Technische Universität München

Bachelor Thesis

Calibration Software for the HADES Electromagnetic Calorimeter

Author:
Dimitar Mihaylov

Supervisors:
Prof. Dr. Laura Fabbietti
Dr. Kirill Lapidus

A thesis submitted in fulfilment of the requirements for the degree of Bachelor of Science

in the

Physics Department

July 2012

Declaration of Authorship

I, Dimitar Mihaylov, declare that this thesis titled, 'Calibration Software for the HADES Electromagnetic Calorimeter' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: Dimitar Mihaylov

Date: 24.07.2012

TECHNISCHE UNIVERSITÄT MÜNCHEN

Abstract

Chair E12 for Experimental Physics
Physics Department

Bachelor of Science

Calibration Software for the HADES Electromagnetic Calorimeter

by Dimitar MIHAYLOV

The High Acceptance Di-Electron Spectrometer (HADES) is located at the GSI Helmholtz Centre for Heavy Ion Research. It is dedicated to study dense nuclear matter. This is achieved by means of di-electron spectroscopy in nucleus-nucleus collisions. In order to extend the capabilities of the spectrometer to the full-reconstruction of Dalitz and two-photon decays an electromagnetic calorimeter (EMC) has been proposed as a part of the HADES upgrade. It will allow to detect neutral pseudoscalar mesons (π^0, η) by identifying their decay into two photons.

The production of neutral mesons has been studied for collision energies below 2 AGeV and above 40 AGeV. No data, however, have been recorded for energies in the range of 2-40 AGeV. To address this issue it has been proposed to use a lead glass electromagnetic calorimeter together with HADES. HADES is currently operating at the SIS18 synchrotron at energies of up to 2 AGeV. As a part of the FAIR project a new synchrotron SIS100 will be available. SIS100 can provide heavy system collisions in a fixed target configuration at energies of up to 8 AGeV. Thus, measurements of neutral meson production can be carried out for the very first time in the 2-8 AGeV energy range with the EMC. Furthermore, the electron/pion separation will be significantly improved and the HADES strangeness program can also benefit from photon measurements.

The data analysis of the signal from the calorimeter is a complex process. The photon reconstruction procedure has sizeable systematic errors related to unaccounted energy losses inside the calorimeter. Therefore, a calibration procedure is required. The objective of this thesis was to develop a calibration procedure suitable for the EMC at HADES. The mass of the neutral pion was used as a reference point for the calibration. The calibration procedure was implemented in a standalone program that uses only standard C++ and ROOT libraries.

Acknowledgements

I would like to express my gratitude to everyone from the Kaon Cluster Group, especially to the group leader Prof. Dr. Laura Fabbietti for giving me the opportunity to work in such a great team. I am very thankful to my supervisors Dr. Kirill Lapidus and Prof. Dr. Laura Fabbietti for their guidance, patience and support during my Bachelor study.

I would like to thank my girlfriend Iva Hristova who, despite having to suffer times in which I was quite occupied with this work, gave me her full support.

And last but not least I would like to thank my family for their financial and more importantly moral support throughout my studies.

Contents

List of Figures

Abbreviations

EMC	**E**lectro**m**agnetic **C**alorimeter
FAIR	**F**acility for **A**ntiproton and **I**on **R**esearch
GSI	**GSI** Helmholtz Centre for Heavy Ion Research
HI	**H**eavy-**I**on
IM	**I**nvariant **M**ass
IMS	**I**nvariant **M**ass **S**pectrum
PMT	**P**hoto**m**ultiplier **T**ube
SIS	Heavy-Ion Synchrotron (**S**chwer**i**onen**s**ynchrotron)
tRPCs	**t**iming **R**esistive **P**late **C**hamber**s**

Chapter 1

The HADES experiment

1.1 Introduction

The High Acceptance Di-Electron Spectrometer (HADES) is located at the GSI Helmholtzzentrum für Schwerionenforschung in Darmstadt, Germany. The main purpose of this experiment is to investigate the hadron properties inside nuclear matter at normal and high nuclear densities [1]. A state with increased nuclear density can be created for up to 10 fm/c in heavy ion collisions at energies of 1-2 AGeV. These conditions are expected to result in modifications of basic hadron properties such as mass, decay width, etc [1]. The experimental verification could be done by investigating the resulting hadron decays into electron-positron pairs.

A new Facility for Antiproton and Ion Research (FAIR) is under development [2]. The currently used synchrotron SIS18 is able to provide collision energies of up to 2 AGeV. The new synchrotron SIS100 will be able to deliver beam energies up to 8 AGeV and the synchrotron SIS300 can reach HI collision energies of 35 AGeV [3]. No experiment so far supplied dilepton data in the energy range 2-40 AGeV [4]. The HADES experiment will operate at SIS100 and investigate hadron properties at HI collision energies of up to 8 AGeV [5].

1.2 Detector system

The HADES detector is composed of 6 identical sectors. A schematic overview of the system (cross section of two sectors positioned opposite to each other) is shown in

Fig. 1.1. The azimuthal coverage is 85% and the polar angle is covered between 18° and 85° [6].

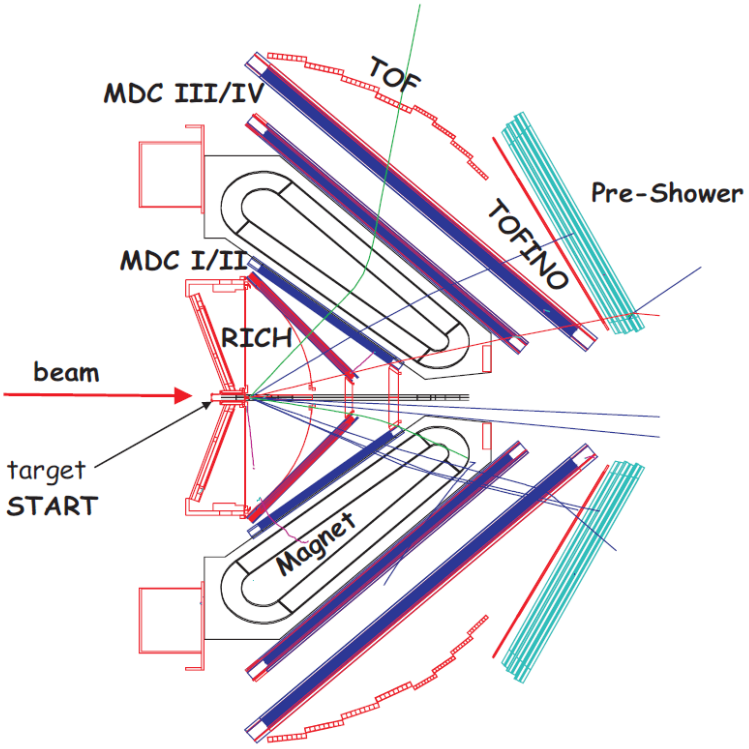

FIGURE 1.1: Schematic layout of the HADES detector [7].

The main components of HADES are a superconducting magnet, a START detector, a Ring Imaging Cherenkov (RICH) detector, two sets of Multiwire Drift Chambers (MDCs) and a Time-of-Flight detector [7].

The beam first hits the START detector and thus the initial time of interaction between the incoming particles and the whole system is registered. The RICH detector provides electron/pion separation: the electron/positron tracks can be distinguished from hadrons by the resultant Cherenkov radiation [1]. The charged particle tracks are reconstructed by four MDC planes, two located before and two after the magnetic field region.

The identification of a particle requires not only momentum measurement but also time of flight (TOF) measurement. This is achieved by the TOF/TOFINO wall. A recent upgrade at HADES saw the replacement of the TOFINO detector with the high-granularity timing Resistive Plate Chambers (tRPCs), which offer a much better time resolution [8].

In the present configuration of HADES photon detection is not possible.

Chapter 2

The HADES Electromagnetic calorimeter

2.1 Introduction

The interpretation of dielectron data relies on the understanding of all processes involved in HI collisions. However, currently there are no experimental data available for the production of neutral pseudoscalar mesons (π^0, η) at energies between 2-40 AGeV [4]. This means that the interpretation of the future dielectron data will entirely depend on theoretical models. If an electromagnetic calorimeter (EMC) is installed at HADES, it will provide for the very first time experimental data on the π^0 and η production at energies in the range of 2-8 AGeV. Additionally, the EMC will significantly improve the electron/pion separation at large momenta (above 400 MeV/c) and can be used for the HADES strangeness program, which addresses, among other goals, spectroscopy of $\Lambda(1405)$ and $\Sigma(1385)$ baryon resonances in elementary and HI reactions [6] [9].

2.2 Structure

The proposed EMC is based on lead-glass modules obtained on loan from the OPAL detector [6]. The lead glass has density of 4.06 g/cm^3, refractive index of 1.708 (at 410 nm) and a radiation length (X_0) of 2.51 cm. The total area of the calorimeter will be about 8 m^2 and will cover polar angles between 12° and 45° (Fig. 2.2) with almost

full azimuthal coverage. The photon energy resolution achieved in test experiments is $\sigma_E/E = 6\%/\sqrt{E/\mathrm{GeV}}$.

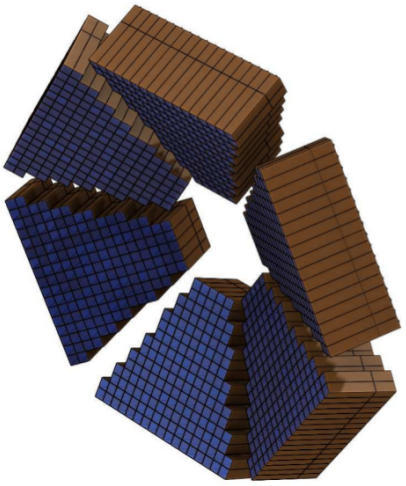

FIGURE 2.1: 3D view of the EMC arrangement [6].

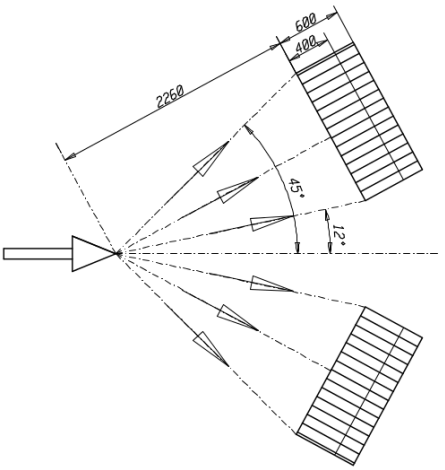

FIGURE 2.2: Layout of modules in vertical plane along the beam axis [6].

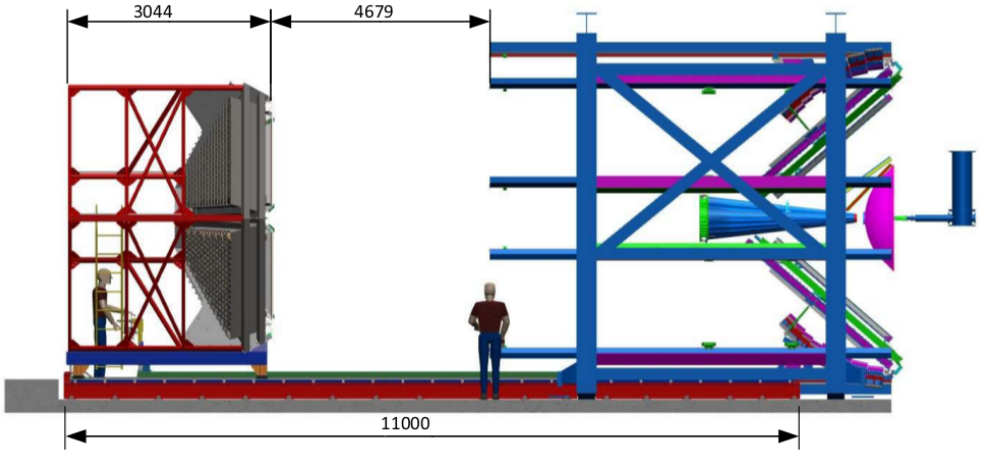

FIGURE 2.3: Side view of EMC with dimensions [6].

As described in chapter 1 the HADES detector system is divided into six sectors. The EMC will cover all of them and will be also divided into six sectors as shown in Fig. 2.1 and 2.4. The sectors will have a trapezoidal form, each of them containing 163 lead-glass modules [6]. The EMC will substitute the existing Pre-Shower detectors.

2.3 Detection process

The detection of photons is realized inside each module of the calorimeter. A single calorimeter module is schematically represented in Fig. 2.5.

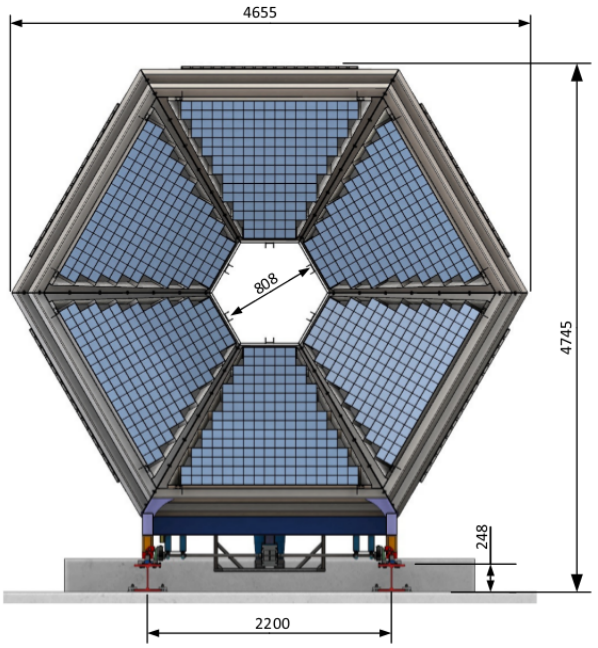

FIGURE 2.4: Front view of the EMC with dimensions [6].

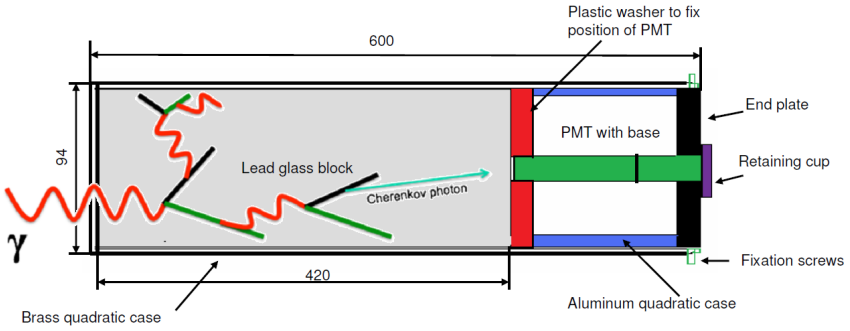

FIGURE 2.5: Schematic view of the calorimeter module [6].

Each module has a lead-glass block and a photomultiplier tube (PMT) located behind the block. An incident photon causes an electromagnetic shower inside the lead-glass block. The development of a shower is primarily dominated by pair production ($\gamma \rightarrow e^- + e^+$) and the Bremsstrahlung. Since the lead-glass has a high refractive index the speed of light inside the material is relatively low, thus the high-momenta electrons and positrons produce Cherenkov radiation. The Cherenkov photons are detected via the PMTs at the back-end of each calorimeter module (see Fig. 2.5).

Chapter 3

Data analysis

The signal produced in the detector comes from the PMTs at the back-end of each calorimeter module. The photons detected by the PMTs are the Cherenkov photons that result from the shower developed inside the lead glass block (see chapter 2). A complex software package allows the reconstruction of the photons entering the calorimeter blocks from the PMTs signals. Since the main objective of the experiment is to detect neutral mesons (π^0, η) through their decay into two photons it is essential to study the invariant mass spectrum (IMS) of photon pairs.

3.1 Invariant mass spectrum

The square of the four-momentum P^2 is a Lorentz-invariant quantity. For a single particle with mass m:

$$P^2 = m^2. \tag{3.1}$$

Hence the invariant mass (IM) of a two-particle system is

$$M_{inv}^2 := P^2 = (P_1 + P_2)^2 = m_1^2 + m_2^2 + 2E_1E_2 - 2p_1p_2cos\eta, \tag{3.2}$$

where P_1 and P_2 are the corresponding four-momenta of the particles, E_1 and E_2 are their corresponding energies, and η denotes the angle between them. If the two particles are coming from the decay of a mother particle, due to conservation laws the four-momentum of the mother particle is $P = P_1 + P_2$ and its mass is M_{inv}. For massless

decay products equation (3.2) can be simplified to:

$$M_{inv}^2 = 2E_1E_2(1 - cos\eta), \tag{3.3}$$

$$M_{inv} = \sqrt{2E_1E_2\,(1 - cos\eta)}. \tag{3.4}$$

The EMC allows to detect photons from the decays of mesons (most notably π^0 and η). Thus, using equation (3.4) and a sample of reconstructed photons, one can build an IMS. Figure 3.1 presents IMS for photons produced in a simulated Ni+Ni collisions at 8 AGeV. Each data point represents the number of photon pairs with a specific invariant

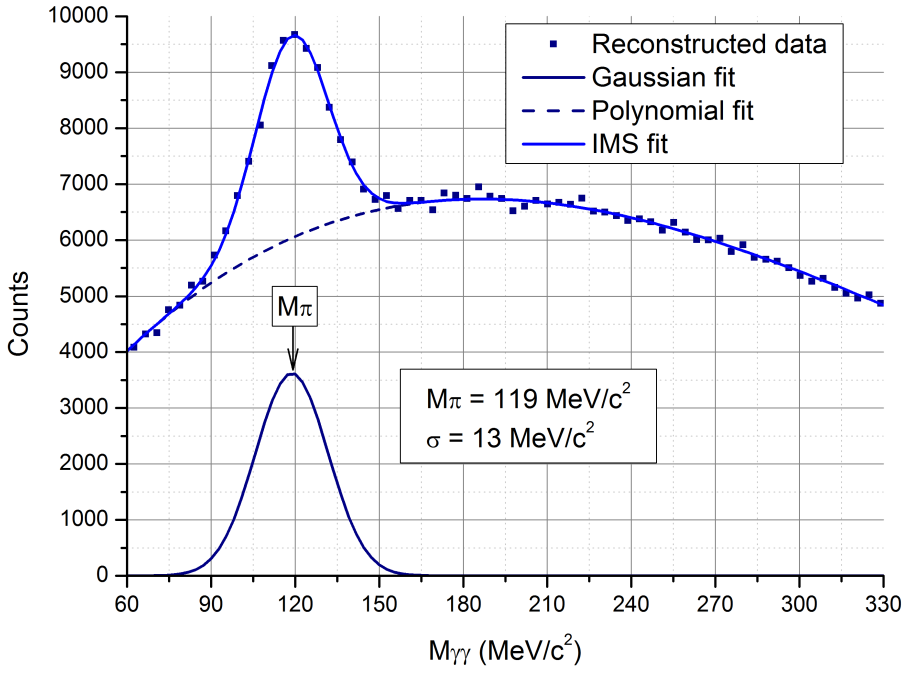

FIGURE 3.1: IMS for photons detected in 8 AGeV Ni+Ni collisions (simulation). The blue line represents the fit of the data. It is the sum of a 3rd degree polynomial function (dashed line) describing the combinatorial background and a Gaussian function (dark blue line) describing the peak resulting from the π^0 decay.

mass $M_{\gamma\gamma}$. In general the positions of the peaks in IMS correspond to the masses of the initial mesons. In Fig. 3.1 there is only one observable peak and it represents the decay of neutral pions. The exact position of the peak is determined by fitting IMS with the sum (blue line) of two functions: a polynomial function describing the background (dashed line) and a Gaussian function describing the peak itself (dark blue line). The observed background is of combinatorial origin: if an event contains more than two

photons it is impossible to determine which pair of photons originates from a given meson and all possible photon pairs should be taken into account. This creates a lot of fake pairs, resulting in this large combinatorial background.

3.2 Available software

The EMC data analysis software allows for the reconstruction of the initial incident photons. This reconstruction procedure should be able to correctly recreate the shower cascade and transport of Cherenkov photons inside a calorimeter block. Due to the complexity and randomness of these processes it is impossible to reconstruct the initial photons with a high precision. Some statistical uncertainties are inevitable but more worryingly, some systematic errors will also occur. These errors result mostly from unaccounted energy losses inside the blocks.

The second software package available for the calorimeter is a simulation software. It generates a known decay of a meson (e.g. π^0, η) into a pair of photons and simulates the interaction of those photons with the calorimeter, i.e. the electromagnetic shower development inside a block, the transport of Cherenkov photons and their detection with the PMTs. This simulation package is essential when testing a new software as it can quickly provide output data for any known physical process, without the need of a real experiment.

The simultaneous use of the two procedures (simulation and reconstruction) makes it possible to analyse any inaccuracies in the reconstruction procedure. It can be tested by reconstructing simulated data and comparing the reconstructed and initially simulated values.

Further details on the topic can be found in chapter VI. of [6].

3.3 Purposes of the calibration

The shower development and transport of Cherenkov photons inside matter are extremely difficult processes to be traced back. As discussed above, energy losses and signal alteration due to the geometry of the calorimeter are impossible to be perfectly reproduced. Thus the reconstruction procedure cannot be completely accurate. As described in section 3.1 Fig. 3.1 presents IMS for photons produced in a simulated Ni+Ni

collisions at 8 AGeV and the position of the observed peak is expected to correspond to the neutral pion mass $m_{\pi^0} \approx 135$ MeV/c^2 [10]. The actual position of the peak is calculated to be ≈ 119 MeV/c^2: the reconstruction procedure delivers a systematic underestimation of the π^0-mass. Those considerations clearly emphasize the necessity for a calibration procedure capable to compensate for the systematic errors in the reconstruction procedure.

Chapter 4

Calibration procedure

The calibration procedure is based on a simplified model that calibrates only the energies of single photons [11]. The mass of the neutral pion is a known physical quantity which is used by the calibration procedure as a reference. The procedure itself is implemented as a stand-alone program called IMSexpert. It is written in C++ and employs only standard C++ and ROOT [12] libraries. Thus, IMSexpert can be used as a calibration tool for any other electromagnetic calorimeter. The input variables of the program are the reconstructed momenta of the detected photons and, optionally, a few parameters related to the geometry of the calorimeter (see the IMSexpert User's Guide in the appendix A). IMSexpert is able to create and fit IMS as well as to calibrate the energies of the photons, using the mathematical model described below.

4.1 Overview of the calibration procedure

The calibration procedure should be applied to individual photons and must have the following general form:

$$\vec{p}_C = M\vec{p}, \tag{4.1}$$

where $\vec{p}$ is the originally reconstructed momentum of the photon, $\vec{p}_C$ is its momentum after calibration and M denotes a calibration matrix.

Throughout the whole text the momentum $\vec{p}$ will be considered in spherical coordinates (E, θ, φ). By performing reconstruction on simulated data and comparing the reconstructed (E, θ, φ) with the initially simulated (E_S, θ_S, φ_S) values it is possible to analyse the error in the reconstruction procedure. Figures 4.1 - 4.3 show the distribution

of the ratios E_S/E, θ_S/θ and φ_S/φ. These ratios represent the deviation of E, θ and φ from their true values E_S, θ_S and φ_S. As can be seen in Fig. 4.1 - 4.3 the average

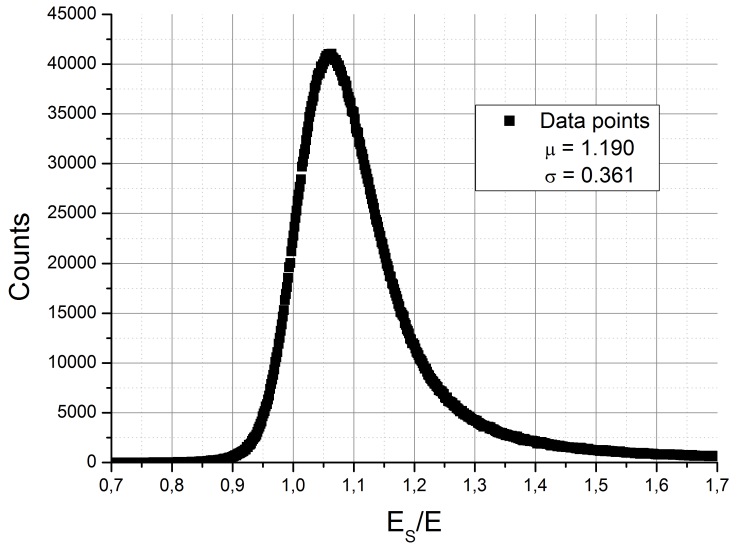

FIGURE 4.1: Distribution of E_S/E. μ is the mean value of E_S/E and σ is the standard deviation of E_S/E.

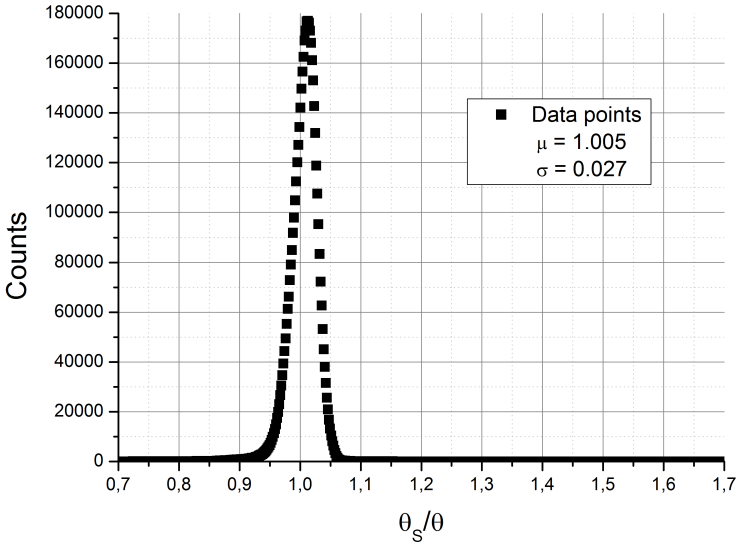

FIGURE 4.2: Distribution of θ_S/θ. μ is the mean value of θ_S/θ and σ is the standard deviation of θ_S/θ.

systematic shift of the energy E is approximately 19%, whereas the average systematic shifts of θ and φ are significantly less ($< 1.2\%$). Therefore, it is reasonable to assume that the improvement in the accuracy will be sufficient, if only the energy is calibrated.

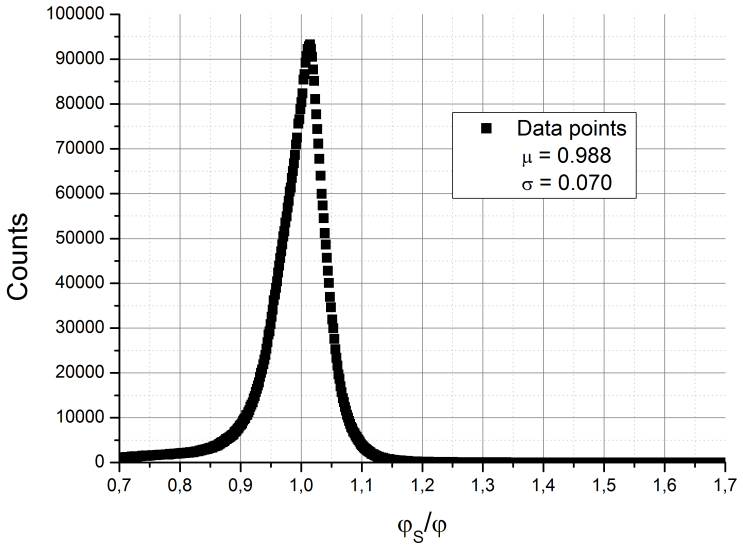

FIGURE 4.3: Distribution of φ_S/φ. μ is the mean value of φ_S/φ and σ is the standard deviation of φ_S/φ.

Equation (4.1) can thus be simplified to:

$$E_C = Ef\left(E, \theta, \varphi\right), \tag{4.2}$$

$$f\left(E, \theta, \varphi\right) = \frac{E_C}{E}, \tag{4.3}$$

where E_C and E are the calibrated and reconstructed energies, respectively, and f is a calibration function, which should in general depend on all components of the momentum (i.e. E, θ and φ).

The function used for calibration is given by equation (4.3). The rest of this chapter focuses on the exact methods to find out the explicit form of the calibration function f and to calculate the corresponding parameters.

4.2 Mathematical model

The next important step is to choose a suitable calibration function f. Since the exponential function is easy to be mathematically handled, a convenient general form of f is:

$$f(E, \theta, \varphi) = exp(\sum_i A_i F_i). \tag{4.4}$$

$F_i(E, \theta, \varphi)$ are expressions depending on the momentum components and A_i are fixed parameters called calibration coefficients [11]. For a given set of expressions F_i IMSexpert should be able to calculate the calibration coefficients A_i and thus perform the calibration.

Each calibration procedure needs a reference point. A well known physical quantity is the mass of the neutral pion $m_{\pi^0} \approx 135$ MeV/c^2. The position of the IMS peak M_{π^0} for simulated pions corresponds to the neutral pion mass. As can be clearly seen in Fig. 3.1, without calibration, the peak position M_{π^0} is shifted towards lower masses. In the ideal case the position of the peak is exactly at m_{π^0}. Therefore a likelihood function $\mathcal{L}$ can be defined as follows [11]:

$$\mathcal{L} = \sum_{j=1}^{N} [lnM_{\gamma\gamma j} - lnm_{\pi^0}]^2. \tag{4.5}$$

The summation is over all photon pairs in the sample, $M_{\gamma\gamma j}$ is the invariant mass of the calibrated j-th photon pair and m_{π^0} is the expected invariant mass, which is the mass of π^0. If the summation is performed only over photon pairs with invariant masses $M_{\gamma\gamma}$ in the range $M_{\pi^0} \pm \sigma_{M_{\pi^0}}$, then the likelihood function $\mathcal{L}$ should converge towards zero. Thus the calibration coefficients A_i can be determined by minimizing $\mathcal{L}$ with respect to A_i. Using the following designations:

$$A_i : \text{Calibration coefficients},$$

$$F_i : \text{The expressions after } A_i,$$

$$\lambda_i := \tfrac{1}{2}(F_{1i} + F_{2i}) : \text{valid for a pair of photons},$$

$$\rho := lnM_{\gamma\gamma} - lnm_{\pi^0} = ln\left(\tfrac{M_{\gamma\gamma}}{m_{\pi^0}}\right),$$

the relations involved are:

$$\mathcal{L} = \sum_{j=1}^{N} [lnM_{\gamma\gamma j} - lnm_{\pi^0}]^2 = \sum_{j=1}^{N} \rho_j^2, \tag{4.6}$$

$$R_i := \frac{1}{2}\frac{\partial \mathcal{L}}{\partial A_i} = \sum_{j=1}^{N} \rho_j \lambda_{ij}, \tag{4.7}$$

$$G_{il} := \frac{\partial R_i}{\partial A_l} = \frac{1}{2}\frac{\partial^2 \mathcal{L}}{\partial A_i \partial A_l} = \sum_{j=1}^{N} \lambda_{lj}\lambda_{ij}. \tag{4.8}$$

The detailed derivation of the above equations is given in the appendix B.

The likelihood function $\mathcal{L}$ has to be minimized, therefore R_i should be equal to zero. If $R_i \neq 0$ a small correction δR_i should be applied, such that $R_i + \delta R_i = 0$. If δR_i is small enough:

$$\frac{\partial R_i}{\partial A_l} \approx \frac{\delta R_i}{\delta A_l}, \tag{4.9}$$

$$\delta R_i = \frac{\partial R_i}{\partial A_l} \delta A_l = G_{il} \delta A_l, \tag{4.10}$$

$$R_i + \delta R_i = R_i + G_{il} \delta A_l = 0, \tag{4.11}$$

$$R_i = -G_{il} \delta A_l, \tag{4.12}$$

$$\delta A_l = -G_{il}^{-1} R_i. \tag{4.13}$$

Equation (4.13) is a matrix equation. G_{il} and R_i depend only on the parameters ρ and λ_i, which are fully determined by the input data and the calibration function. Thus by calculating ρ_j and λ_{ij} for each photon pair and using equation (4.13) the correction coefficients δA_i can be determined. The initial values of A_i are typically set to zero ($f = 1$) and using the described method A_i are corrected by some amount δA_i. As the approximation (4.9) is not always valid, the calibration procedure should iteratively be repeated with the new values of A_i until condition (4.11) is fulfilled.

4.3 Choice of the calibration function

In a simulation or an experiment the calibrated energy E_C should ideally correspond to the original (simulated or experimental) value of the energy E_S. By analysing the ratio $f_0 := E_S/E$, where E is the reconstructed energy, the expected form of $f(E, \theta, \varphi) = E_C/E$ can be determined.

Figure 4.4 represents the average value of f_0 plotted against the E and the angle θ. As can be seen in this figure, the dependence of f_0 on θ at fixed E is approximately the

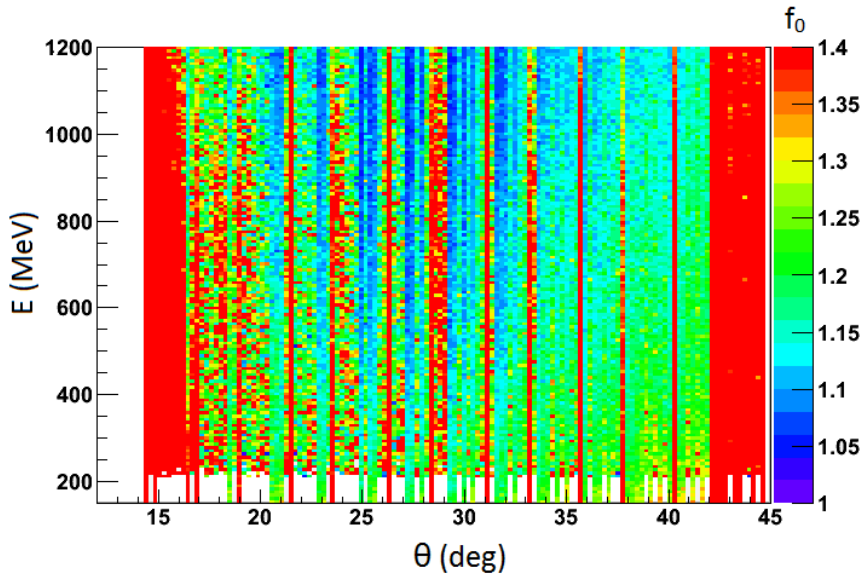

FIGURE 4.4: Values of f_0 depending on E and θ according to reconstruction of simulated data.

same for each E and vice versa. The same considerations can be done for the dependence of the average value of f_0 on E and φ (see Fig. 4.5). Thus the energy dependence of

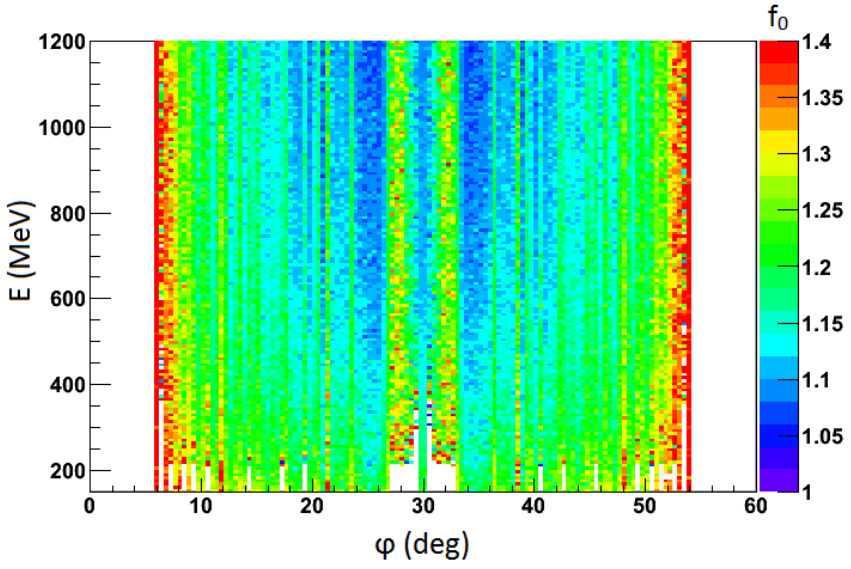

FIGURE 4.5: Values of f_0 depending on E and φ according to reconstruction of simulated data.

f_0 can be analysed separately from the θ and φ dependencies. However, as it will be explained in subsection 4.3.2, the dependencies of f_0 on θ and φ are strongly correlated

(see Fig. 4.7). Thus the dependencies of f_0 on θ and φ cannot be analysed separately for the two variables.

4.3.1 Dependency on the energy

Figure 4.6 shows the dependency of f_0 on the reconstructed energy E averaged over the angles θ and φ.

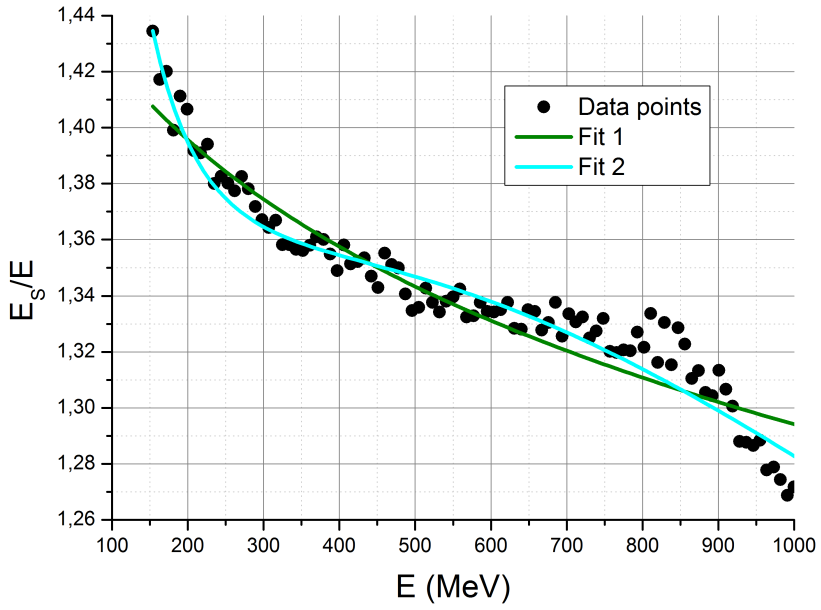

FIGURE 4.6: f_0 according to reconstruction of simulated data. The fits are given by equations (4.14) and (4.15). The value of χ^2/NDF$\approx$ 5.34 for Fit 1 and $\approx$ 2.96 for Fit 2.

The shown fits have been done with the functions

$$E_{fit1} = exp(P_0 + P_1 lnE + P_2 ln^2 E), \tag{4.14}$$

$$E_{fit2} = exp(P_0 + P_1 lnE + P_2 ln^2 E + P_3 ln^3 E), \tag{4.15}$$

where P_i are the parameters of the fit.

4.3.2 Dependency on the polar and azimuthal angles

Fig. 4.7 presents the average value of f_0 depending on θ and φ. The periodicity along both axes results from the periodic arrangement of blocks inside each sector of the calorimeter.

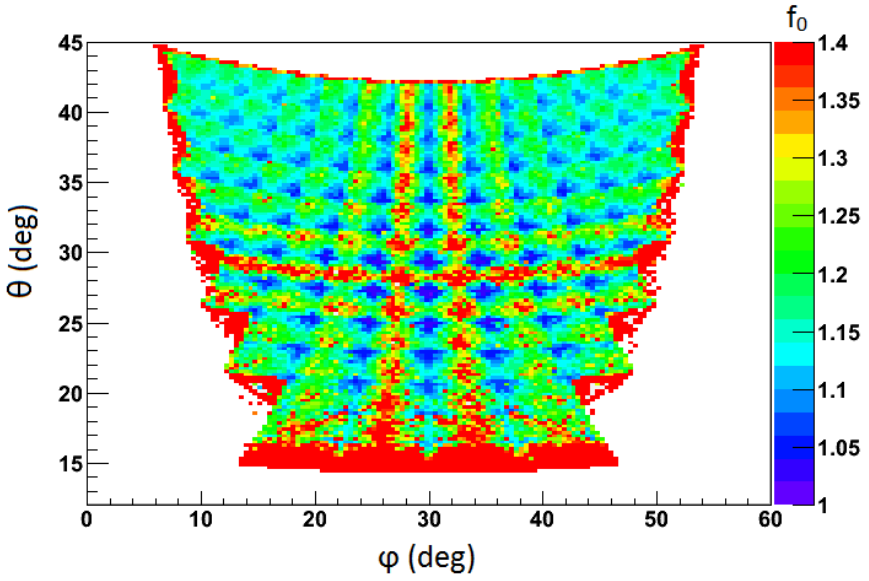

FIGURE 4.7: Values of f_0 depending on θ and φ according to reconstruction of simulated data.

Each minima line (blue) along the θ axis represents the centre of each column of blocks (see Fig. 2.4), while each minima line along the φ axis reflects the centre of each row of blocks. Thus the crossing points of the minima lines along the θ axis and those along the φ axis correspond to the centres of the calorimeter blocks. Each sector is essentially flat, thus the distance between the calorimeter blocks is not equidistant in spherical coordinates, as can be seen in Fig. 4.7. The observed relation between θ and φ can be represented with the following function:

$$f_{\theta,\varphi} = exp\left[P_1 cos\left(\tilde{\theta}\omega_\theta cos\tilde{\varphi} - \theta_{shift}\right) + P_2 cos\left(\tilde{\varphi}\omega_\varphi sin(\theta) - \varphi_{shift}\right)\right], \tag{4.16}$$

with

φ_{min} : φ of the centre of the upper-left (see Fig. 2.4) block of the sector,

φ_{max} : φ of the centre of the upper-right block of the sector,

$$\varphi_m = (\varphi_{max} - \varphi_{min})/2,$$

θ_{min} : θ of the centre of the first row of blocks,

θ_{max} : θ of the centre of the last row of blocks,

$$\tilde{\theta} = \theta - \theta_{min},$$

$$\tilde{\varphi} = \varphi - \varphi_m,$$

$$\omega_\theta = (n_\theta - 1)\frac{360^\circ}{\theta_{max} - \theta_{min}},$$

n_θ : number of columns of blocks,

$$\omega_\varphi = (n_\varphi - 1)\frac{360^\circ}{\varphi_{max} - \varphi_{min}},$$
$$n_\varphi : \text{number of rows of blocks},$$
$$\theta_{shift} = 180^\circ,$$
$$\varphi_{shift} = 180^\circ \times \mathcal{P}(n_\varphi),$$
$$\mathcal{P}(n_\varphi) : \text{Parity of } n_\varphi.$$

The form of $f_{\theta,\varphi}(\theta, \varphi)$ is illustrated in Fig. 4.8. The parameters of the function are

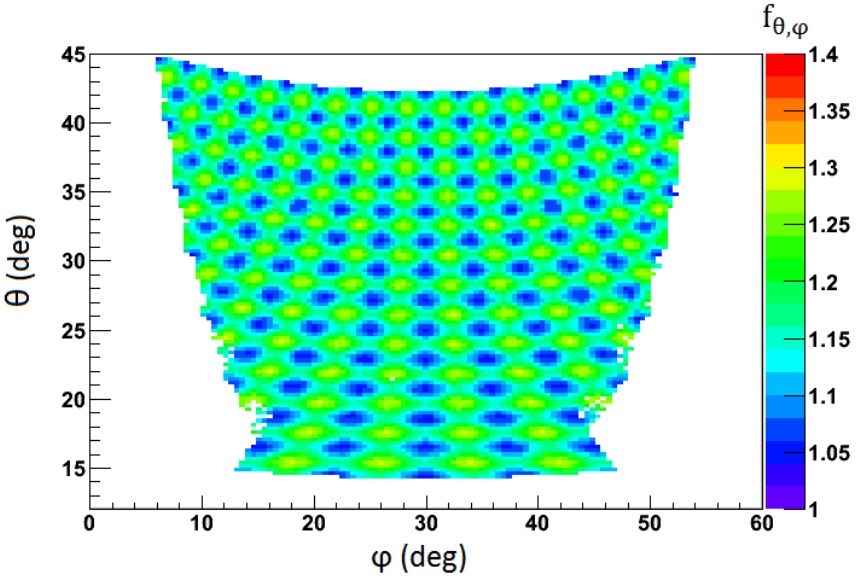

FIGURE 4.8: Form of $f_{\theta,\varphi}(\theta, \varphi)$.

chosen in such a way, so that the positions and values of the minima and maxima of f_0 and $f_{\theta,\varphi}$ coincide. Thus, if $f_{\theta,\varphi}$ can describe f_0 accurately, the ratio $f_{\theta,\varphi}/f_0$ should be approximately unity. As can be seen in Fig. 4.9 $f_{\theta,\varphi}/f_0$ is indeed close to unity for most values of θ and φ. There are still deviations from unity, up to 20%, but the general form of the function $f_{\theta,\varphi}$, i.e. the periodicity of both θ and φ as well as the shifts in the periodicity depending on θ and φ, represents the expected functional form of f_0 very accurately.

Hence, if equation (4.16) is included as a term in the calibration function, the dependency of the inaccuracy associated with both spatial angles is significantly reduced, thus allowing for a better estimation of the energy dependence of the calibration function.

4.3.3 Accuracy

There are many factors that contribute to the choice of the calibration function. The form of the function should be chosen carefully by finding the balance between

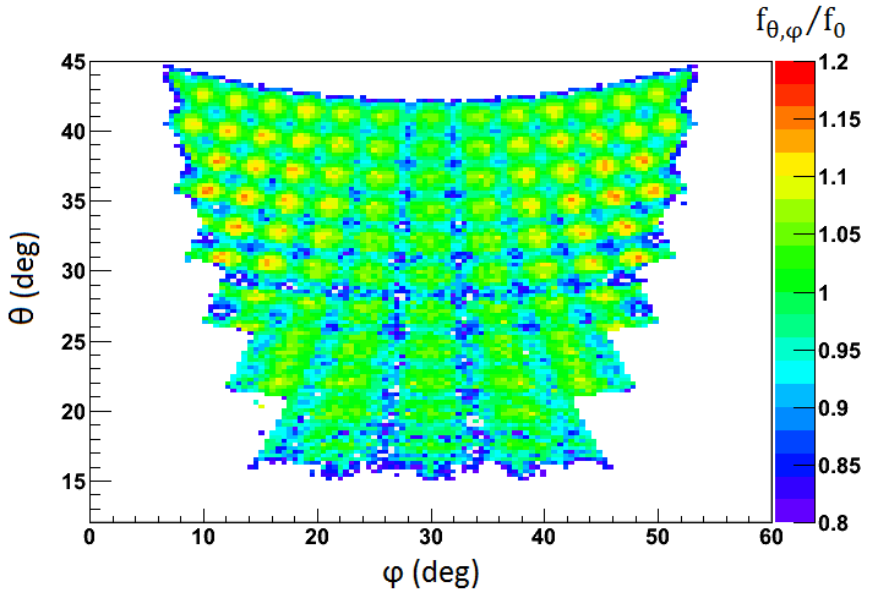

FIGURE 4.9: Ratio between $f_{\theta,\varphi}$ and f_0.

computational time and accuracy. During the development of IMSexpert many tests were performed in order to find this balance. It was determined that the most important term in the calibration function is the logarithmic dependency on the energy. Fig. 4.10

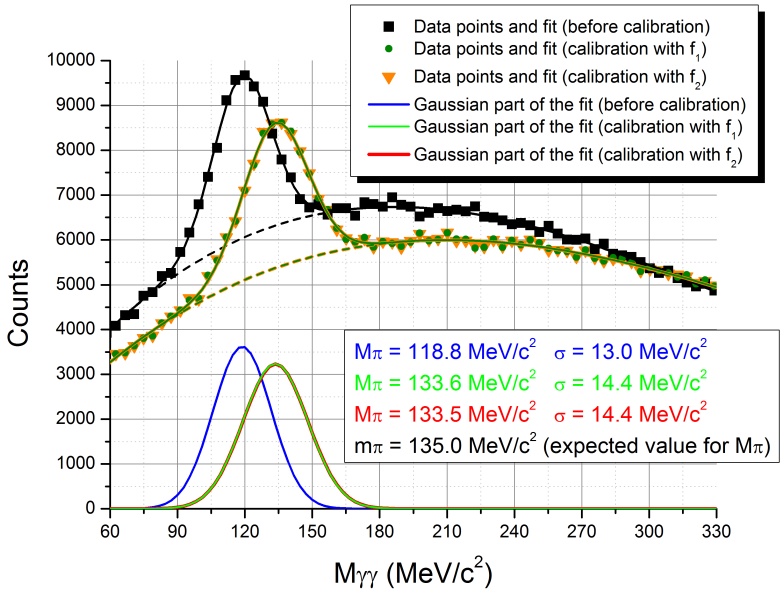

FIGURE 4.10: IMS before and after calibration. The data for IMS are from simulated Ni+Ni collisions at 8 AGeV.

shows the invariant mass spectrum after calibration with the function

$$f_1(E) = exp(A_0 + A_1 lnE + A_2 ln^2 E). \tag{4.17}$$

A function that is a little bit more accurate than f_1 is:

$$f_2(E,\theta,\varphi) = exp(A_0 + A_1 lnE + A_2 ln^2 E + A_3 ln^3 E + \\ + A_4 cos\left[\tilde{\theta}\omega_\theta cos\tilde{\varphi} - \theta_{shift}\right] + A_5 cos\left[\tilde{\varphi}\omega_\varphi sin(\theta) - \varphi_{shift}\right]). \tag{4.18}$$

Compared to f_1, f_2 has a higher order of logarithmic dependency on the energy and equation 4.16 (see 4.3.2) is used to decrease the dependency of the inaccuracy on the angles θ and φ. The designations in equation (4.18) are the same as in equation (4.16).

The results from the calibrations with different functions are presented in Fig. 4.11 - 4.13. Fig. 4.11 shows the exact form of the calibration functions f_1, f_2 and $f_3(E) = exp(A_0 + A_1 lnE + A_2 ln^2 E + A_3 ln^3 E)$.

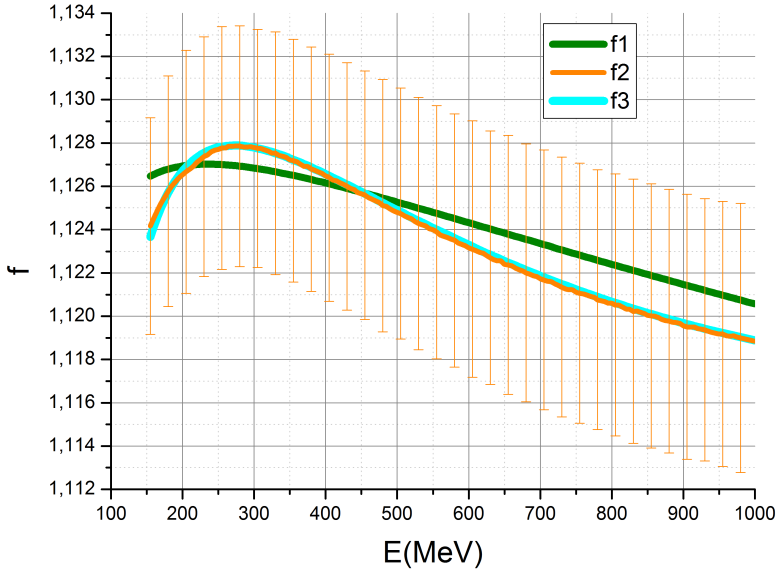

FIGURE 4.11: Calibration function f as a function of the energy. f_2 depends on the angles θ and φ. The solid line represents the average values of f_2 and the error bars represent the fluctuations of f_2 at fixed E due to its θ and φ dependency.

Those data were obtained by reconstructing and calibrating known simulated data. As can be seen, the accuracy of the presented calibrations is almost identical. In fact, multiple calibration functions with different θ and φ dependencies were tested and all of them delivered IM spectra extremely similar to those with either f_1 or f_2. The difference in the overall shape comes only from the order of the logarithmic term. To confirm this statement the calibration function

$$f_3(E) = exp(A_0 + A_1 lnE + A_2 ln^2 E + A_3 ln^3 E) \tag{4.19}$$

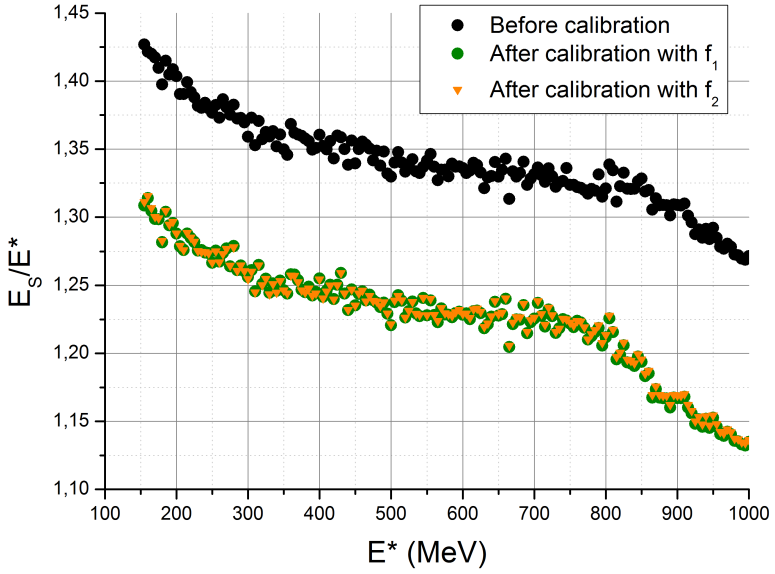

FIGURE 4.12: Error in the energy of single photons before and after calibration. For the data before calibration E^* is the reconstructed energy E. For the data after calibration E^* is the calibrated energy E_C.

has been shown for a comparison in Fig. 4.11.

The reference point for the calibration is the mass of the neutral pion $m_{\pi^0} \approx 135\ \mathrm{MeV}/c^2$. As explained in chapter 4.2 the position of the peak M_{π^0} in IMS for simulated pions should be equal to m_{π^0}. Fig. 4.10 shows IMS before and after calibration with f_1. The position of the peak M_{π^0} is clearly shifted to the correct position (i.e. $M_{\pi^0} \approx m_{\pi^0} \approx 135\ \mathrm{MeV}/c^2$) after calibration. The standard deviation of the peak stays approximately the same, changing from 13 MeV/c^2 to 14 MeV/c^2. This means that the systematic errors from the reconstruction procedure are compensated by the calibration, but the statistical error remains approximately the same. This can be expected because the statistical error is mostly related to the energy resolution of the calorimeter, which is $\sigma_E/E = 6\%/\sqrt{E/\mathrm{GeV}}$, or approximately 20 MeV for energies at m_{π^0}. By making a new simulation that includes not only neutral pions but also η mesons and looking at the position of the peak M_η corresponding to the mass of the η meson $m_\eta \approx 548\ \mathrm{MeV}/c^2$ [10], one can draw conclusions about the accuracy of the calibration at energies comparable to m_η. Fig. 4.14 shows the position of M_η after calibration. Its value is $\approx 561\ \mathrm{MeV}/c^2$, which deviates only by $\approx 2.4\%$ from the expected value $m_\eta \approx 548\ \mathrm{MeV}/c^2$. This deviation can be considered as the error in M_η related to the calibration procedure. Since the calibration error is less than the experimental energy resolution, which for energies

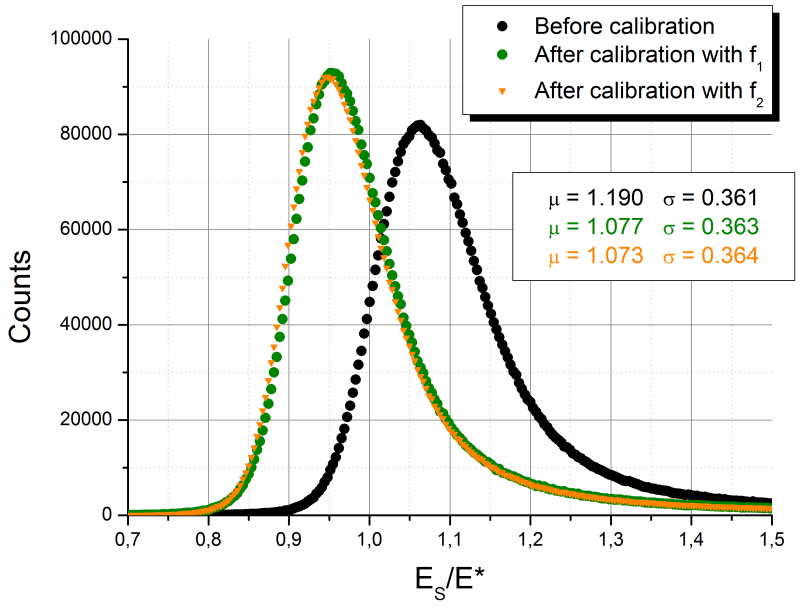

FIGURE 4.13: Distributions of the errors in the energy before and after calibration. For the data before calibration E^* is the reconstructed energy E. For the data after calibration E^* is the calibrated energy E_C.

at 548 MeV is approximately 8%, the calibration is considered as accurate.

Therefore, the task of this thesis to create a calibration procedure capable of reconstructing the invariant mass spectrum correctly for invariant masses between $m_{\pi^0} \approx 135$ MeV/c^2 and $m_\eta \approx 548$ MeV/c^2 has been achieved. However as shown in Fig. 4.12, despite the decrease in the errors of the energies of single photons, they still remain significant.

4.4 Implementation in C++

The described calibration procedure has been implemented in a computer program called IMSexpert. The source code uses only the standard C++ and ROOT [12] libraries. All functions of the program are realized as one class and several functions saved in the files IMSexpert.cc, IMSexpert.h, IMShelp.cc and IMShelp.h. The input data are taken from a file. They need to be contained in a standard ROOT TTree object. The j-th entry of the TTree should contain the spherical coordinates of the momenta of the photons in the j-th pair.

The program is able to fit the invariant mass spectrum and, using the mathematical method described in chapter 4.2, can find the appropriate calibration coefficients A_i.

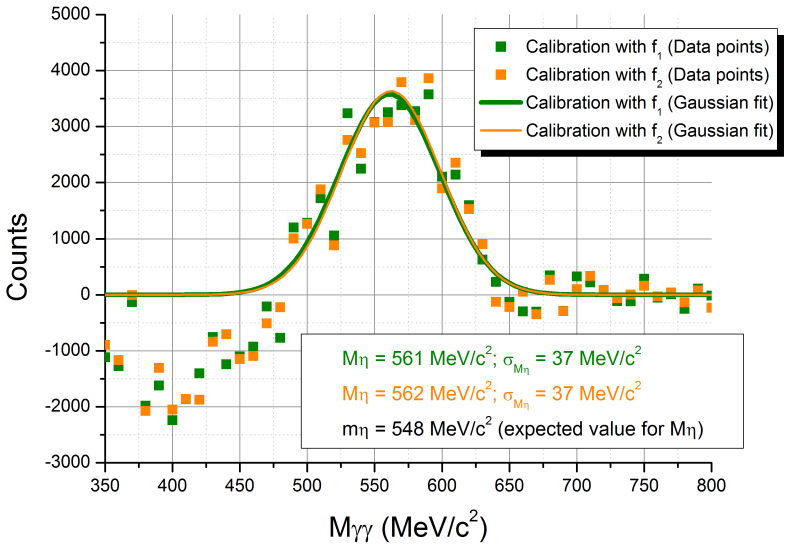

FIGURE 4.14: IMS before and after calibration (η mesons). The data for IMS are from simulated Ni+Ni collisions at 8 AGeV. The background has been removed using the mixed event technique. Since the absolute level of the background is about two orders of magnitude stronger than the height of the actual peak, after the subtraction of the background there are still remaining background fluctuations in the vicinity of the peak, which may also be negative. Apparently this is physically impossible and thus the baseline of the peak is set to be zero.

The calibrated values of the energies are saved in a new file, as a TTree object. Full information about the calibration procedure, including the coefficients A_i is saved in a binary file. For further calibrations of the same type IMSexpert can use the calibration coefficients saved in the binary file, thus eliminating the necessity to repeat the calculations. IMSexpert uses equation (4.17) as the default calibration function. The program, however, can operate with different calibration functions, including some that depend on θ and φ (e.g. (4.18) and (4.19)). All possible calibration functions as well as the additional input parameters that they may require are described in detail in the User's guide of IMSexpert (appendix A).

Chapter 5

Conclusion

5.1 Summary

In order to fulfil the requirements for data analysis of collisions at energies between 2-8 AGeV, a calibration procedure applicable for the new EMC detector at HADES is proposed. The calibration is done on individual photons and is represented by the equation:

$$E_C = E f\,(E, \theta, \varphi),$$

where E_C is the calibrated value of the energy, E is the reconstructed value of the energy and $f(E, \theta, \varphi)$ is the calibration function [11]. By analysing data that have been reconstructed from known simulated events, the general form of f can be obtained. Two forms of the calibration function are shown to be especially useful:

$$f_1(E) = exp(A_0 + A_1 lnE + A_2 ln^2 E), \tag{5.1}$$

and

$$\begin{aligned} f_2(E, \theta, \varphi) = exp(&A_0 + A_1 lnE + A_2 ln^2 E + A_3 ln^3 E + \\ &+ A_4 cos\left[\tilde{\theta}\omega_\theta cos\tilde{\varphi} - \theta_{shift}\right] + A_5 cos\left[\tilde{\varphi}\omega_\varphi sin(\theta) - \varphi_{shift}\right]). \end{aligned} \tag{5.2}$$

In the above equations A_i are called calibration coefficients and are the parameters that have to be fitted during the calibration procedure. The designations in equation (5.2) are the same as in equation (4.16).

Despite its simplicity, equation (5.1) is accurate enough for the main purposes (π^0 and η reconstruction) of EMC at HADES. It is a function easy to work with and thus the calibration procedure requires very little computational time.

The general improvement in the accuracy by calibrating with f_2 instead of f_1 is negligibly small. However the terms describing the dependency on θ and φ considerably decrease the fluctuations of the inaccuracy that depend on those angles. Hence the remaining errors are significantly less dependent on θ and φ.

The calibration procedure is implemented in a standalone C++ program called IMSexpert. The user can chose from various calibration functions, apply cuts and make simple fits of IMS.

5.2 Outlook

IMSexpert is developed for use at the proposed EMC at HADES. For the main purposes of this detector [6], the calibration procedure presented in this work is accurate enough. IMSexpert can be used for any other calorimeter, but depending on the energy range, resolution, etc., the accuracy might be insufficient and further improvements of the calibration procedure will be required. As discussed in chapter 4.3.3, the energy calibration cannot be substantially improved. Thus for further overall improvement of the calibration not only the energy but all components of the momentum vector of each photon need to be calibrated (see equation (4.1)). Finding a suitable calibration matrix would be a very difficult task. A simpler approach is to perform the momentum calibration using the following system of equations:

$$\left|\begin{array}{l} E_C = E f_E(E,\theta,\varphi) \\ \theta_C = \theta f_\theta(E,\theta,\varphi) \\ \varphi_C = \varphi f_\varphi(E,\theta,\varphi), \end{array}\right. \tag{5.3}$$

where f_E, f_θ and f_φ are the calibration functions for E, θ and φ respectively. The system of equations (5.3) can be solved using again the method based on the likelihood function $\mathcal{L}$ (see chapter 4.2). In this case the derivatives of $\mathcal{L}$ will be much more complex, which would eventually lead to different expressions for R_i and G_{il}. An important point is that R_i and G_{il} will be once again possible to be calculated only from the initial photon momenta, while the calibration coefficients A_i can be found using equation (4.13). A

lot more coefficients A_i will be needed though, which will lead to a significant increase in the calculation time. If, however, the energy calibration is not good enough for a specific calorimeter or very accurate measurements of individual photons are needed, such a method would be extremely useful to be developed.

For the purposes of less computational time the method described above can be further simplified by creating look-up tables for the calibration of θ and φ. The following discussion is identical for both θ and φ, thus without loss of generality the discussion will be only about θ. The look-up table for the calibration of θ can be represented by a three-dimensional array. Each element of the array will correspond to the value of f_θ for specific energy E, polar angle θ and azimuthal angle φ. Ideally, the calibrated values θ_C should be equal to the simulated values θ_S. Thus, by simulating HI collisions and then reconstructing the detected photons, the expected values of $f_\theta(E, \theta, \varphi)$ can be calculated for some discrete values of E, θ and φ using equation (5.3) and $\theta_C = \theta_S$. The look-up table is created by saving the values of f_θ in the array elements corresponding to E, θ and φ.

The calibration described by equation (5.3) can be performed by firstly calibrating the angles θ and φ with the two look-up tables and then calibrating the energy E with IMSexpert. It should be, however, noted that the look-up tables are individual for each detector and strongly depend on the accuracy of the simulation procedure.

Appendix A

IMSexpert User's Guide

This section explains in short how to use the program IMSexpert. Please do note that everything described in this section is valid at the time of submitting this bachelor thesis (July 2012). If you are using IMSexpert be sure to find the full and updated user's guide!

IMSexpert is implemented in a single class called IMSexpert and a few additional functions. The header file of the class is IMSexpert.h, the header file for the additional functions is IMShelp.h. Below are listed and explained all public function members (methods) of the class IMSexpert.

A.1 Data import and export

Bool_t ImportData(TTree * TempTree);

It imports the experimental data from a TTree object [12]. The TTree object should contain the following TBranches:

Default name	**Meaning**
E1	The energy of the 1st photon (in a photon pair)
E2	The energy of the 2nd photon (in a photon pair)
theta1	The polar angle θ of the 1st photon (in a photon pair)
theta2	The polar angle θ of the 2nd photon (in a photon pair)
phi1	The azimuthal angle φ of the 1st photon (in a photon pair)
phi2	The azimuthal angle φ of the 2nd photon (in a photon pair)
Mgg	The IM of the photon pair (optional)
eta	Angle between the two photons (optional)

Since the value of the IM and the angle between two photons can be calculated from their energies, θ and φ angles, if Mgg and eta are not provided, IMSexpert calculates them. The function returns the import status, "true" if successful.

Bool_t ImportData(const Char_t * FileName);

It is the same function as Bool_t ImportData(TTree * TempTree) but the TTree object is saved in a file. The default name for the TTree object is tree_Mgg. The function returns the import status, "true" if successful.

Bool_t ExportData(TTree *TR);

It saves the experimental data in a TTree object. The function returns the export status, "true" if successful.

Bool_t ExportData(Char_t * FileName);

It saves the experimental data in a ROOT file. The file will contain a TTree object. The function returns the export status, "true" if successful.

A.2 Names

void DefaultNames();

It sets the names of the TTree object containing the experimental data and its TBranches

to a predefined default names (see Bool_t ImportData(TTree * TempTree) and ImportData(const Char_t * FileName)).

void SetNameEnergy1(const Char_t * Name);
It sets the name of the TBranch of the TTree object containing the experimental values of the energy of the 1st photon.

void SetNameEnergy2(const Char_t * Name);
It sets the name of the TBranch of the TTree object containing the experimental values of the energy of the 2nd photon.

void SetNameTheta1(const Char_t * Name);
It sets the name of the TBranch of the TTree object containing the experimental values of the θ angle of the 1st photon.

void SetNameTheta2(const Char_t * Name);
It sets the name of the TBranch of the TTree object containing the experimental values of the θ angle of the 2nd photon.

void SetNamePhi1(const Char_t * Name);
It sets the name of the TBranch of the TTree object containing the experimental values of the φ angle of the 1st photon.

void SetNamePhi2(const Char_t * Name);
It sets the name of the TBranch of the TTree object containing the experimental values of the φ angle of the 2nd photon.

void SetNameEta(const Char_t * Name);
It sets the name of the TBranch of the TTree object containing the values of the angle between the two photons.

void SetNameMgg(const Char_t * Name);

It sets the name of the TBranch of the TTree object containing the values of the IM of the two photons.

void SetNameTree(const Char_t * Name);

It sets the name of the TTree object containing the experimental data.

A.3 Errors

void SetErrors(Float_t dEnergy, Float_t dTheta, Float_t dPhi);

It sets the energy, θ and φ resolution. For the energy, the resolution is calculated by σ_E/E =dEnergy$/\sqrt{E/\mathrm{GeV}}$.

A.4 Fitting procedure, cuts

void SetRange(Float_t *R);

It sets the range [R[0]; R[1]] in which the fitting procedure will be executed.

void SetRange(Float_t R1, Float_t R2);

It sets the range [R1; R2] in which the fitting procedure will be executed.

void GetRange(Float_t *R);

It reads the range [R[0]; R[1]] in which the fitting procedure will be executed.

void SetIMSCut(Float_t Mgg_min, Float_t Mgg_max);

It applies cut on the IM, i.e. IMSexpert doesn't take into account all photon pairs with IM smaller than Mgg_min and larger than Mgg_max.

void SetIMSCut(Float_t *Mgg_cut);

It applies cut on the IM, i.e. IMSexpert doesn't take into account all photon pairs with

IM smaller than Mgg_cut[0] and larger than Mgg_cut[1].

void GetIMSCut(Float_t *Mgg_cut);

It reads the applied IM cut and saves it to Float_t *Mgg_cut.

void SetEnergyCut(Float_t E_min, Float_t E_max);

It applies cut on the energy, i.e. IMSexpert doesn't take into account all photons with energy smaller than E_min and larger than E_max.

void SetEnergyCut(Float_t *E_cut);

It applies cut on the energy, i.e. IMSexpert doesn't take into account all photons with energy smaller than E_cut[0] and larger than E_cut[1].

void GetEnergyCut(Float_t *E_cut);

It reads the applied energy cut and saves it to Float_t *E_cut.

void SetThetaCut(Float_t theta_min, Float_t theta_max);

It applies cut on the polar angle θ, i.e. IMSexpert doesn't take into account all photons with θ smaller than theta_min and larger than theta_max.

void SetThetaCut(Float_t *theta_cut);

It applies cut on the polar angle θ, i.e. IMSexpert doesn't take into account all photons with θ smaller than theta_cut[0] and larger than theta_cut[1].

void GetThetaCut(Float_t *theta_cut);

It reads the applied polar angle cut and saves it to Float_t *theta_cut.

void SetPhiCut(Float_t phi_min, Float_t phi_max);

It applies cut on the azimuthal angle φ, i.e. IMSexpert doesn't take into account all photons with φ smaller than phi_min and larger than phi_max.

void SetPhiCut(Float_t *phi_cut);

It applies cut on the azimuthal angle φ, i.e. IMSexpert doesn't take into account all photons with φ smaller than phi_cut[0] and larger than phi_cut[1].

void GetPhiCut(Float_t *phi_cut);

It reads the applied azimuthal angle cut and saves it to Float_t *phi_cut.

void IMS_Fit(Char_t * OutputFile, Float_t Peak, Short_t n_of_par);

It creates and fits IMS of the experimental data. Saves the result in a *.root file. IMS_Fit will search for a peak in the vicinity of Float_t Peak. The order of the polynomial fit of the background is n_of_par.

void IMS_Fit(Char_t * OutputFile, Float_t Peak);

It creates and fits IMS of the experimental data. Saves the result in a *.root file. IMS_Fit will search for a peak in the vicinity of Float_t Peak. The order of the polynomial fit of the background is 3.

void IMS_Fit(TH1F *histo_Mgg, Char_t * OutputFile, Float_t Peak, Short_t n_of_par);

It creates and fits IMS of the data saved in histo_Mgg. Saves the result in a *.root file. IMS_Fit will search for a peak in the vicinity of Float_t Peak. The order of the polynomial fit of the background is n_of_par.

void IMS_Fit(TH1F *histo_Mgg, Char_t * OutputFile, Float_t Peak);

It creates and fits IMS of the data saved in histo_Mgg. Saves the result in a *.root file. IMS_Fit will search for a peak in the vicinity of Float_t Peak. The order of the polynomial fit of the background is 3.

void IMS_Fit(Float_t **DataArray, Double_t *PAR, Short_t n_of_par, Float_t Peak);

It creates and fits IMS of the data saved in Float_t **DataArray. The parameters of the fit are saved in Double_t *PAR. IMS_Fit will search for a peak in the vicinity of Float_t

Peak. The order of the polynomial fit of the background is n_of_par.

Float_t **DataArray has the following form:

DataArray[j][0] → IM of the j-th photon pair.

DataArray[j][1] → Energy of the 1st photon from the j-th pair,

DataArray[j][2] → Energy of the 2nd photon from the j-th pair,

DataArray[j][3] → θ of the 1st photon from the j-th pair,

DataArray[j][4] → θ of the 2nd photon from the j-th pair,

DataArray[j][5] → φ of the 1st photon from the j-th pair,

DataArray[j][6] → φ of the 2nd photon from the j-th pair.

The fit parameters are saved in the following way:

$$f_{fit}(M_{\gamma\gamma}) = \mathrm{PAR}[0] * exp\left[\frac{1}{2}\left(\frac{M_{\gamma\gamma} - \mathrm{PAR}[1]}{\mathrm{PAR}[2]}\right)^2\right] + \sum_{i=0}^{\mathrm{PAR}[3]} \mathrm{PAR}[4{+}\mathrm{i}] * M_{\gamma\gamma}^{i}, \qquad \text{(A.1)}$$

where PAR[3] == n_of_par.

A.5 Calibration

Bool_t Calibrate(Float_t m0, Char_t * OutputFile, Char_t * OutputFile2);
It performs the calibration procedure on the experimental data, using m0 as a reference point (see chapter 4.2). The calibration function is given according to equation 4.17.

The calibrated values are saved in the file OutputFile as a TTree object. The calibration coefficients are saved in a binary form in the file OutputFile2. This file can be used for quick calibration with the same function and coefficients. Additionally one *.txt file, with the same name as OutputFile2, is created and the details about the calibration procedure are saved as text.

Bool_t Calibrate(Float_t m0, Short_t NAE, Short_t NAt, Short_t NAp, Short_t NAEt, Char_t * OutputFile, Char_t * OutputFile2);
It performs the calibration procedure on the experimental data, using m0 as a reference point (see chapter 4.2). The calibration function has the following form:

$$f(E,\theta,\varphi) = exp[A_0 + \sum_{i=1}^{\mathrm{NAE}} A_i ln^i E + \\ + \sum_{i=1}^{\mathrm{NAt}} A_{(\mathrm{NAE}+i)} cos^i(k_2(\theta - \theta_{MIN})) + \sum_{i=1}^{\mathrm{NAp}} A_{(\mathrm{NAE}+\mathrm{NAt}+i)} cos^i\varphi + \\ + \mathrm{NAEt} \times A_{(\mathrm{NAE}+\mathrm{NAt}+\mathrm{NAp}+1)} lnE cos\theta], \tag{A.2}$$

The designations are explained below:

k_1 and k_2 set the correct periodicity. The parameters θ_{MIN}, k_1 and k_2 all depend on the geometry of EMC and should be given as input values by the user. If the angle θ varies between θ_{MIN} and θ_{MAX} and each sector has n_θ number of blocks along θ the following relations are valid:

$$k_1 = \frac{360^\circ}{\theta_{MAX} - \theta_{MIN}}, \tag{A.3}$$

$$k_2 = n_\theta \times \frac{360^\circ}{\theta_{MAX} - \theta_{MIN}}, \tag{A.4}$$

The files OutputFile and OutputFile2 are the same as in the function Bool_t Calibrate(Float_t m0, Char_t * OutputFile, Char_t * OutputFile2).

Bool_t Calibrate(Float_t m0, Char_t type, Char_t * OutputFile, Char_t * OutputFile2);

This function is very similar to the former described function. The only difference is, that through the variable Char_t type the user has the access to a number of preset calibration functions. Those functions (f_{type}) are described below:

$$f_{(\mathrm{type}=0)}(E,\theta,\varphi) = exp(A_0 + A_1 lnE + A_2 ln^2 E + A_3 ln^3 E + \\ + A_4 cos\left[\tilde{\theta}\omega_\theta cos\tilde{\varphi} - \theta_{shift}\right] + A_5 cos\left[\tilde{\varphi}\omega_\varphi sin(\theta) - \varphi_{shift}\right]). \tag{A.5}$$

For details about this function see equation (4.18).

$$f_{(\mathrm{type}=1)}(E,\theta,\varphi) = exp(A_0 + A_1 lnE + A_2 ln^2 E + A_3 ln^3 E + \\ + A_4 cos\left[\frac{k_2}{n_\theta}(\theta - \theta_{MIN})\right] + A_5 cos[k_2(\theta - \theta_{MIN})]). \tag{A.6}$$

$$f_{(\mathrm{type}=2)}(E,\theta,\varphi) = exp(A_0 + A_1 lnE + A_2 ln^2 E + A_3 ln^3 E + \\ + A_4 \left[\theta - \theta_{MIN} - \frac{\theta_{MAX} - \theta_{MIN}}{2}\right]^2 + A_5 cos[k_2(\theta - \theta_{MIN})]). \tag{A.7}$$

The designations in $f_{(\text{type}=1)}$ and $f_{(\text{type}=2)}$ are the same as in equation (A.2).

Bool_t CalibImport(const Char_t * FileName);

It loads the data obtained from a previous calibration procedure from the binary file FileName. FileName should have been created by IMSexpert automatically whilst performing the previous calibration.

Bool_t CalibExport(Char_t * FileName);

It creates the binary file for future calibrations manually.

Bool_t PresetCalib(Char_t * CalibFile, Char_t * OutputFile);

It applies the calibration using the parameters from a previous calibration. CalibFile is the binary file created by IMSexpert in the previous calibration and OutputFile is the file in which the new calibrated IM values will be saved.

Bool_t PresetCalib(Char_t * OutputFile);

It applies the calibration using the parameters from a previous calibration. OutputFile is the file in which the new calibrated IM values will be saved. Note, that the binary file from the previous calibration should already be loaded before using this function.

A.6 Detector parameters

void SetThetaRange(Float_t *R);

It sets the minimum (R[0]) and maximum (R[1]) value of θ.

void GetThetaRange(Float_t *R);

It reads the minimum (R[0]) and maximum (R[1]) value of θ.

void SetThetaMin(Float_t m);

It sets the minimum value of θ.

Float_t GetThetaMin();

It returns the minimum value of θ.

void SetThetaMax(Float_t m);

It sets the maximum value of θ.

Float_t GetThetaMax();

It returns the maximum value of θ.

N.B. The above described minimum and maximum values of θ correspond to θ_{MIN} and θ_{MAX} introduced in equation (A.2). The description of the rest of the parameters in this section is given in chapter 4.3.2.

void SetTheta_min(Float_t m);

It sets the value of θ of the centre of the first row of blocks θ_{min}.

Float_t GetTheta_min();

It returns the value of θ of the centre of the first row of blocks θ_{min}.

void SetTheta_max(Float_t m);

It sets the value of θ of the centre of the last row of blocks θ_{max}.

Float_t GetTheta_max();

It returns the value of θ of the centre of the last row of blocks θ_{max}.

void SetNumThetaBlocks(Short_t n);

It sets the number of blocks across θ at the top of the sector n_θ.

Short_t GetNumThetaBlocks();

It returns the number of blocks across θ at the top of the sector n_θ.

void SetPhi_min(Float_t m);

It sets the value of the centre of the upper-left (see Fig. 2.4) block of the sector φ_{min}.

Float_t GetPhi_min();

It returns the value of the centre of the upper-left block of the sector φ_{min}.

void SetPhi_max(Float_t m);

It sets the value of the centre of the upper-right block of the sector φ_{max}.

Float_t GetPhi_max();

It returns the value of the centre of the upper-right block of the sector φ_{max}.

void SetNumPhiBlocks(Short_t n);

It sets the number of blocks across φ at the middle of the sector n_{φ}.

Short_t GetNumPhiBlocks();

It returns the number of blocks across φ at the middle of the sector n_{φ}.

A.7 File management

void SetOverwriteMode(Bool_t OverwriteAll);

When creating a new file, if OverwriteAll==true, IMSexpert will overwrite all existing files with the same names. If OverwriteAll==false IMSexpert will ask the user for confirmation before overwriting. The default value of OverwriteAll is set to false.

Bool_t GetOverwriteMode();

It returns the value of OverwriteAll (see the above function).

A.8 Examples

The following example calibrates the experimental data using f_2 from equation (4.18):

```
#include <stdio.h>
#include <string.h>
#include <IMSexpert.h>
#include <TString.h>

void Example1(){

    //value of the neutral pion mass
    Float_t m_pion = 134.98;

    //name of the file containing the experimental data
    Char_t ImportFile[32];
    strcpy(ImportFile, "Data.root");

    //name of the file in which the calibrated values will be saved
    Char_t CalibData[32];
    strcpy(CalibData, "Data_calib.root");

    //name of the file in which the calibration coefficients will be saved
    Char_t CalibCoeff[32];
    strcpy(CalibCoeff, "calib_coeff.bin");

    IMSexpert IMStest;

    //importing the data
    if(!IMStest.ImportData(ImportFile)){
        return;
    }

    IMStest.SetRange(50,250); //range for the fitting procedure

    //experimental uncertainties of E, theta and phi
    IMStest.SetErrors(0.06, 0.03, 0.03);

    IMStest.SetOverwriteMode(true);

    Char_t str[32];
    strcpy(str, "data_fit.root");

    //fits the data and saves the resulting histogram in data_fit.root
```

```
    IMStest.IMS_Fit(str, m_pion);

    //sets the needed parameters for the calibration with function f_2
    IMStest.SetNumThetaBlocks(14);
    IMStest.SetTheta_min(14.34);
    IMStest.SetTheta_max(42.13);
    IMStest.SetNumPhiBlocks(17);
    IMStest.SetPhi_min(6.22);
    IMStest.SetPhi_max(53.78);

    //calibrates with type==0 (f_2)
    IMStest.Calibrate(m_pion, 0, CalibData, CalibCoeff);

    //analysing the calibrated data
    IMSexpert IMScalib;
    IMScalib.SetOverwriteMode(true);
    IMScalib.ImportData(CalibData);
    IMScalib.SetRange(50,250);
    IMScalib.IMS_Fit("calib_fit.root", m_pion);
}

int main(int argc, char **argv){

    Example1();
    return 0;
}
```

Appendix B

Mathematical model

This section shows the full mathematical derivations leading from equations (4.4) and (4.5) to equation (4.8).

Relations for single photons:

E : reconstructed energy of a photon;

θ : reconstructed polar angle of a photon;

φ : reconstructed azimuthal angle of a photon;

E_C : calibrated energy of a photon;

$$E_C := Ef\left(E,\theta,\varphi\right); \tag{B.1}$$

$$f(E,\theta,\varphi) := exp\left(\sum_i A_i F_i(E,\theta,\varphi)\right); \tag{B.2}$$

$$\frac{\partial f}{\partial A_i} = fF_i; \tag{B.3}$$

$$\Rightarrow \frac{\partial E_C}{\partial A_i} = E\frac{\partial f}{\partial A_i} = EfF_i = E_C F_i. \tag{B.4}$$

Relations for pairs of photons:

η : angle between the two photons;

$$M_{\gamma\gamma} := \sqrt{2E_{C1}E_{C2}(1-cos\eta)}; \tag{B.5}$$

$$\rho := lnM_{\gamma\gamma} - lnm_{\pi^0} = ln\left(\frac{M_{\gamma\gamma}}{m_{\pi^0}}\right); \tag{B.6}$$

$$\frac{\partial \rho}{\partial M_{\gamma\gamma}} = \frac{\partial (lnM_{\gamma\gamma})}{\partial M_{\gamma\gamma}} = \frac{1}{M_{\gamma\gamma}}; \tag{B.7}$$

$$\lambda_i := \frac{1}{2}(F_{1i} + F_{2i}); \tag{B.8}$$

$$\begin{aligned} \frac{\partial M_{\gamma\gamma}}{\partial A_i} &= \frac{1}{2}\frac{1}{M_{\gamma\gamma}}\left[2(1-cos\eta)\left(\frac{\partial E_{C1}}{\partial A_i}E_{C2} + E_{C1}\frac{\partial E_{C2}}{\partial A_i}\right)\right] = \\ &= \frac{1-cos\eta}{M_{\gamma\gamma}}\left[E_{C1}E_{C2}F_{1i} + E_{C1}E_{C2}F_{2i}\right] = \\ &= \frac{E_{C1}E_{C2}(1-cos\eta)}{\sqrt{2E_{C1}E_{C2}(1-cos\eta)}}(F_{1i} + F_{2i}) = \\ &= \frac{2E_{C1}E_{C2}(1-cos\eta)}{\sqrt{2E_{C1}E_{C2}(1-cos\eta)}}\frac{1}{2}(F_{1i} + F_{2i}) = M_{\gamma\gamma}\lambda_i; \end{aligned} \tag{B.9}$$

$$\mathcal{L} = \sum_{j=1}^{N}[lnM_{\gamma\gamma j} - lnm_{\pi^0}]^2 = \sum_{j=1}^{N}\rho_j^2, \tag{B.10}$$

where j represents the j-th pair of photons;

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial A_i} &= 2\sum_{j=1}^{N}\rho_j\frac{\partial \rho_j}{\partial A_i} = 2\sum_{j=1}^{N}\rho_j\frac{\partial \rho_j}{\partial M_{\gamma\gamma j}}\frac{\partial M_{\gamma\gamma j}}{\partial A_i} = \\ &= 2\sum_{j=1}^{N}\rho_j\frac{1}{M_{\gamma\gamma j}}M_{\gamma\gamma j}\lambda_{ij} = 2\sum_{j=1}^{N}\rho_j\lambda_{ij} \end{aligned} \tag{B.11}$$

$$R_i := \frac{1}{2}\frac{\partial \mathcal{L}}{\partial A_i} = \sum_{j=1}^{N}\rho_j\lambda_{ij}, \tag{B.12}$$

$$\begin{aligned} G_{il} := \frac{\partial R_i}{\partial A_l} &= \sum_{j=1}^{N}\frac{\partial \rho_j}{\partial A_l}\lambda_{ij} + \rho_j\frac{\partial \lambda_{ij}}{\partial A_l} = \\ &= \sum_{j=1}^{N}\frac{\partial \rho_j}{\partial A_l}\lambda_{ij} = \sum_{j=1}^{N}\frac{\partial \rho_j}{\partial M_{\gamma\gamma j}}\frac{\partial M_{\gamma\gamma j}}{\partial A_l}\lambda_{ij} = \\ &= \sum_{j=1}^{N}\frac{1}{M_{\gamma\gamma j}}M_{\gamma\gamma j}\lambda_{lj}\lambda_{ij} = \sum_{j=1}^{N}\lambda_{lj}\lambda_{ij}. \end{aligned} \tag{B.13}$$

Bibliography

[1] HADES, July 2012. http://www-hades.gsi.de/.

[2] C. Sturm and H. Stocker. The facility for antiproton and ion research FAIR. *Phys.Part.Nucl.Lett.*, 8:865–868, 2011.

[3] V. Friese. The CBM experiment at GSI / FAIR, 11 2005. https://www-alt.gsi.de/documents/DOC-2006-Dec-91.html.

[4] I. Frohlich et al. Future perspectives at SIS-100 with HADES-at-FAIR. *arXiv nucl-ex 0906.0091*, 2009.

[5] K. Lapidus et al. The HADES-at-FAIR project. *Phys.Atom.Nucl.*, 75:589–593, 2012.

[6] W. Czyzycki, E. Epple, L. Fabbietti, M. Golubeva, F. Guber, et al. Electromagnetic Calorimeter for HADES. *arXiv nucl-ex 1109.5550*, 2011.

[7] G. Agakishiev et al. The High-Acceptance Dielectron Spectrometer HADES. *Eur.Phys.J.*, A41:243–277, 2009.

[8] H. Alvarez-Pol et al. A large area timing RPC prototype for ion collisions in the HADES spectrometer. *Nucl.Instrum.Meth.*, A535:277–282, 2004.

[9] K. Lapidus. $\Lambda(1405)$: A New Hope. Measurements with the pion beam. *arXiv nucl-ex 1104.1926v1*, 2011.

[10] Particle Data Group (PDG), July 2012. http://pdg.lbl.gov/2012/tables/rpp2012-tab-mesons-light.pdf.

[11] David J. Tanner. Energy Calibration of the BaBar EMC Using the Pi0 Invariant Mass Method. 1998. http://www.slac.stanford.edu/cgi-wrap/getdoc/slac-r-850.pdf.

[12] ROOT, a data analysis framework, July 2012. http://root.cern.ch/drupal/.