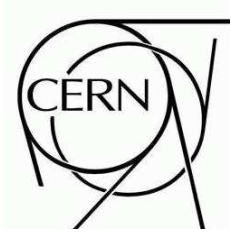
Draft version 1.0

ATLAS NOTE

December 19, 2008



Common Approach to ATLAS Performance Determination in Data

M. Schott, CERN, CH-1211, Genve 23, Switzerland

ATL-SOFT-PUB-2009-003
06 February 2009

Abstract

A possible approach for a common detector performance determination algorithms is presented in this note. It summarizes the outcome of ongoing discussions during past months. Special focus is given to the basic implementation design of performance determination algorithms inside the ATHENA-framework. Moreover, a common representation of efficiencies, resolutions and scales is proposed inside a newly-developed database structure.

Contents

1 Introduction

The determination of the ATLAS detector performance in data (often referred to as in-situ performance determination) is essential for all physics analyses and even more important to understand the detector during the first data taking period. Hence a common package for the in-situ performance determination inside the ATLAS software ATHENA would provide a useful tool for various applications. The package should contain standard algorithms for the determination of the detector performance, which should be provided by the combined performance and standard model working groups. Moreover, the determined detector performance should be represented in a coherent way within the package. In this note it is discussed what such in-situ performance determination algorithms should provide and how they should be structured and implemented. Moreover a common database structure for the storage of detector performance information is proposed.

One can distinguish between three different views of physics observables o in high energy physics experiments. The first view describes the real measured value of the observable, denoted as o_{Data}. The second view of o is the Monte Carlo simulated detector response of the measurement, which we denote as o_{MC}. The third and most relevant view is the truth value of the observable, i.e. o_{Truth}. The basic goal of each physics analysis is to deduce o_{Truth} from o_{Data}.

In general one has to understand the underlying detector in detail to perform such a deconvolution. The best guess of the detector understanding is given by the full Monte Carlo simulation of the detector response. The ATLAS experiment uses for this purposes a detailed detector description and the Geant4 toolkit to model the whole ATLAS detector [7]. This simulation step can be interpreted as a function

$$o_{MC} = f_{MC}^{Truth}(o_{Truth})$$

which takes as input the truth value of the observable and gives back the Monte Carlo prediction of the detector response. On the other hand, one needs to now the response detector response, i.e. the function

$$o_{Data} = f_{Data}^{Truth}(o_{Truth})$$

to perform the physics analysis, i.e $o_{Truth} = f_{Data}^{Truth}(o_{Data})^{-1}$. It must be goal for all groups, beginning from subdetector experts to physics groups to minimize the differences between f_{MC}^{Truth} and f_{Data}^{Truth}. This is a long and very challenging task, which is not addressed in this note. The remaining differences in the two functions can be described by a third function, which takes a input variable o_{MC} and adjusts it to the measured data, i.e

$$o_{Data} = f_{Data}^{MC}(o_{MC})$$

Hence, the function f_{Data}^{Truth} is nothing else as a convolution of f_{MC}^{Truth} and f_{Data}^{MC}. In some cases the function f_{Data}^{Truth} can be directly accessed, e.g. reconstruction or trigger efficiencies for leptons. In most cases such a direct estimation is not possible or not advisable. In these cases the Monte Carlo prediction has to be modified. It must be noted that is always preferable to adjust f_{MC}^{Truth} via f_{Data}^{MC} than trying to directly to model f_{Data}^{Truth} since all the known correlations and effects are automatically taken into account in the Monte Carlo predicted detector response. An example is the determination of the lepton momentum resolution which depends on the transverse momentum, the η and ϕ regime of the detector, the isolation and the underlying events. All these dependencies are taken into account via construction in the Monte Carlo simulation, while a direct assessment of f_{Data}^{Truth} require the description of these dependencies via hand. The goal of this note is to discuss a common approach for the determination storage of f_{Data}^{MC}.

The note is structured as follows. In section 2.1 some example methods for the in-situ performance determination are discussed and their common features summarized in section 2.2. The motivation and

basic functionality of a common framework is introduced in section 3.1. The logical separation of the algorithms, the basic implementation design inside the ATHENA framework and the representation of the performance quantities is discussed in sections 3.2 and 3.3, respectively. The last section 3.4 describes a possible database implementation in order to store and handle the determined detector performance.

2 Strategies for Performance Determination in Data

2.1 Some basic examples

In the following, three approaches for the detector performance determination in data have been chosen to illustrate their common features. These features imply already a basic structure of a common underlying framework. The determination of the lepton momentum scale and lepton trigger and reconstruction efficiencies will be discussed in detail. Other performance quantities, e.g. the jet energy scale, the b-tagging efficiency or the missing energy can also be accessed in-situ in data. It must be noted that the following discussion is only meant as an example of possible in-situ approaches and does not aim at covering physics analysis implementations in any detail. Some examples are discussed in detail in [2], [3] and [4].

2.1.1 Lepton Momentum Scale and Resolution

The knowledge of the energy scale and resolution for leptons is essential for many physics analyses, since it has a direct impact on the number of events which pass the kinematic selection cuts. One of the most common approaches for the determination of the scale and resolution within the energy range of 20 GeV and 60 GeV is the study of the reconstructed Z boson resonance. The energy resolution of the lepton reconstruction has a direct impact on the measured width, while the energy scale has a direct impact on the measured mean value. In order to determine the lepton momentum scale and resolution, the following algorithm could be used. The energy resolution function e predicted by Monte Carlo simulations is iteratively adjusted in its width and scale, via

$$e_{Data} \rightarrow f^{MC}_{Data}(e_{MC}, s, \delta\sigma) \tag{1}$$

where s describes the energy scale and $\delta\sigma$ corresponds to an additional resolution smearing. As an example the determination of the overall scale and resolution with a simple gaussian will be discussed, briefly. In this example case the transversal momentum for each muon track will be modified via

$$e_{Data} \rightarrow e_{MC} + gaus(x_m = s, \delta\sigma) \tag{2}$$

The parameter s describes an additive energy scale and $\delta\sigma$ represents the width of an additional gaussian smearing function. For each variation of the parameters, the resulting Z boson mass distribution can be calculated, by applying f^{MC}_{Data} on the full Monte Carlo simulated data. An effective variation can be achieved by χ^2-minimum search algorithm like *TMinuit* which searches for a minimal difference between the predicted Z boson mass distribution and the measured distribution. This iterative process stops if the new predicted Z boson mass distribution agrees within its statistical error to the measured distribution and the chosen parameters s and $\delta\sigma$ are defined as measured momentum scale and resolution, respectively. A schematic example of several predicted mass distribution for various values of s is shown in Figure 1. A detailed discussion can be found in [1].

This simple example can be extended for different kinematic regimes, e.g. a separation between $20\,\text{GeV} < p_T < 40\,\text{GeV}$ and $40\,\text{GeV} < p_T < 60\,\text{GeV}$. In this example three different Z boson mass

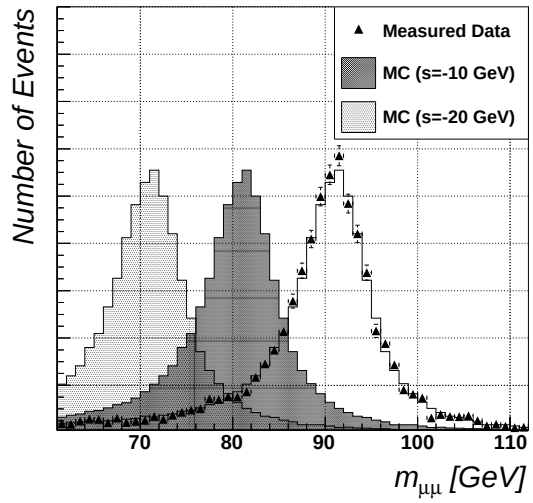


Figure 1: Measured invariant mass distribution and Monte Carlo predicted distributions for various lepton momentum scales.

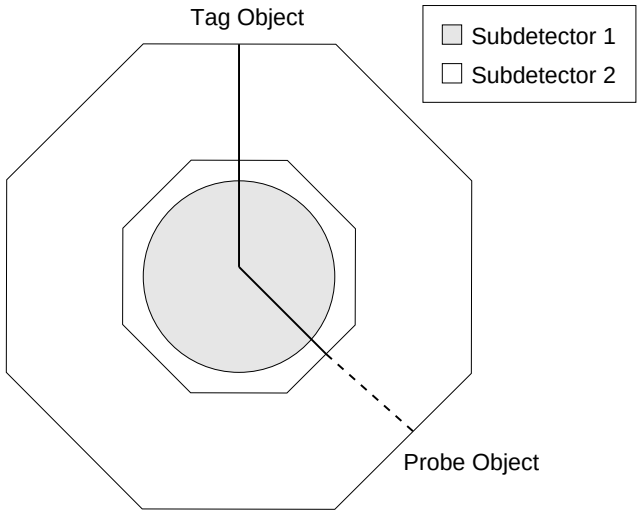


Figure 2: Principle of 'Tag and Probe'-method: Two independent subdetectors measure the same quantity. The probe object tests the efficiency.

distributions, for each possible combination of the two muons[1)], must be measured and fitted. This requires higher statistics to achieve good precision and leads to a multidimensional χ^2-fitting procedure.

Moreover, not only the Z boson decay, but also other well known resonances like the J/ψ decay can be used to determine the lepton momentum scale and resolution.

2.1.2 Muon Reconstruction Efficiencies without background

The basic principle to access trigger and reconstruction efficiencies in data is based on the so-called 'tag and probe' method. It relies on the fact that muons and electrons can be detected by two different detector systems, e.g. by the inner detector and the muon spectrometer or the electromagnetic calorimeter, respectively. Moreover, the decay of a well known particle is used. The decay of the Z boson into two muons is discussed here again as a first example in order to determine the reconstruction efficiency of the muon spectrometer. The two decay muons can be detected as two tracks in the inner detector and two corresponding tracks in the muon spectrometer. The so-called 'tag' muon is defined as a combined muon track, i.e. a track which is reconstructed both in the inner detector and the muon spectrometer. The 'probe' muon is an inner detector track, which has not been tested if it also matches with a muon spectrometer track, as illustrated in Figure 2. To minimize background under the peak, it is required that the tag and probe muon yield an invariant mass close to the Z boson mass. Moreover, kinematic and isolation cuts are applied to further reject background processes. This leads to a clean sample of Z boson decays, i.e. it is known that the probe-track was caused by a muon as a Z boson always decays into leptons of same flavor. It is crucial that no information from the muon spectrometer was used for the probe-track. Hence it can be tested if a track in the muon system can be associated to the probe-muon. Applying this procedure to a large sample of Z boson events leads to a direct determination of the overall track reconstruction efficiency of the muon spectrometer.

Several things have to be noted: the selection cuts, which are applied on the tag and probe muons, lead to a background free selection and the reconstruction efficiency can be tested for each event. Monte Carlo simulations predict a background contribution of less than 0.2%. In data this can be tested via

[1)] 1: both muons in the first region, 2: both muons in the second region, 3: each muon in a different region

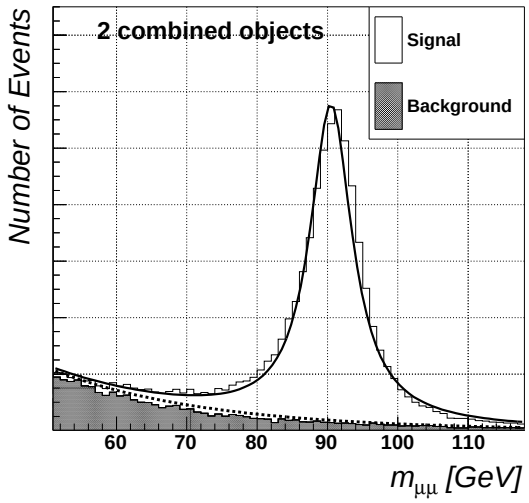


Figure 3: Invariant mass distribution for two tag-objects in each event.

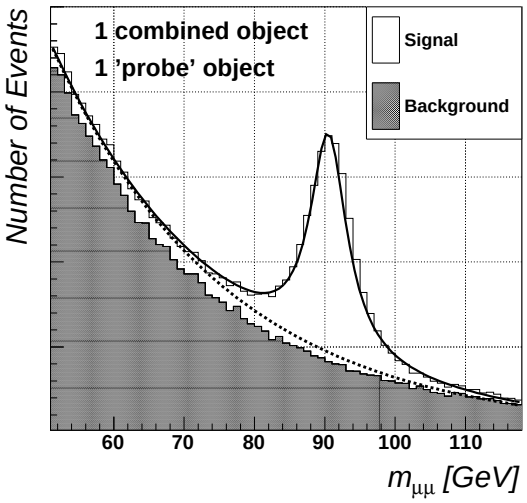


Figure 4: Invariant mass distribution for one tag-object and one probe-object in each event.

a like-sign, opposite-sign comparison. Therefore it is sufficient to store the selected probe-tracks for each event and the information on whether a muon spectrometer track could be associated or not. This information is enough to determine the reconstruction efficiency.

2.1.3 Muon Reconstruction Efficiencies with background contribution

The situation changes significantly if one does not rely on isolation requirements for the muon as QCD background contributions cannot be neglected anymore. The probe-track may arise from a background process and therefore the test for a corresponding signature in the Muon Spectrometer is likely to fail. In this case, the background has to be estimated from data, e.g. via a side-band subtraction approach which will be discussed for the Z boson decay in muons in the following. In a first histogram the invariant mass of all tag-objects per event, i.e. combined reconstructed muons, is plotted which is shown schematically in Figure 3. The invariant mass of the tag- and the probe-objects is plotted in a second histogram (Figure 4). Both distributions are fitted by a function which contains a signal and a background contribution, e.g. a combination of an exponential and a Breit-Wigner function. The area under the Breit-Wigner part can be interpreted as the number of signal events, i.e. events which result from a Z boson decay.

The following numbers of events are expected for the two cases:

$$N_{2\text{ tags}} \sim \varepsilon^2 \cdot N \tag{3}$$

$$N_{1\text{ tags, }1\text{ probe}} \sim \varepsilon(1-\varepsilon) \cdot N \tag{4}$$

where ε is the reconstruction efficiency of the Muon Spectrometer and N is the number of total selected signal (background-subtracted) events. The efficiency can then be determined by the ratio of $N_{1\text{ tags, }1\text{ probe}}$ and $N_{2\text{ tags}}$. It is important to note that the efficiency test cannot be achieved on an event-by-event basis in this case. Moreover, the calculation of a binned efficiency, e.g. for different η-regions of the detectors, requires at least two histograms for each kinematic region. This implies significantly larger statistics than the case previously discussed.

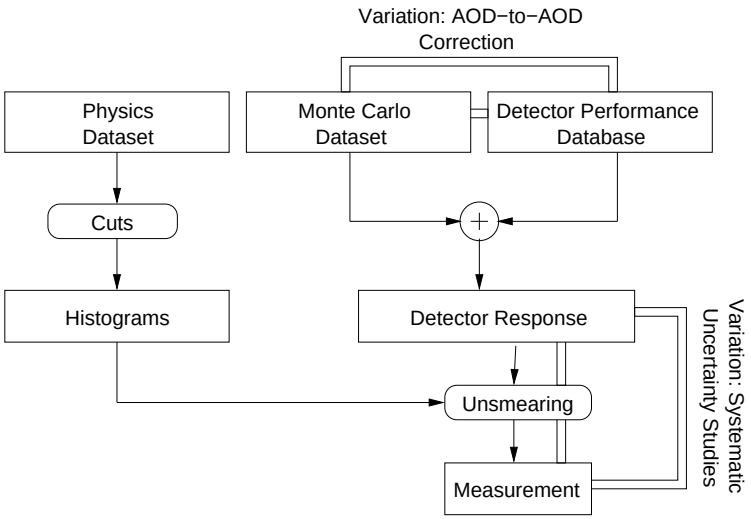


Figure 5: Schematic sketch of the physics analysis model.

2.2 Summary of underlying structure

The presented approaches seem quite different at a first glance, but have a common underlying structure containing three steps. All methods start with a signal selection, e.g. the selection of probe tracks, or the filling of histograms. In a second step, the corresponding performance quantity is calculated, e.g. by fitting several functions or counting how many probe tracks have been matched. The last step involves the representation of the performance quantity and how it is stored, e.g. via a simple histogram.

3 Common Framework Design

3.1 Motivation and basic functionality

The reasons for a common detector performance framework are numerous. The primary reason becomes evident when discussing a simplified physics analysis model as sketched in Figure 5. Each physics analysis is based on physics data, usually represented by histograms, which have been selected via several cuts. In order to understand these histograms, one also depends on detailed Monte Carlo simulations of the proton collisions and the corresponding detector response. The pure Monte Carlo simulation is always only a first guess and must be modified or at least confirmed by the detector performance, which has been directly measured in data, i.e. stored in a kind of detector performance database. Combining the Monte Carlo simulation of the detector and the in-situ determined performance quantities enables the unsmearing of the measured data and lead to the actual measurement. As already mentioned in the introduction this note does not address the difficult process of minimizing the difference between the Monte Carlo description of the detector response and the real measurements.

The database which contains the in-situ determined quantities can be used in two ways. First of all, it can be used if required to modify directly the Monto-Carlo simulated data, e.g. with an AOD-to-AOD correction. Secondly it can be used to estimate systematic uncertainties (e.g. lepton momentum scale) which results in differences in the unsmearing of the detector response. While the first analysis will usually be done through e.g. a common reprocessing of AOD data for a sub group, the second will be often performed by users.

A common detector performance database is however only one aspect of a common framework. It is usually sufficient for the estimation of systematic uncertainties in physics searches to use common

efficiency and resolution information stored in a common detector performance database. This does not hold for cross-section measurements, since they depend crucially on the precision of efficiencies and resolutions. Each cross-section analysis might use slightly different physics object selections. If such a selection does not correspond to the selection used for the efficiency determination in the database, then this will result in a systematic error in the measurement. In general it can be said, that the performance determination is such an essential part of each precision measurement that it should not be separated from the analysis itself. Therefore, the second aspect of a common framework is the general usability of the performance determination algorithms by every individual user. Nevertheless, even in the second case the common detector performance database must be used for cross-checks and validation of the individual or small-group analysis. The following points should be respected in order to ensure a general usability of the performance determination algorithms:

- Each algorithm for each physics object must be implemented within ATHENA and part of the official ATHENA release.
- The algorithms should be maintained and tested by the performance groups or in special cases by physics groups.
- All relevant cuts should be accessible via the *JobOption*-File and an example should be provided.
- For each in-situ algorithm a corresponding performance algorithm which is based on Monte Carlo truth quantities must be implemented, in order to allow systematic comparisons (e.g. the same definition of efficiency via track-matching must be used)
- The determined detector performance parameters should be stored for standard cuts in the detector performance database as a reference for systematic comparisons
- The authors of each algorithm should maintain documentation pages
- A common repository inside the ATHENA-framework would be desirable. The current location is under *PhysicsAnalysis/AnalysisCommon/InsituPerformance*

Following the above guidelines has various advantages for the average user who wants to perform a precision analysis: It will not be needed to develop one's own code for performance determination, but the existing, verified code inside the ATHENA-framework can be used and - if applicable - modified. Having the code inside the ATHENA-framework forces common standards and ensures the compatibility of the code in various environments. Hence also the existing grid infrastructures can be easily used.

Having a common framework with common standards enables the user to access and understand the performance quantities for various objects in the same way. E.g. having worked previously on the muon reconstruction efficiency it is straightforward for the user to access the electron identification efficiency, since a common representation infrastructure has been used for both.

3.2 Logical Seperation of Analysis Tools and Basic Implementation Design

As already discussed in section 2.2, the common algorithms have three basic phases. During the first phase, the primary event data in ESDs, AODs or DPDs is accessed and relevant information for the performance determination is extracted. Usually this information can be represented in histograms or NTuples. In the following, this information is called 'intermediate data'. The actual performance parameter determination is then based on these 'intermediate data' and not directly on primary event data.

It must be ensured that the size of intermediate data is small enough to be handled on a single computing note, since fitting algorithms usually have to be applied on combined data and cannot be

parallelized. Therefore it is essential that intermediate data is additive, i.e. can be added together from several grid jobs.

This basic requirement in illustrated in Figure 6, which shows the suggested computing model. The first part of the job is execute on the grid and each grid-job provides his own intermediate data. This data is then added together on the local machine and processed in order to determine the detector performance. Histograms, NTuples and DPDs can be added trivially and hence are a good choice for the representation of intermediate data.

It is proposed to use exclusively DPDs[2)] as intermediate data. DPDs have a significantly larger size than plain ROOT-NTuples as they contain event relevant parameters and a certain overhead to be readable by the ATHENA-frameworks. First estimates show a file-size of $500\,MB$ for the above described scenario, which is still easily manageable by a single PC. The writing out of plain ROOT-NTuples and histograms will be only supported for debugging purposes.

The content of the DPDs is fully in control of the user and the framework does not give any restrictions. Nevertheless, experience has shown that it is most convenient for most users to use rather simple and flexibly defined objects. Therefore the current implementation of the framework supports a generic *UserObject* which can be written via the official ATHENA-tools to DPDs on disk, i.e. no dependencies to other analysis packages are needed. The individual user can define this *UserObject* to his needs, i.e. to store only the relevant parameters per event which are needed for the performance determination.

Intermediate data, which are based on simple ROOT-NTuples usually contain basic physics object information, like energy, η or ϕ plus some additional information (isolation, track association, ...), while the histograms contain distributions of physics quantities, like invariant mass distributions. Even though the AOD data size contains several TB of data and has to be processed on the grid, the resulting intermediate data has only a small size. This is illustrated again on the muon efficiency determination based on the Z boson decay. Assuming 100.000 produced Z boson events per day, which decay into muons, per during the high luminosity phase of ATLAS, results in approximately 1 million probe tracks in ten days. Storing six float quantities per probe track leads to a data size of $\sim 100\,MB$, which can easily be handled on a local machine. The intermediate data which is based on histograms results in even less data and hence can also be handled on a single PC.

The large disadvantage of using plain ROOT-NTuples or Histograms as intermediate data, is the fact that their reprocessing for the performance determination does not fit to processing scheme behind ATHENA. ATHENA is based on an event-by-event evaluation scheme and performs overall fits at the end of the event-based processing. Since simple ROOT-NTuples cannot be processed within the ATHENA-framework they would have to be read in and evaluated completely within the *finalize()*-method of the algorithm, as the histograms. A further disadvantage results from the fact, that using plain NTuples and Histograms lead to the loose of information about the primary event, which makes debugging much harder.

The usage of DPDs as intermediate data already implies the basic implementation design which is shown in Figure 7. The actual performance determination is separated in two independent ATHENA algorithms. The first algorithm is responsible for the signal selection, i.e. the generation of the intermediate data and could be run on the grid. The signal selection itself is separated in an ATHENA tool from the algorithm to ensure a greater flexibility. Keep in mind, that the same signal selection might be used by different performance determination algorithms. For example the determination of the muon reconstruction efficiency and the muon trigger efficiency is based on the same selected probe tracks. The communication between the main algorithm and the ATHENA tool is handled via the Storegate service.

The performance determination (second phase), i.e. the application of fitting functions, as well as the storage of the results in a database (third phase) is done in the second algorithm. This algorithms treats the intermediate data as input and calculates directly efficiencies (as for example the muon reconstruction

[2)] DPDs are an Athena readable data format of NTuples

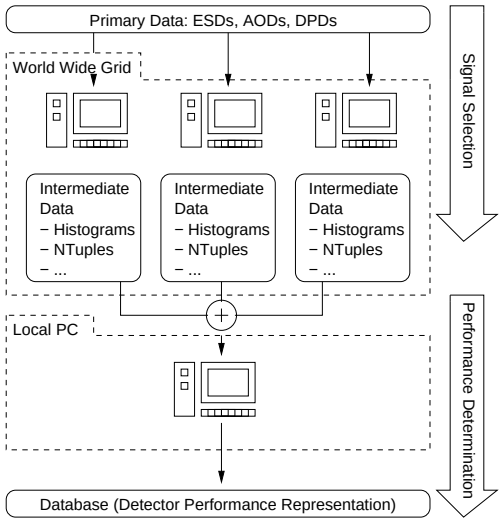


Figure 6: Sketch of the computing model for the insitu performance framework.

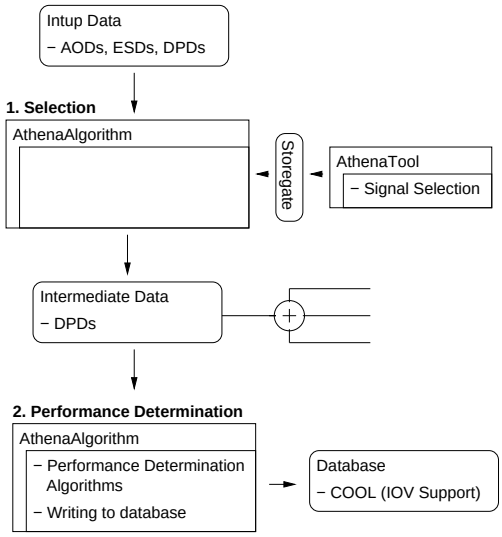


Figure 7: Sketch of the implementation structure of the insitu performance framework.

efficiency) or fills histograms, which are needed for example for the momentum resolution determination. In the *finalize()*-method of the algorithm all necessary calculations performed which need accumulated data, e.g. fits, are performed. It might be advantageous to encapsulate also the second phase in an external ATHENA tool, since some fitting algorithms might be common for various performance determinations. Moreover, the final results are written to the database after the algorithm has finished.

The separation into independent signal selection and performance determination algorithms has a further advantage: It allows to process several sets of intermediate data, which has been created by several different signal selections, by the same performance determination algorithms. The proposed structure even allows the user to run on small Monte Carlo samples in one go, i.e. the user is not obliged to run the two algorithms after one and the other but can do in parallel, as ATHENA-Storegate is used for writing and reading intermediate data.

3.3 Representation of Performance Parameters

Detector performance parameters can be generally categorised as efficiencies[3)] and resolutions[4)]. Both depend usually on several parameters like η, p_T and ϕ. Therefore the performance parameters should be represented by an N-dimensional matrix, where N is the number of parameters used for the description of the detector performance. The electron reconstruction efficiency can be described for example in a two dimensional matrix along p_T and η of the electron. Each bin of the matrix should contain

- measured performance parameter
- statistical uncertainty of the measurement
- weight of this measurement
- optional: an estimate of the systematic uncertainty of the measurement

[3)] Note that also fake-rates can be described as efficiencies
[4)] Note that scales fall also in this category

The weight of a measurement needs to be discussed in more detail: having such a weight allows the addition of several performance matrices. Keep in mind, that usually several performance matrices, each representing the measured performance of a certain number of data taking runs, have to be added in order to get an overview of the detector performance which is relevant for the analysed data-sets. If for example the physics analysis uses run 1 to 3 and the performance was estimated for each run separately, then the user wants to add the measured performance of these runs together. One could argue that this kind of weight information is encoded in the number of runs which have been used for the calculation, i.e. by the amount of data. But it has to be noted that different runs imply differences in the detector performance and hence differences in the signal selection efficiency. Therefore the weight should be stored explicitly.

In the following we discuss three different approaches to store the performance parameters where we distinguish between efficiencies and resolutions.

3.3.1 Efficiency Representation

For the background free efficiency determination (see section 2.1.2) it is proposed to store

$$\varepsilon = \frac{N_{Successes}}{N_{Trials}}$$

where $N_{Successes}$ is the number of trails which have been successfully chosen and N_{Trials} is the total number of trails. These two numbers, $N_{Successes}$ and N_{Trials}, can be used for the efficiency representation which has several advantages. First of all, these two numbers can be trivially added together from several runs and hence guarantees such an additiveness. Moreover, they contain already the statistical uncertainty information.

This representation is not suited for situations where a background is presented and e.g. a side-bind substruction has to be performed as discussed in section 2.1.3. In this case it is proposed to store an object which contains two histograms in each bin of the N dimensional matrix. The two histograms correspond to the two histograms introduced section 2.1.3. The object must also provide methods to perform the efficiency calculation based on fits on the two histograms. The advantage of this approach is again the additiveness of the two histograms for several runs.

3.3.2 Energy Response Representation

For efficiencies the function f_{Data}^{Truth} can be in general directly accessed in data as already mentioned in section 1. This changes for more complicated objects like the energy response of the detector, which describes in general scales and resolutions. In these cases it is proposed to determine the function f_{Data}^{MC} which modifies the Monte Carlo prediction f_{MC}^{Truth}. Hence we have to distinguish between the representation of f_{MC}^{Truth} and f_{Data}^{MC}.

The resolution and scale of an observable is encoded in the distribution of

$$\frac{q_{Truth} - q_{Reco}}{q_{Truth}}$$

where q_{Truth} is the truth value of the observed quantity and the q_{Reco} is the reconstructed value of the observed quantity. The scale is usually defined as the mean value of the distribution and the resolution corresponds usually to the rms value. Nevertheless it is clear that the full information is only represented in the full distribution, which can be interpreted as a probability density function (PDF).

It is proposed to store f_{MC}^{Truth} in each bin of the N dimensional matrix as a histogram of the corresponding PDF. The scale and resolution are automatically encoded in such a histogram. Moreover, no functional form must be assumed and also the additiveness is ensured.

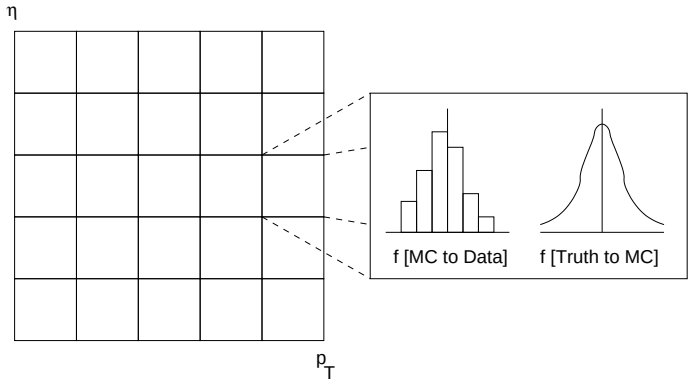


Figure 8: Illustration of the representation of scales and resolutions.

For the description of f_{Data}^{MC} it is usually sufficient to store a functional form which modifies the PDF of f_{MC}^{Truth}. An example would be a gaussian function. In this case only the mean and width of the gaussian would be stored in each bin of the N dimensional matrix. The large advantage of storing a functional form of f_{Data}^{MC} is that it allows also to describe a better resolution of the detector than it is predicted by Monte Carlo simulation. The most simple example of f_{Data}^{MC} which reduces the predicted resolution is given by

$$f_{Data}^{MC}(o_{MC}, o_{Truth}) = o_{Truth} + \sigma(o_{MC} - o_{Truth})$$

No additional smearing is applied for $\sigma = 1$, the resolution broadens for $\sigma > 1$ and gets better for $\sigma < 1$. It should be noted that in this case not only o_{MC} but also o_{Truth} is needed as input. In this example only the parameter σ would be stored in the performance matrix.

In summary it is proposed to store an object which represents f_{MC}^{Truth} or f_{Data}^{MC} in each bin of the performance matrix as illustrated in Figure 8.

3.4 Database-Structure

It was already mentioned in the previous sections that the detector performance parameters parameters should be stored in a common database. In this section a possible database structure is proposed. The basic features of such a database can be summarized as

- unique identification of a specific detector performance parameter, e.g. the muon spectrometer stand alone reconstruction efficiency
- storage of in-situ determined detector performance parameters
- storage of Monte Carlo predicted detector performance parameters for systematic comparisons
- handling of performance parameters for individual runs or luminosity blocks
- user friendly access for reading and writing inside and outside the ATHENA-framework
- official parameters provided by the performance groups for the ATLAS collaboration in particular for the physics groups

In Table 1 a collection of identifiers is presented which allow a unique specification of performance parameters. The identifiers *Object*, *Container*, *Type*, *Channel*, *Author*, *RecoSWV* are filled with simple string-variables by the user. The *Object* identifier describes which physics object is determined (e.g.

muons) and the *Container* identifier stands for the AOD-collection-name (e.g.) where the reconstructed quantities of the object were stored. The information of these two identifiers is redundant. Nevertheless, it allows for an easier search for users of the database, since the naming conventions of AOD containers are usually not self explanatory. The *RecoSWV* encodes the software version which was used for reconstruction. It might be important to specify which conditions data was used in the reconstruction and hence the AMI configuration tag might be a possible extension to this identifier. The last identifier in this list, called interval of validity (*IOV*), specifies which runs have been used for the performance determination. In case of a performance determination which was based on Monte Carlo simulation just the software version, which was used for the simulation, is used. Each performance parameter entry can be described in a unique way, once the identifiers are provided to the database.

Database ID	**Description**	**Examples**
Object	Which kind of physics parameter is stored	Muon, Electron, Tau, Jet, ...
Container	Name of AOD/ESD container	StacoMuonCollection, Trigger_Mu20, ...
Type	Description of Performance Parameter	Efficiency, Scale/Resolution, Fake-Rate, ...
Channel	Which physics channel was used	$Z \rightarrow \mu\mu$, $J/Psi \rightarrow ee$, $t\bar{t}$ ...
Author	Who is the author of the performance entry	MuonPerformanceGroup, PrivateMSchott, ...
RecoSWV	Software Version used for reconstruction (eventually also the AMI configuration tag)	14.0.21, ...
IOV	For data: Which runs have been used to determine performance (start and end number) For Monte Carlo: Simulation Software Release	 1432 - 1438 13.0.1

Table 1: Overview of identifiers for a unique representation of performance quantities in a common database

The actual implementation of the database depends on the use-case. One can differ between the usage for individual or small group physics analyses and an official ATLAS performance database. A simple ROOT-based database approach is not suited for an official database implementation. Hence, it is proposed to use a COOL-database for such a large scale application. An interface class could hide the actual implementation of the underlying database from the user.

It is worth to have a closer look to the basic principle of a COOL-database, which is illustrated in Figure 9. Each database entry can be accessed via a unique string identifier. To each entry also an IOV is stored. Simple data like a few numbers can be stored directly in the database. For more complicated objects, e.g. histograms, COOL stores only references to external files, e.g. ROOT-files [5].

Hence, it is proposed to use simple ROOT-files to store the performance parameters. These ROOT-files can then be added to the COOL-database, which handles the unique identification and IOVs.

An advantage of this approach is the easy access of users to performance quantities. It should be noted that a large part of final physics analysis are not carried out within the ATHENA-framework, but mainly in ROOT. Hence the access of performance parameters in ROOT must also be possible. A COOL-database allows an extraction of the relevant files, which contain the specified performance information.

It remains to be specified how the performance parameters are stored in the ROOT-file itself. Each performance parameter for a specified block is saved in a ROOT-directory whose name contains all database identifiers, previously discussed. Such a file can be inspected and simply accessed within the ROOT-framework, which will be the basis for most end-user analyses as already mentioned. The plain directory structure and naming convention gives an intuitive feeling for what is essentially stored in each

COOL − Database

DATA	IOV
Data [float, float, int, int , float, ...]	10−23
Data [float, int , int]	12−35
...	
Data [Reference to ROOT−File]	11−18
...	

Figure 9: Schematic structure of a COOL-database.

directory. Moreover, different ROOT-files can be easily merged together, which is essential if many persons work on the same or similar analyses.

4 Acknowledgements

I would like to thank Arno Strassner, Maarten Bonekamp, Daniel Froidevaux, Claudio Gatti and Richard Hawkings for their input and their long time for fruitful coffee discussions.

References

[1] ATLAS Collaboration, CSC-Note, ”In-Situ Muon Spectrometer Performance Determination”, CERN 2008, to be published as a CERN report

[2] ATLAS Collaboration, CSC-Note, ”J10+J9+J12: In situ calibration”, CERN 2008, to be published as a CERN report

[3] Hawkings, R., ”Flavour tagging calibration using topological analysis of t-tbar events”, ATL-PHYS-INT-2008-024; ATL-COM-PHYS-2008-083

[4] . ATLAS Collaboration, CSC-Note, ”J13+J14+J15 Etmiss”, CERN 2008, to be published as a CERN report

[5] Root-Homepage: http://root.cern.ch/root/doc/RootDoc.html

[6] ROOFit-Homepage: http://roofit.sourceforge.net/

[7] S. Agostinelli et al. Geant4: A simulation toolkit. Nucl. Instrum. Meth., A506:250303, 2003.