An Objective Analysis Package For High Energy Physics

Forest Rouse [1]
Department of Physics, University California, Davis

This paper describes the status of the search for the Λ_c in Deep Inelastic 1994 data sample at H1 using an portable objective analysis package. This paper describes the package and some of the lessons learned in how objective analysis can be used to analyze high energy physics data.

Key words: Analysis; Objects; HERA; H1;

1 Introduction

Overall, the production of charm at HERA is an important lens on the fragmentation mechanism in high Q^2, low x interactions. Photon–gluon fusion is by far the predominate mechanism of charm production at HERA energies [1]. Λ_c production can be used to augment our understanding of this process.

The use of objects in the analysis derives from the regularity and similarities of what is required for resonance analysis[2]. The tracks, the abstract description of a physics final state, particle decay modes and properties, the propagation of errors, and cuts are all well described by objects.

The program kinematic fits final states much like SQUAW[3]. The novelty of the software comes from it's ability to *recursively* search for final states. Also, unlike SQUAW, any one track can be used in any number of final state candidates. The software also can be ported to different experiments. A user needs about 20 lines of FORTRAN subroutines to use the system. The system is itself written in C++.

2 Analysis

We searched for Λ_c decays into 3 separate decay modes: $\Lambda_c \to \mathrm{pK}^0$, $\Lambda_c \to \Lambda\pi$, and $\Lambda_c \to \Lambda\pi\pi\pi$. Standard H1 tracking cuts are applied[4]. All Λ_c candidates falling within a mass window of $\pm 300\,\mathrm{MeV}$ are kept. Additional cuts are placed on the energy fraction (z) carried by the Λ_c, and the helicity angle of the decay products from the decays of the Λ_c and separately from the V0 (K^0 or Λ^0). The helicity angle distribution reduces the combinatoric background from jets.

All surviving Λ_c candidates are then written to an n-tuple for final off-line analysis. The above cuts reduced the about two million 1994 Deep Inelastic Scattering events into about five thousand six hundred Λ_c candidates.

The results of this analysis are not final. However, the success of the strategy can be judged by figure 1. This plot shows the invariant mass plot of the Λ_c candidates after additional offline analysis and shows an enhancement in the Λ_c mass region for the decay mode $\Lambda_c \to \Lambda\pi$. Hence, the front end objective analysis successfully gave a reduced data sample that could be used for the decay modes in question.

[1] Work supported under Department of Energy grant DE–FG–91ER40674.

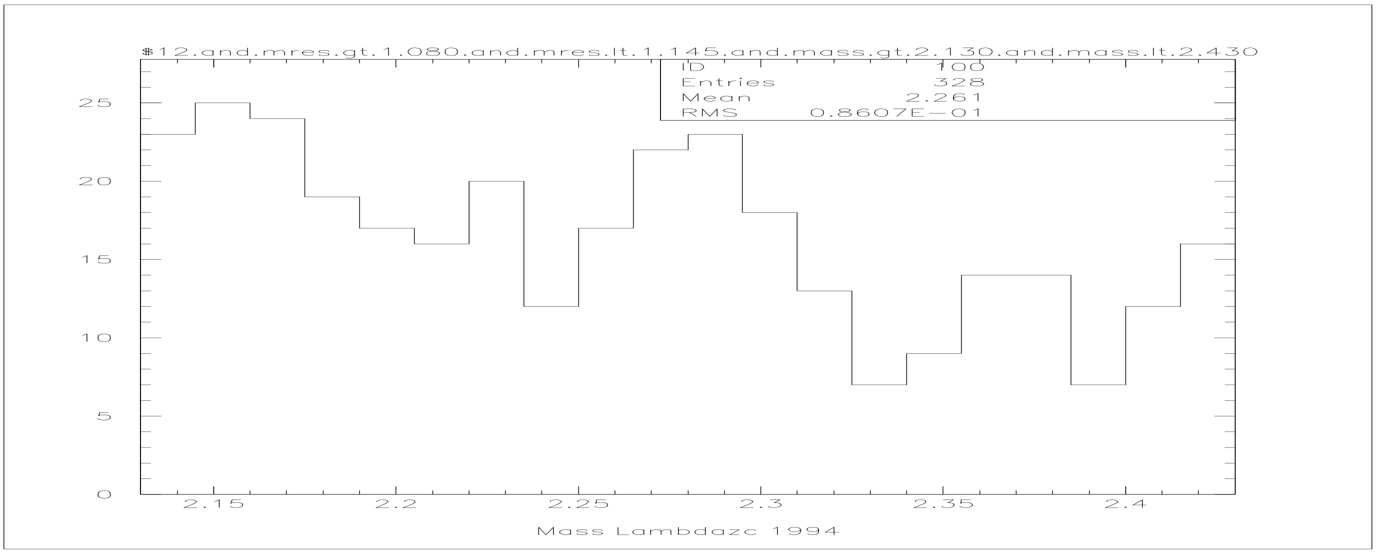

Fig. 1. $\Lambda\pi$ decay mode with no z or p_t cuts.

3 Tracking and Vertexing

All detected tracks are stored in a tracking object. The tracking object can extrapolate both it's position and errors along it's path. This allows us to propagate tracking errors to candidate final states.

Charged tracks are stored by the standard five H1 tracking parameters (R, ϕ, θ, the distance of closest approach to the origin (dca), and z_0). A subclass is used to represent tracks that come from a secondary vertex. Neutrals are stored in a separate subclass of tracks and are known by a point on it's trajectory, and a 4-momentum.

Errors are extrapolated by using the appropriate Jacobian matrices to transform the track covariance matrix. The method "covMatrixToVertexError" transforms the covariance matrix into a position error along the track trajectory. The method "derivativeMatrix" returns the derivative matrix required to transform the covariance matrix from the track's internal representation into the representation required for the candidate we are building[3].

I use a simple vertexing algorithm. The helices of two charged tracks are intersected; first in the $x-y$ plane and then if there is still an ambiguity by using the z information. I intersect a helix and a line for the intersection of one charged and one neutral track. Finally the point of "minimum total distance" is used if I am intersecting two neutral tracks.

If more than two candidate tracks are required in the same vertex, then we simply ask the third and subsequent tracks what the point of closest approach of that track to the "seed" vertex.

4 Targets and Candidates

The strategy for resonance finding is to first build lists of candidate particles. A decay object describes each possible decay mode of the resonance. A chooser object is used to construct all possible combinations of that decay. For instance, if an event has 2 possible K^+ particles, and 3 possible π^-, there are six possible $D^- \to K^+\pi^-\pi^-$ decays. A spin lock is placed on the detected tracks in each decay product candidate, to insure that the same track is not used twice in the same decay.

Once we have a viable decay mode, we subject the candidate resonance to “cuts”. Each target final state has an associated cut object. Each candidate then calls back to the cut object by invoking the method “passCut”. Each different subclass of the cut object carries out the different cuts associated with a particular physics entity (such as vertex, track, decays, etc), or even different cuts based on the particular target state.

Notice the above description can be abstracted from resonance finding to any final physics state. Topologies (such as requiring missing energy), or requiring correlations can be added to the system.

The objective description of a candidate allows all final states to be handled within one framework. All candidates hold a 4-vector description of themselves. Users then can query the candidate for it’s 4-vector and invariant mass, Lorentz boost the candidate, get the detected tracks that make up the candidate, compute the opening angles between individual tracks or between composing sub-candidates, etc.

Different subclasses of Candidate implement different types of final states. There are “topology” candidates and “particle” candidates. Particle candidates can be stable or decay. Decaying candidates are implemented as a subclass of particle candidates.

5 Conclusion

The objective description of physics data analysis is quite powerful. It allows users to systematically search for, apply cuts on, and extract information about abstractly defined final states. Significant modifications to the analysis can be made by subclassing physics classes as opposed to replacing any of the central code of the system.

References

[1] J.C. Anjos et al. Photon–Gluon Fusion Analysis of Charm Photoproduction. *Phys.Rev.Letters*, 65(20), November 1990.

[2] P.F. Kunz. Physics analysis tools. Technical report, SLAC–PUB–5520, 1991. Invited talk at 9th Conference on Computing in High Energy Physics, Tsukuba, Japan.

[3] O.I. Dahl, T.B. Day, F.T. Solmitz, and N.L. Gould. SQUAW Kinematic Fitting Program. Group A Programming Note P-126, University of California, Lawrence Radiation Laboratory, July 1968.

[4] H1 Collaboration. Inclusive D^0 and $D^{*\pm}$ Production in Deep Inelastic ep Scattering at HERA. Technical Report 96-138, DESY, 1996. *Submitted to Z.Phys.C.*