π^0 Reconstruction for Cleo III

Jon Kennedy

Department of Physics

Worcester Polytechnic Institute, Worcester, MA, 01609

Abstract

The purpose of this project is the implementation of π^0 reconstruction for the CLEO III experiment. A constrained fitter is used to fit relevant event data (calorimeter showers and angles) and reconstruct the π^0's 4-momentum and error matrix. The efficiency, purity and resolution of the algorithm is then compared with the previous CLEO II package

Introduction

The π^0 is an interesting particle, and there are on average 5 of them in a typical hadronic event. It is the lightest hadron with a mass of 135 MeV, which is about 260 times the mass of the electron. As its name suggests, the π^0 has zero electric charge but it also has zero spin. It has a very short lifetime, on the order of 10^{-16} sec, and even though it moves at very close to the speed of light it travels less than .001 mm. With its very small lifetime and neutral charge the π^0 is impossible to detect directly but is extremely important because it is a signal for many decay modes including $B \rightarrow \pi^0 l \nu$.

So how does one determine if a π^0 was present in an event? It is known that when the π^0 decays, it does so 98.8% of the time to two photons. Using principles of special relativity one can derive a relationship between the mass of the π^0, which is well known, and the energies and directions of the daughter photons. In principle then, with sufficient information about the photons in an event, one could deduce the existence of a π^0. In practice, more specifically with the CLEO detector, the events are extremely complex, there are many photons and other particles and there is of course experimental errors associated with the measurements. To determine if there were one or more π^0's in a given event, one must accumulate all of the information gathered by the detector concerning photons. Then using the relationship mentioned above, one must calculate which photons were most likely to have come from a π^0. The energy and directions of the daughter photons are then used to reconstruct the 4-momentum of the π^0. The main goal of this project is to develop a CLEO III package to reconstruct the π^0's in a given event.

Detection and Reconstruction

To reconstruct the π^0 and its 4-momentum, one needs the energies and directions of the photons in an event. To detect photons and measure their energies the CLEO detector make use of an electromagnetic calorimeter. This calorimeter is made up of Thallium-doped Cesium Iodide (CsI) crystals. The calorimeter detects both charged particles and high energy photons but since it is an electromagnetic calorimeter it is designed such that only electrons, positrons and photons deposit all of their energy. When a photon enters a crystal it pair produces an electron and a positron because of the presence of the Cesium atom. These electrons are accelerated by the fields in the crystal and they radiate more photons which then pair produce. Some of the electrons excite the Thallium which deexcites by emitting light. The light produced by these processes is converted to an electrical signal by four silicon photodiodes which are mounted on the rear face of each crystal and sensitive to the frequency of light emitted by the Thallium atoms when they deexcite. These signals are then shaped and analyzed by a computer. To determine all of the energy deposited by a given photon it is often necessary to analyze the light from several of the surrounding crystals using clustering algorithms, because the energy a particle deposits in the detector spreads over several crystals.

In a given event there are many showers and decays occurring and the situation in general is quite complex. In an effort to optimize the reconstruction process many calorimeter showers are discarded or cut from the list of possible showers because of non-desirable characteristics. For example, the calorimeter detects both charged particles and photons, but in the case of the π^0, only photon showers are of interest. So the first cut is to eliminate all showers matched to charged particle tracks in the drift chamber, since these showers were probably caused by charged particles.

Another cut often used is on the location and appearance of a given shower. There are certain areas of the calorimeter which are better than others. For example, the barrel of the calorimeter has much better resolution than the end caps for a couple of reasons. First, there is much more material between the drift chamber and the calorimeter in the end caps. This material is there to support the drift chamber wires but it also shields the end cap crystals and decreases their resolution. Secondly, the barrel crystals point toward the interaction region, whereas the crystals in the end caps do not. This also decreases the resolution of the end cap crystals. Therefore, cuts are made on the position of the shower in the calorimeter. It is also known that photon showers have certain characteristics that distinguish them from other showers. This is because it is an electromagnetic calorimeter and is designed to detect photons, electrons and positrons. If a shower does not have the typical characteristics of a photon shower, it is cut. In this way we narrow down our possibilities and make the problem more manageable.

With these cuts in place the actual process of reconstructing the π^0 can begin. In this process the resulting shower data is used to calculate the 4-momentum of the parent π^0. This is accomplished using a constrained fitting algorithm.

Constrained Fitting

After the general cuts discussed above are made, the resulting shower data is assumed to be the result of daughter photons of the π^0. With this assumption, the π^0 reconstruction package will reconstruct a large number of possible π^0's, and the user of the package will be able to select cuts which narrow down the candidates as he or she chooses.

To reconstruct the π^0's whose daughter photons caused the remaining calorimeter showers, a constrained fitting algorithm is used. A constrained fit is a mathematical algorithm which improves the measurements of a process by utilizing the physical laws governing that process. In the case of the π^0, it is known that it decays 98.8% of the time to two photons and is a relativistic particle, so one can derive the following relationship from relativity:

$$m_{\pi^0}^2 = 4E_1E_2\sin^2(\frac{\theta}{2}) \tag{1}$$

where $m_{\pi^0}^2$ is the invariant mass of the π^0, E_1 and E_2 are the energies of the photons and θ is the opening angle between them. It is assumed that the photons leave all of their energy in the calorimeter so E_1 and E_2 are also known as the energies of the calorimeter showers. This equation is also often written in spherical coordinates because the positions of the crystals in the calorimeter are described in this coordinate system. In spherical coordinates Eq. (1) takes the form:

$$m_{\pi^0}^2 = 2E_1E_2(1 - \cos\phi_1\sin\theta_1\cos\phi_2\sin\theta_2 - \sin\phi_1\sin\theta_1\sin\phi_2\sin\phi_2 - \cos\theta_1\cos\theta_2) \tag{2}$$

where θ_1 and ϕ_1 are the angles of the shower E_1, and θ_2, ϕ_2 correspond to E_2.

Since the mass of the π^0 is a well known quantity, one can use a constrained fit algorithm to adjust the measured values of E_1, E_2, θ_1, ϕ_1, θ_2, ϕ_2 to satisfy Eq (2). With these fitted values one can proceed to calculate the 4-momentum of the π^0.

To perform a constrained fit, one minimizes the χ^2 with the constraint equation using the method of Lagrange multipliers. In general this minimization is a nonlinear problem and so a typical simplification is an expansion to first order about an approximate solution. In our case, the approximate solution about which the expansion is performed is the experimentally measured values. The Lagrange multipliers, symbolized by λ, are added variables which allow for the constraint equation to be included. Depending on the number of constraint equations, the method of Lagrange multipliers generally involves many matrix calculations. However, in the case of the π^0 where there is only one constraint, many of the matrices reduce to numbers and the calculations become much easier. In the particular case of the π^0 the relevant matrices of the algorithm (as defined in [1]) take the following form:

$$\alpha = (E_1, \theta_1, \phi_1, E_2, \theta_2, \phi_2)$$

$$\begin{aligned}
\mathbf{D}(1,1) &= -2E_2(1 - \cos\phi_1\sin\theta_1\cos\phi_2\sin\theta_2 - \sin\phi_1\sin\theta_1\sin\phi_2\sin\phi_2 - \cos\theta_1\cos\theta_2) \\
\mathbf{D}(1,2) &= E_1E_2(\cos\phi_1\cos\theta_1\cos\phi_2\sin\theta_2 + \sin\phi_1\cos\theta_1\sin\phi_2\sin\theta_2 - \sin\theta_1\cos\theta_2) \\
\mathbf{D}(1,3) &= E_1E_2(-\sin\phi_1\sin\theta_1\cos\phi_2\sin\theta_2 + \cos\phi_1\sin\theta_1\sin\phi_2\sin\theta_2) \\
\mathbf{D}(1,4) &= -2E_1(1 - \cos\phi_1\sin\theta_1\cos\phi_2\sin\theta_2 - \sin\phi_1\sin\theta_1\sin\phi_2\sin\phi_2 - \cos\theta_1\cos\theta_2) \\
\mathbf{D}(1,5) &= E_1E_2(\cos\phi_1\sin\theta_1\cos\phi_2\cos\theta_2 + \sin\phi_1\sin\theta_1\sin\phi_2\cos\theta_2 - \cos\theta_1\sin\theta_2) \\
\mathbf{D}(1,6) &= E_1E_2(-\cos\phi_1\sin\theta_1\sin\phi_2\sin\theta_2 + \sin\phi_1\sin\theta_1\cos\phi_2\sin\theta_2)
\end{aligned}$$

$$\mathbf{V}_{\alpha_0} = \begin{pmatrix} \sigma_{E_1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{\theta_1} & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{\phi_1} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{E_2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{\theta_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{\phi_2} \end{pmatrix}$$

$$\mathbf{d} = (.135)^2 - 2E_1E_2(1 - \cos\phi_1\sin\theta_1\cos\phi_2\sin\theta_2 - \sin\phi_1\sin\theta_1\sin\phi_2\sin\phi_2 - \cos\theta_1\cos\theta_2)$$

where α is the vector containing the quantities we want to solve for. $\mathbf{D}$ is a 1×6 matrix containing the derivatives of the constraint equation w.r.t. the six variables evaluated at a particular value of α, and is written in component form for clarity. $\mathbf{V}_{\alpha_0}$ is the initial error or covariance matrix, it is diagonal because the measurements are all independent of one another. $\mathbf{d}$ represents the value of the constraint equation at a particular value of α. From these matrices all of the other matrices α, $\mathbf{V_D}$, $\mathbf{V}_\alpha$ can be calculated.

The fitting algorithm starts with the experimentally measured values of $E_{1,2}$, $\theta_{1,2}$, $\phi_{1,2}$. Using the matrices given above and the calculations as described in [1], these initial values are changed or fit to better satisfy the constraint Eq. (2). These fit values are then used as the initial values for the next iteration. This process is repeated until the χ^2 converges to some minimum value or, in the case when the fitting fails, is stopped if the χ^2 diverges. Table 1 shows how shower data from the detector is fit to more accurately satisfy Eq (2).

TABLE 1. Measured values and Fitted values

Variable	Measured Value	Fitted Value
E_1	.9423	.9383
θ_1	2.34	2.28
ϕ_1	2.00	2.01
E_2	1.20	1.19
θ_2	2.10	2.16
ϕ_1	2.05	2.03

To determine if the fitted values are indeed better values for the daughter photons, one needs to use Eq(2). The value of $m_{\pi^0}^2$ is .018225 GeV. The measured values give a value of .06657 GeV (using Eq (2)), while the fitted values yield a value of .0163 GeV. So here we see that the fitter has adjusted or pulled the measured values so that they more accurately satisfy Eq (2). after only two iterations.

With the fitted values for the energy of the photons and their directions in space, the 4-momentum of the π^0 is calculated. However, one still has to face the question of whether the reconstructed π^0's are real π^0's. This issue needs to be addressed because of the very general assumptions made concerning the photon showers. This final level of selection is performed by the user of the package in his or her own analysis.

Analyzing the Fitter and Analyzing Events

The very liberal assumption of considering all showers remaining after the initial cuts to be the result of a π^0 decay is made because the π^0 reconstruction package is designed to be as efficient as possible at the cost of being impure. Efficiency is the ratio of the total number of correctly reconstructed particles to the actual number in the event. Purity is the ratio of the correctly reconstructed particles to the number of reconstructed particles. If an algorithm is highly efficient it reconstructs all of the particles, it may reconstruct many others, but it finds all of the correct ones. In a very pure algorithm there is a one-to-one correspondence between correctly reconstructed particles and actual particles.

The purity level of the reconstructed π^0's can be increased after the constrained fitter is run. For example, the user of the package can make a cut on the χ^2 of the constrained fit. The constrained fitting algorithm is a minimization routine, it minimizes the χ^2 by fitting the values of the measured variables. If the χ^2 was relatively large, the fit was poor and the photon showers probably did not originate from a π^0. Using cuts such as this the individual user can select the best looking reconstructed π^0's, thereby increasing the purity.

To find the efficiency and purity of the algorithm with no user cuts, the algorithm is run on Monte Carlo simulations. A Monte Carlo simulation is simply a simulation of the detectors response to many types of generated particles and decays. In creating the Monte Carlo simulation the user can set up the event as specifically or as generally as they desire and can find out everything about the event after it has occurred.

The initial task of this project was to determine the efficiency and the purity of the CLEO II package. To do this, the package was run on two different types of Monte Carlo simulations. The first simulation was simply a single π^0 decaying into two photons. The second was a much more physically realistic situation where there were many types of particles and many decays occurring. So in this way we could compare how the package performs in an ideal situation, to how it performs in a more general environment.

This evaluation of the CLEO II code was done to determine the performance of the package, but it was also performed so it could be compared to its CLEO III C++ counterpart. However, the efficiency and purity of the CLEO III code could only be determined in the single π^0 Monte Carlo environment. This is because there does not yet exist tagging software for CLEO III.

With CLEO II software, one could determine if two showers had a common parent and so they could be "tagged" to the particle which created them. The identity of this particle could then be found. This kind of software is not available in CLEO III and so the C++ package could not be tested on a generic Monte Carlo file. Moreover, the purity of the CLEO III package was assumed to be 100% in the case of the single π^0 Monte Carlo because there is no way to calculate this more accurately without some sort of tagging software.

Another measure of an algorithm's performance is how well the reconstructed 4-momentum matches that of the actual particle. This is known as the resolution. For the CLEO II package the energy, momentum and direction resolution was found for both the single π^0 and generic simulations. However, due to time constraints and the absence of tagging software, only the energy resolution on the single π^0 simulation was calculated for the CLEO III C++ package.

Results

The efficiency, purity and resolution was calculated for the CLEO II π^0 package on both single π^0 and generic Monte Carlo files. A constrained fitting algorithm was written for the CLEO III detector and implemented in a class which creates objects that correspond to particles that decay to two photons, including the π^0. The fitter was tested as a stand-alone program and as function of the class. The efficiency, purity and energy resolution of this algorithm on a single π^0 Monte Carlo was calculated and compared to the CLEO II package.

The efficiency and purity of the two packages is summarized in Table 2. From this table one can see that the purity of the CLEO III package is higher, and the efficiency is lower than that of the CLEO II package. One reason for this discrepancy could be the fact that we had to assume 100% purity in testing the CLEO III algorithm because of the absence of tagging software. We expect that when the CLEO III package is tested on a single π^0 Monte Carlo simulation with the aid of tagging software the purity will decrease and the efficiency will increase slightly. Another possible factor could be that we are using two different types of Monte Carlo to test the algorithms. The CLEO II package was run on CLEO II Monte Carlo while the CLEO III was run on CLEO III Monte Carlo. The differences in the Monte Carlo simulations could be a factor in the comparison of the performance.

TABLE 2. Purity and Efficiency Results

Package	Purity	Efficiency
CLEO III C++ (Single π^0) MC	100.0%	77.1%
CLEO II Fortran (Single π^0) MC	97.6%	82.7%
CLEO III C++ (Generic) MC	n/a	n/a
CLEO II Fortran (Generic π^0) MC	16.6%	49%

Table 3 gives the energy resolution values for the CLEO II and CLEO III π^0 packages. The resolution is calculated as a function of energy bins to determine if the resolution changes for different energy intervals. The resolutions are calculated using the sigma generated by fitting a Gaussian to the distribution. A typical CLEO III energy resolution plot and its Gaussian fit are shown in Figures 1 and 2.

From Table 3 we can see that the energy resolution of the two packages are in agreement if one takes into account the error. One can also see that the resolution values for CLEO III are all slightly larger than their corresponding CLEO II values. This could be some kind of systematic effect but more analysis needs to be conducted to determine if it is significant.

TABLE 3. Resolution Values for Single Monte Carlo π^0 5000 events

Energy Bin (GeV)	CLEO II Fortran Package Res (%)	CLEO III C++ Res (%)
1 (0.00...0.25)	$.84 \pm .09$	$.96 \pm .06$
2 (0.25...0.50)	$1.60 \pm .07$	$1.71 \pm .06$
3 (0.50...0.75)	$1.75 \pm .09$	$1.79 \pm .06$
4 (0.75...1.00)	$1.68 \pm .07$	$1.83 \pm .07$
5 (1.00...1.50)	$1.62 \pm .05$	$1.68 \pm .04$

The CLEO II software resolution, in bins of energy as defined in Table 3, of momentum, θ, and ϕ for both the single π^0 and the generic Monte Carlo are shown in Figure 3, Figure 4, and Figure 5 respectively. From these figures one can see a general trend in the resolution values, which can be explained nicely by the theory. In Figures 3, 4 and 5, one should notice that at low energy the resolution is relatively high, then at some medium energy value the resolution reaches a minimum, then the resolution begins to increase as the energy increases. This behavior can be explained if one looks at Eq (1). Since the *sin* function is bounded and $m^2_{\pi^0}$ is constant, natural limits exist on the simultaneous values of $E_{1,2}$ and θ. At low energy the value of θ must be relatively large. Due to this large angle the resolution of the

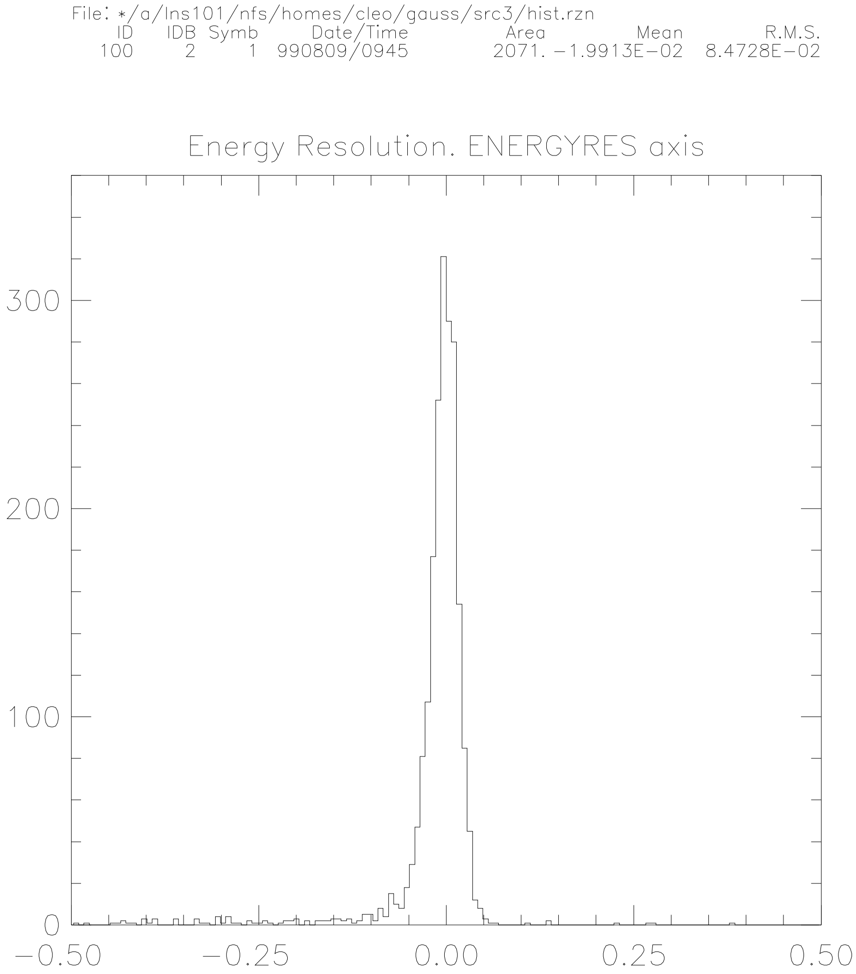

FIGURE 1. CLEO III Energy Resolution in the energy interval $(0.25, 0.50)$ GeV

π^0 is decreased. At high energy the value of θ must be small which causes the showers to overlap and decreases the resolution. In the middle energy range, where θ is also in between extremes, we should see the best resolution and this is in fact what we see.

Conclusions

The constrained fit needs to be tested on a generic Monte Carlo simulation. While it has been tested on a Monte Carlo simulation consisting of a single π^0, it needs to be run on a more realistic physical situation. The main reason why it was not tested on such a simulation is because there does not exist at this time any means of tagging the particles in CLEO III simulations. Once such a tagger is developed, the π^0 package should be tested on a generic Monte Carlo simulation and compared to the CLEO II package.

Acknowledgments

I would first like to thank Professor David Cassel of Cornell University for accepting me into this REU program and working so hard to make it run smoothly. Secondly, I would like to thank Véronique Boisvert for her infinite patience and assistance during this entire

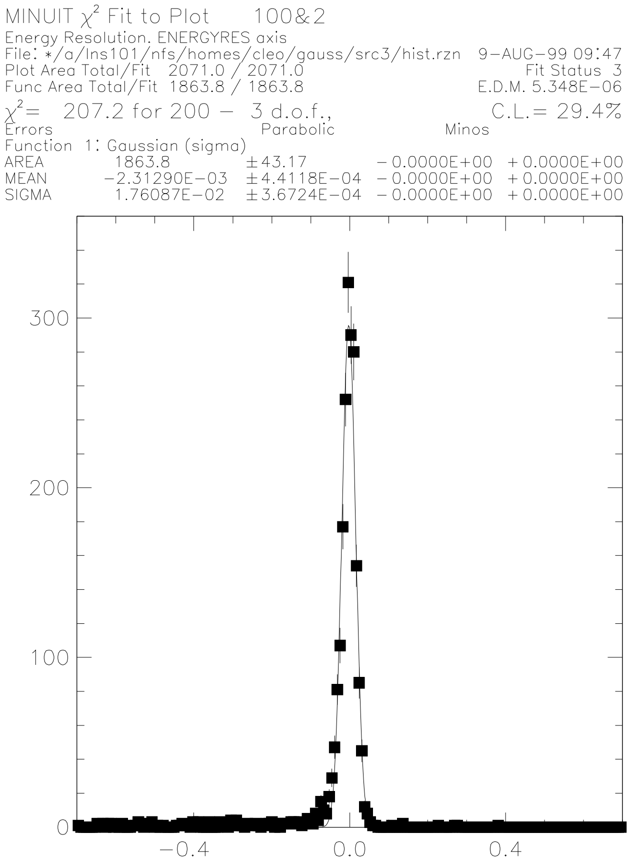

FIGURE 2. CLEO III Energy Resolution Fit in the energy interval $(0.25, 0.50)$ GeV

project. I would also like to say thank you to Dr. Brian Heltsley for his help with the actual physics of the problem, the mathematics of the fitter as well as his much needed help with CLEO III software. I would also like to thank Professor Lawrence Gibbons for his help on the complex technical issues.I would wish to thank Aren Jansen, the C++ master, who would always answer one of my stupid questions about an unfamiliar language. Finally, I would like to thank Chris Jones and Merlin Meyer-Mitchell for their help with CLEO III software.

This work was supported by the National Science Foundation REU grant PHY-9731882 and research grant PHY-9809799.

Footnotes and References

1. Paul Avery, " Fitting Theory I: General Least Squares Fitting Theory", October 18, 1991
2. Paul Avery, " Fitting Theory VI: Formulas for Kinematic Fitting ", June 9, 1998
3. "The Crystal Calorimeter" http://www.lns.cornell.edu/public/lab-info/cc.html

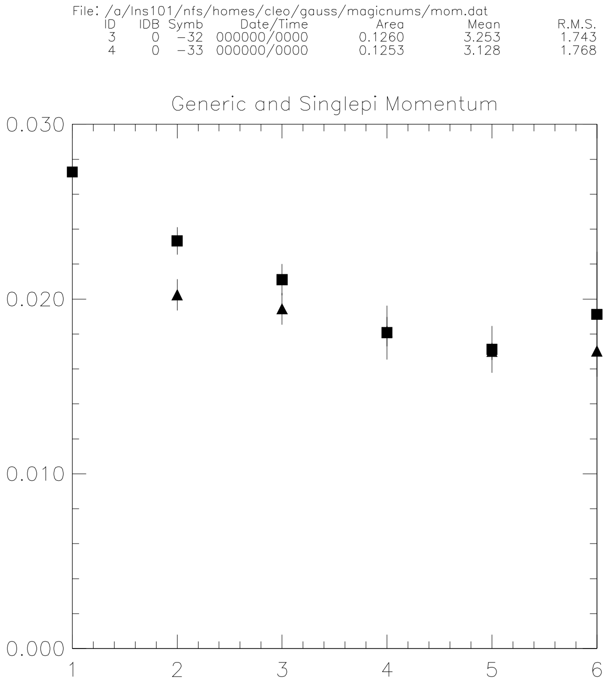

FIGURE 3. Comparison of Generic MC and Single π^0 CLEO II Momentum Resolution, Squares are Generic values, Triangles are Single π^0

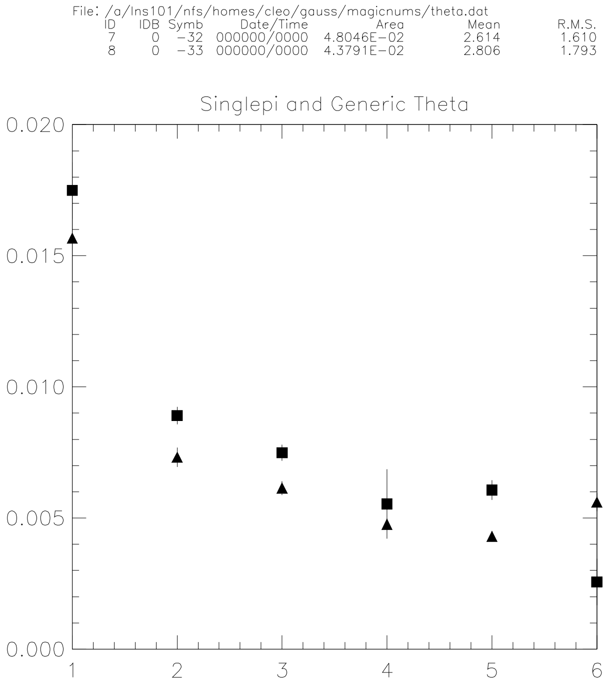

FIGURE 4. Comparison of Generic MC and Single π^0 CLEO II θ Resolution, Squares are Generic values, Triangles are Single π^0

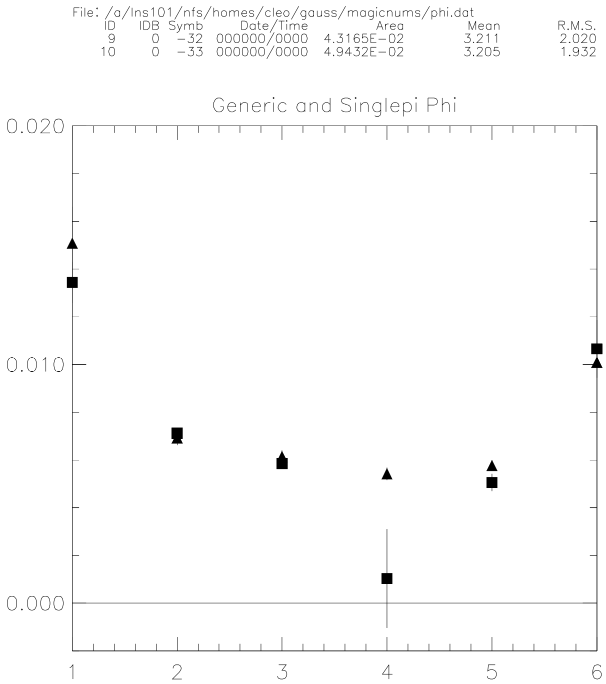

FIGURE 5. Comparison of Generic MC and Single π^0 CLEO II ϕ Resolution, Squares are Generic values, Triangles are Single π^0