta_hod is P.hod.beta that is associated with the track in the CoarseProcess method of THcHodoscope class. In CoarseProcess, P.hod.beta is set to zero for a possible track, if the beta can not be calculated. beta_p is beta calculated from the measured particle momentum and particle mass set by *ppartmass* in the kinematics file.

Track Degree of Freedom Keep the track if the track's degree of freedom is >= *pprune_df*. *pprune_df* is a parameter. If it is not set then the default value is 1 which in effect means that it is not applied.

Number of PMTs Keep the track if the track's number of PMTs hit is >= *pprune_pmt*. *pprune_pmt* is a parameter. If it is not set then the default value is 6.

beta chi-square Keep the track if beta_chisq is < *pprune_chibeta* and > 0.01. *pprune_chibeta* is a parameter. If it is not set then the default value is 100.

focal plane time Keep the track if the abs(fptime-starttime) is < *pprune_fptime* and > 0.01. *pprune_fptime* is a parameter. If it is not set then the default value is 1000.

Hodoscope Y2 plane test Keep the track if the track hit the hodoscope Y2 plane. The criteria is always used.

Hodoscope X2 plane test Keep the track if the track hit the hodoscope X2 plane. The criteria is always used.

Track chi-square Keep the track with the lowest chi-squared per degree of freedom. The criteria is always used. If more than one track with keep=kTRUE, it will select one track.

4 Monitoring the track selection

If the pruning method is used for track selection, then one can look at the tree variable *P.tr.PruneSelect* to see which track criteria was used to winnow the tracks down to one golden track. *P.tr.PruneSelect* is a integer which is from 0-13. *P.tr.PruneSelect* = 0 , if there was only one possible track. *P.tr.PruneSelect* =13 if chi-squared is used to determine the golden track. *P.tr.PruneSelect* = 1-12 depending on which was the last selection criteria in the ordered listed in the previous section.

The other variables in *P.tr* are arrays with the size of the number of possible tracks found for that event. In principle, one can use this array to help decide (or check) the parameters used in the track selection. To know which track in the array was picked as the golden track check the tree variable *P.gtr.index*. *P.gtr.index* will give the element number in the *P.tr* array that is the golden track. The size of the array for that event is stored in Ndata.P.tr.beta, for example. Some variables are for Hall A analysis and are not listed. The some variables used by Hall C are:

P.tr.beta beta calculated using the hodoscope times and track information in the TOF correction.

P.tr.betachisq chi-square of the beta calculated using the hodoscope times and track information in the TOF correction.

P.tr.chisq Chi-squared per degree of freedom for the track.

P.tr.x x (cm) at focal plane

P.tr.y y (cm) at focal plane

P.tr.th dx/dz (radians) at focal plane

P.tr.ph dy/dz (radians) at focal plane

P.tr.tg_x x (cm) at target

P.tr.tg_y y (cm) at target

P.tr.tg_th dx/dz (radians) at target

P.tr.tg_ph dy/dz (radians) at target

Need to add variables to *P.tr*, so that each track selection criteria can be checked.