

A C++ framework for automatic search and identification of resonances

S. Chekanov

HEP division, Argonne National Laboratory, 9700 S.Cass Avenue, Argonne, IL 60439, USA

E-mail: chekanov@mail.desy.de

Abstract

This paper describes the first proof-of-concept version of a C++ program designed for peak searches in invariant-mass distributions. The program can be useful for searches of new states as well as for the reconstruction of known resonances.

Presently, there is a considerable progress in understanding the rich spectrum of hadron resonances. However, even in case of known baryonic resonances, there are still many open questions. The Particle Data Group (PDG) quotes more than 100 baryonic states, but only half of them are reasonably well established. Recently, the revitalized interest to baryon spectroscopy was triggered by observation of narrow peaks in the $K^\pm n$ and $K_S^0 p$ invariant-mass distributions which can be interpreted as pentaquarks. There is also longstanding interest to other exotic multiquark states, glueballs, hybrids, baryonia, etc. The HERA experiments have active program for searching such states [1]. At the LHC this physics program will continue.

In hadron spectroscopy, searches for new resonances and measurements of known states are based on the reconstruction of invariant-mass distributions of two or more tracks in order to find the rest mass of the originally decaying particle. The usual procedure for such studies is to assign certain masses to tracks, and then to combine their four-momenta to form the invariant-mass distributions. From the observed peaks, one can determine resonance masses, widths and cross sections.

Obviously, a particle identification is highly desirable for any experiment in order to reduce combinatorial background. Without the ability to identify tracks, the combinatorial background rises as $\sim 0.5n^2$, where n is the number of produced particles used in the searches. Thus high-energy experiments have to deal with a significant background for searches in fully inclusive events. For the reconstruction of three- and four-body decays, the combinatorial background is even higher than for the two-body decays.

Particle identification can be achieved using various approaches, such as the energy loss per unit length (dE/dx), silicon-strip detectors, Cherenkov counters etc.. In many cases, the particle identification cannot be perfect. For example, when the dE/dx method is used for pentaquark searches, it is difficult to obtain a high-purity sample of kaons or protons for tracks with large momenta ($p > 1$ GeV) due to significant overlap of the dE/dx bands for different particle species in this momentum range. In this case, several mass assumptions for invariant-mass distributions are needed to be checked in order to exclude peaks due to possible misidentification.

Searches for new resonances using various mass assumptions remain to be a tedious task since no much progress has been made so far to develop a tool which can automate this procedure. For example, the reconstruction of two-, three-, four-body decays involving only three mass assumptions leads to 36 possible non-identical invariant-mass distributions (6 distributions for two-body decays, 10 - for three-particle decays and 20 - for four-particle decays). All these invariant-mass distributions should be reconstructed, analyzed and possible reflections from known PDG states, when different mass assumptions are used, should be disregarded. Obviously, this time consuming work can be simplified and automated.

The program called "SBumps", which is still at the early stage of the development, attempts to accomplish this task. It rather represents the first proof-of-concept version of a program which helps to

perform automatic searches of peaks in invariant-mass distributions. To perform the reconstruction of invariant-mass distributions, the user should specify:

- the event record, i.e. a list of tracks and (optionally) the probabilities that tracks belong to certain particle species. Such probabilities can be obtained using various particle-identification technics;
- the names of particle species used during the mass assignments;
- the statistical significance of expected peaks;
- instrumental invariant-mass resolution;
- how many tracks should be combined to the invariant-mass distributions.

Once the initial conditions are specified, the program runs over the event list and reconstructs all possible invariant-mass distributions with the mass assumptions as specified by the user. SBumps can reconstruct at the same time 2-, 3- and 4- body decays in different combinations. At the end of the run, the program saves all the created invariant-mass distributions to the ROOT histograms [2].

At the second stage, the program analyses the created distributions and attempts to find statistically significant peaks. SBumps uses the build-in ROOT fast peak finder (from the TSpectrum class), which uses a fast deconvolution method [3] based on a Markov approach for peak searching in presence of a background and statistical noise. This algorithm was mainly developed for narrow, high-amplitude peaks which are characteristic for γ -ray physics. Therefore, this algorithm is not completely appropriate for peak searches in the invariant-mass spectra without tunings of the initial parameters of this algorithm. In case of the SBumps package, the deconvolution techniques is only used to identify the so-called seeds, i.e. bins with positions of possible peaks above a smooth background. The number of seeds can be rather large, and not all of them correspond to statistically significant peaks. At this stage, several adjustable parameters are available, such as the resolution of neighboring peaks, the peak sensitivity and the peak thresholds. Such parameters can be set using the steering file before each run.

Next, the program evaluates each seed by calculating the statistical significance, $N(S)/\Delta N$, where $\Delta N = \sqrt{N(B) + N(S)}$, with $N(S)$ $N(B)$ being the number of the signal (background) events. The calculation was done by comparing the values at the seed positions with the background level, which is determined from the neighboring bins. Then the peaks, which have the statistical significance above the level specified by the user, are considered for further analysis. This part of the program can be further improved, introducing more complicated algorithms. For example, sufficiently broad resonances might be overlooked by the present prescription.

The two-step procedure described above simplifies the peak search since the user normally does not need to deal with tunings of the initial parameters for the fast deconvolution method. At the same time, the method is sufficiently fast and can possibly be extended by introducing other algorithms to identify the peaks. For example, algorithms based on a fitting procedure can also be used; in this case, the seed positions can be used as the initial parameters of the fit functions.

At the third stage, the program attempts to identify the found peaks by comparing them with the masses of known PDG states. For this, a look-up table containing the information on established PDG resonances is used. Since the errors on the reconstructed peaks and the errors on the masses of known resonances taken from the PDG look-up table are known, the program matches the peaks using a simple criteria: $|L|/\Delta L < S$, where S is a free parameter given by the user, L is the distance between the PDG mass and the peak position and ΔL is the error on the reconstructed peak position combined with the error on the mass of known PDG resonances. For this matching procedure, the information on charge of decaying resonance is properly taken into account. In future, the information on specific decay channel can also be taken into account in order to reduce misidentification in the matching procedure.

Figure 1 shows the output of the SBumps program for events generated with PYTHIA Monte Carlo model. This example shows one histogram for two-body decays with two automatically identified peaks corresponding to known states. The required statistical significance for the final peaks was 4σ .

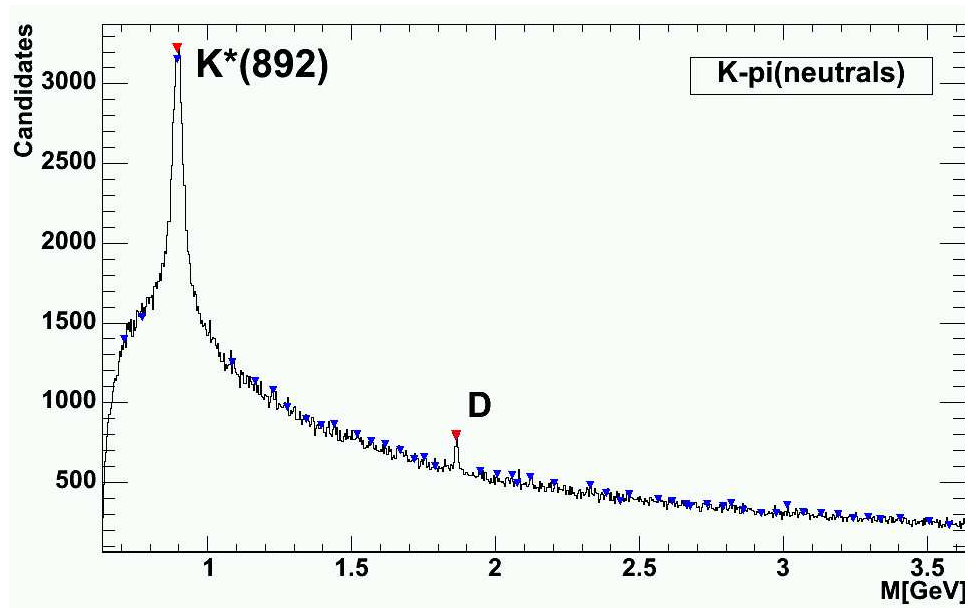


Fig. 1: The two-particle invariant-mass distribution calculated using the events generated with the PYTHIA Monte Carlo model (pp collisions at 1.4 TeV), when two mass hypothesis are used (K^\pm and π mesons). The blue triangle symbols show the seeds used for the calculations of statistical significance. The seeds were found by using the fast deconvolution method. Most of the seed peaks were disregarded after the final calculation of the statistical significance. The program correctly identifies the known PDG states, K^* and D^0 , after comparing the reconstructed peak positions with the PDG look-up table.

This example shows that the program can easily find and identify rather broad resonances (K^*), as well as narrow peaks which do not have high statistical significance.

The proof-of-concept version of the SBumps program is available as a loadable RMC file of the RunMC analysis framework [4]. After loading the “sbumps.rmc” module, RunMC first creates a ROOT tree with Monte Carlo events. The initial conditions are given in the file “sbumps.cards”. The executable file “sbumps.exe” can be used to run over the events. At the end of the calculations, the ROOT browser should display the reconstructed histograms with invariant-mass distributions. The peaks which have the statistical significance above that specified by the user should be labeled.

References

- [1] HERMES Coll., A. Airapetian et al., Phys. Lett. **B 585**, 213 (2004);
 ZEUS Coll., S. Chekanov et al., Eur. Phys. J. **C38**, 29 (2004);
 ZEUS Coll., S. Chekanov et al., Phys. Lett. **B591**, 7 (2004);
 ZEUS Coll., S. Chekanov et al., Phys. Lett. **B610**, 212 (2005);
 H1 Coll., A. Aktas et al., Phys. Lett. **B588**, 17 (2004).
- [2] R. Brun and F. Rademakers, Nucl. Instrum. Meth. **A389**, 81 (1997);
 R. Brun, F. Rademakers, P. Canal and M. Goto, ECONF **C0303241**, MOJT001 (2003).
- [3] M. Morhac et al., Nucl. Instrum. Meth. **A401**, 113 (1997).
- [4] S. Chekanov, RUNMC - C++ *object-oriented framework for monte carlo models*. Preprint hep-ph/0411080, Comp. Phys. Comm. (in press), available on
<http://hepforge.cedrak.ru/runc/>
<http://www.hep.anl.gov/rmhac>