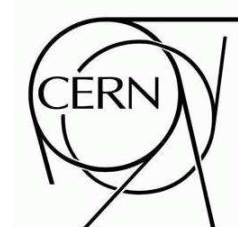




ATLAS NOTE

ATL-PHYS-PUB-2009-000

March 17, 2009



HLT Track Reconstruction Performance

The ATLAS Collaboration¹⁾

This note is part of CERN-OPEN-2008-020. This version of the note should not be cited: all citations should be to CERN-OPEN-2008-020.

Abstract

This note reviews the tracking algorithms used at the L2 and Event Filter stages of the High Level Trigger of ATLAS. The tracking performance (efficiency, resolution) is studied for different topologies (single tracks, high and low p_T jets) using simulated data. Detailed information on the execution time of the algorithms is also given.

¹⁾This note is prepared by S. Ask, J.T. Baines, A. Coccaro, D. Emelianov, I. Grabowska-Bold, J. Kirk, N. Konstantinidis, J. Masik, E. Ozcan, F. Parodi, C. Schiavi, S. Sivoklov, M. Sutton.



1 Introduction

The aim of this note is to describe the tracking algorithms used at the L2 and Event Filter stages of the High Level Trigger of ATLAS and to study their performance.

The definitions of the relevant quantities (efficiency, fake rate, and resolution) given in this note can differ with the ones used in the different selection algorithms of specific trigger objects (e/γ , muon, taus, b -jet and B -physics): here the purpose is to define a common language to study and compare the different tracking algorithms.

A detailed description of the complete ATLAS detector and its performance can be found in [1]. In addition to a description of the trigger system, it also contains relevant information on the ATLAS tracking detectors, the Inner Detector (ID) and its subsystems (Pixel, SCT and TRT).

2 Track reconstruction at L2

2.1 Data preparation

Detector data should be converted before it can be used by tracking algorithms. The conversion process includes the bytestream decoding, the cluster formation in the Pixel and SCT detectors [1], and their conversion in spatial coordinates (space points).

2.2 IDScan

IDScan is a set of algorithms for fast pattern recognition and track reconstruction at the second level trigger, using space points provided by the tools described in Section 2.1. These algorithms first determine the z -position of the interaction point along the beam axis and then perform combinatorial tracking only inside groups of space points that point back to that determined position.

The first algorithm, aiming to determine the z -position of the primary vertex, divides the region-of-interest (RoI) into many equally-sized ϕ slices, whose width is tuned according to the individual RoI type, based on the lowest track momenta desired and the level of background hits in the detector. While tracks of high momenta produce most of their hits in a small number of neighbouring slices, hits from lower-momentum, curved tracks populate several different slices. Every space point is paired (or optionally every space point from the innermost three silicon layers) in each slice to the other space points in that slice and in a few neighboring slices, and each pair is used to calculate a z -position by linear extrapolation to the beam line. (This exploits the fact that helical trajectories of charged particles in a solenoidal magnetic field are straight lines in the ρ - z projection.) A one-dimensional histogram accumulates all the calculated z values, and the peak(s) in this histogram provide the rest of the IDScan algorithms with the z -coordinate of the pp interaction point. The correct position is identified in more than 98% of the RoIs, with a resolution between 150 and 200 μm (depending on the type of RoI) for the central RoIs.

Using the z -position previously reconstructed, the second algorithm computes the pseudorapidity for all the space points in the RoI and fills a two-dimensional histogram in (η, ϕ) . Since all hits from a given (sufficiently) high- P_T track tend to be contained in a small solid angle (with its apex at the origin for the track), the space points from each track that originates from the computed z -position on the beam axis form a cluster of neighbouring bins in this histogram. When the bin size is small enough, the occupancy for each bin is low and each cluster, called a group, often contains the space points of a single track. Fake candidates are reduced by keeping track of which detector layers contribute space points to each bin and requiring that at least four out of an expected seven layers to have contributed to a given bin or its immediate neighbours, before that bin is included in a group.

After the groups have been identified random space points and/or space points from multiple tracks in each group are separated. This is achieved by considering all possible triplets of space points within a group and making use of the fact that any three hits from a track can be used to extract the same track parameters ϕ_0 and $1/P_T$ in the transverse plane. The algorithm fills a two-dimensional histogram with extracted $(\phi_0, 1/P_T)$ values and considering combinations containing space points from at least four different silicon layers.

Finally the cleaned groups are subjected to the clone removal algorithm, which identifies groups sharing at least a certain number of space points (currently 2 or 3 depending on the RoI type) and removes all but the one with the highest number of space points. Furthermore, a group is removed if it shares more than 45% of its space points with others. This step significantly reduces the number of fake groups that contain a few random space points in addition to a small number of space points from an actual track.

After all these steps, the remaining groups are passed on to a track fitter. The default fitter used by IDScan is described in Section 2.5.

2.3 SiTrack

The SiTrack L2 algorithm adopts a combinatorial pattern recognition approach to reconstruct tracks starting from space points formed in the ID silicon detectors.

In order to perform space point combinations, these are first of all grouped into sets from which the entries of each combination will then be extracted; the grouping is implemented in SiTrack, using the idea of “logical layers”. These correspond to a list of physical detector layers, i.e. barrel layers and end-cap disks, and are labeled with increasing numbers moving away from the beam line. The same physical layer can be included in more logical layers, to increase the robustness of the track finding process. To provide a tangible example, the first logical layer adopted for the reconstruction of high- p_T isolated leptons includes the innermost two pixel barrel layers and the innermost pixel end-cap disk.

Once the space points have been associated to the logical layers they belong to, the track reconstruction algorithm proceeds through the following five steps:

- formation of track seeds;
- optional primary vertex reconstruction along the beam line;
- extension of track seeds;
- merging of extended seeds;
- clone removal.

The formation of track seeds corresponds to a combinatorial pairing of space points coming from the innermost two logical layers. For each seed, the extrapolation to the beam line is evaluated, using a straight line approximation; this process is depicted in Fig. 1. At this point a cut on the transverse impact parameter is applied. This cut, meant to reduce the number of seeds to be further processed, is particularly important, as it fixes the lowest reconstructible track p_T value.

The subsequent step is the reconstruction of the position of the primary interaction vertex along the beam line, used to reject tracks not coming from the primary interaction. The vertex reconstruction is performed filling a histogram with the longitudinal impact parameter of the seeds and searching for histogram maxima; more vertex candidates can be retained and seeds not pointing to any of the reconstructed vertexes are discarded. This optional step is useful for high track multiplicity topologies like jets, but is typically skipped in the case of low multiplicity event topologies, e.g. for the reconstruction of single isolated leptons. Each retained seed is extended, as depicted in Fig. 1, extrapolating it to the

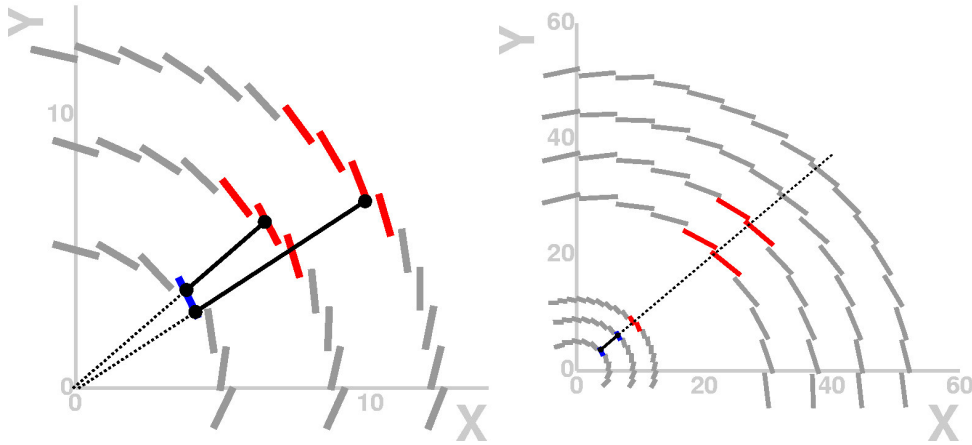


Figure 1: Pictorial scheme of the SiTrack combinatorial strategy for track seeds formation (left) and track seeds extension (right).

outer logical layers and forming one or more space point triplets for each seed; extensions are selected applying a cut on the distance between the outer space point and the extrapolated seed. Each extended seed is then fitted by a straight line in the longitudinal plane and parametrized as a circle in the transverse plane.

All the extensions found for each seed must then be merged into a single full track, grouping the triplets having similar track parameters after the fit. The full track is thus formed by the union of the space points from all the merged extensions. All the triplets not involved in the merging process are discarded, while track parameters are re-evaluated for the full track, fitting it with a straight line in the longitudinal plane and a circle in the transverse one.

Two full tracks obtained from different track seeds may still share most of their space point; these tracks are defined as clones. To eliminate these ambiguous cases, only the clone track containing the largest number of space points is retained; in case more clone tracks contain the same number of space points, the one with the lowest χ^2 value prevails. The retained full tracks are finally refit using one of the available common fit tools.

2.4 TRT tracking

Information from the TRT part of Inner Detector can be used as the basis of a L2 tracking algorithm. The core of the algorithm is a set of utilities from the offline reconstruction package xKalman [2] for the reconstruction of tracks in the TRT detector. It is based on the Hough-transform (histogramming) method. At the initialization step of the algorithm, a set of trajectories in the $\phi - R(Z)$ space is calculated for the barrel and endcap parts of the TRT. The value of the local magnetic field is taken into account at each straw position and coordinate along it when calculating the trajectories. After initialization, a histogram (with a size of 500 bins in ϕ and 70 bins in curvature) is filled for each event with the TRT hit positions. The track candidates can be identified from peaks in the histogram. Bins with at least eight hits are considered as track candidates. These track candidates should satisfy some quality parameters like the number of unique hits and the ratio of hits to number of straws crossed by the trajectory. For each track candidate, the parameters are tuned so that the track lies on the maximum number of drift circle positions. It is at this stage that drift information is taken into account to further improve the resolution of the track parameters.

2.5 Track fitting tools

The track fitting procedure used by the L2 ID algorithms in Pixel and SCT detectors is based on a Kalman filtering technique.

At first, space points are dissolved into clusters and a filtering node is created for each cluster. The filtering nodes encapsulate implementations of the Kalman filter algorithm for various measurement models. The fitter object uses the filtering nodes to update a track state described as a 5-dim vector of track parameters (local x , local y , angles ϕ and θ given in the global coordinate system, and track inverse momentum Q) and corresponding covariance matrix.

The track state update consists of three steps. First, the track state is extrapolated using a simple parabolic approximation of a trajectory in uniform magnetic field. The material-related corrections (multiple scattering, energy losses) are added to the covariance matrix during this step. The extrapolated track state is used to *validate* the next hit: if the χ^2 distance between the hit and state is less than a predefined cut for this filtering node the track state is updated. These “extrapolate-validate-update” steps are repeated for every node. After that, a standard backward smoother is applied.

There exists another tool which performs track fit and simultaneous pattern recognition in the TRT. The implementation of this tool is based on a distributed approach. More details on the L2 TRT track extension tool can be found in Ref. [3].

2.6 Vertex Fitting tools

An essential part of the L2 event selection (e.g. B -physics event selection) is vertex finding and fitting using tracks reconstructed by the L2 tracking algorithms as input. Due to the L2 timing constraints a vertex fitting algorithm for the L2 application has to be fast. An additional requirement stems from the L2 track reconstruction which provides input track parameter errors in form of a covariance matrix. In contrast, vertex fitting algorithms proposed in literature assume uncertainties of the input track parameters to be described by weight (inverse covariance) matrices. However, if only track covariance matrices are available, these algorithms require them to be inverted beforehand thus resulting in substantial computing time overhead.

To alleviate this drawback a fast vertex fitting algorithm capable of using track covariance matrices directly (i.e. without time-consuming inversion) has been developed. The specific feature of the algorithm is that track momenta at perigee points rather than “at-vertex” momenta are selected as the fit parameters. Such a choice of fit parameters makes it possible to apply a decorrelating measurement transformation so that the transformed measurement can be partitioned into two uncorrelated vectors – measured momenta and its linear combination with measured track coordinates at the perigee. This linear combination comprises a new 2-dim measurement model while the measured momenta and the corresponding blocks of the input track covariance matrices are used to initialise a vertex fit parameter vector and covariance matrix of the Kalman filter. This approach provides a mathematically correct and numerically stable initialisation of the vertex fit. A reduced size (2-dim instead of the usual 5-dim) of the measurement model makes the proposed Kalman filter very fast and therefore suitable for an online application in the ATLAS Level 2 Trigger. A detailed description of the L2 vertex fitting algorithm can be found in Ref. [4].

3 Track reconstruction at EF

ID reconstruction at the EF is performed using ATLAS “New Tracking” software [5]. A common approach between offline and online is possible thanks to the modular New Tracking design which allows the replacement of time-critical components and full-featured offline modules by trigger-specific implementations. New Tracking currently covers two sequences, the main inside-out track reconstruction

(track finding starts from the Silicon and then is extended to the TRT) and outside-in tracking (from the TRT to the Silicon). The primary pattern search concepts for both sequences have been to a large extent adopted from the already existing ATLAS ID reconstruction program xKalman [2], but integrated and incorporating additional components in the common New Tracking approach. In the following note, only the inside-out reconstruction of tracks is described since it is the only one used in the EF ID online reconstruction. In the future, an outside-in approach is intended to be used in the trigger in cases where photon conversions are present.

The EF ID reconstruction runs for many different signatures, such as electrons, muons, taus, and b -jets. Each of these triggers contains a very similar algorithm sequence as the ID inside-out tracking and is followed, depending on a given object, by dedicated event reconstruction algorithms including vertex finding, b -tagging or electron processing. In the EF realization of New Tracking dedicated algorithms steer the underlying tools with RoI-seeded input collections ²⁾. The tools used are directly taken from the offline reconstruction chain but operated in a RoI-seeded mode, where the trigger signature defines a width of the RoI.

The EF ID algorithm sequence is divided into three stages defined as pre-processing, inside-out track finding and post-processing. The pre-processing stage is responsible for building clusters and drift circles in the Silicon and TRT detectors, respectively, and the creation of space points as three-dimensional representations of the Silicon detector measurements.

The inside-out track finding starts from the Pixel and SCT to find track seeds and creates track candidates based on the seeds primarily found. The seeded track finding results in a very high number of track candidates, that have to be resolved before an extension into the TRT detector can be done. Many of these tracks share hits, are incomplete, or describe fake tracks, hence ambiguity resolution is necessary. The track extension from the Silicon to the TRT is divided into two modules. First, tracks found in the Silicon detector are used as an input to find a compatible set of TRT measurements. Then each extended track is evaluated with respect to the original Silicon track. A track scoring mechanism is then used to compare the original track with the one after refitting, and the best track is chosen.

The last stage, post-processing, starts from a primary-vertex search, which is based on the Billoir fitting method [6]. A track object is created which is a representation of the track reconstruction results aimed for analysis applications.

Currently several different fitting techniques are implemented in New Tracking and can be chosen at a configuration level:

- Kalman Fitter as a straightforward implementation of the Kalman filter technique [7] that has been adopted for the track fitting in high energy physics experiments. For the ATLAS Silicon detector, the Kalman Fitter has a dedicated extension for fitting of tracks from electrons, that lose stochastically a significant part of their energy due to bremsstrahlung effects. In that case, an assumption about purely Gaussian noise is far from being optimal. A special Dynamic Noise Adjustment technique has been developed [8]. It still uses a Gaussian error assumption but modifies the applied variance based on the amount of traversed material.
- Deterministic Annealing Filter is a deterministic annealing technique [9] which combines the standard Kalman filter formalism with a probabilistic description of the measurement assignment to a track.
- Gaussian Sum Filter is a special multi-Gaussian extension of the standard Kalman fitter [10], dedicated to reconstruction of electron tracks. In the GSF approach, the highly non-gaussian probability density function of electron energy loss is modeled by a mixture of several Gaussians.

²⁾The FullScan operation is an exception which assumes track reconstruction in the entire detector.

- Alignment Kalman Filter is an extended version of the Kalman filter [11] that integrates an update of the detector surface orientation and position into an intrinsic measurement update of a Kalman Fitter step.
- Global Chi2 Fitter [12] is a track fit through a minimization of χ^2 value. Given purely Gaussian process noise, the minimization of the χ^2 value that is built from hit residuals at every measurement surface gives the best set of estimators of the track trajectory. Material effects enter the χ^2 function as additional fitting parameters.

The Kalman Fitter approach without the Deterministic Annealing Filter extension is a default fitter at the EF ID. Other fitters described above can be chosen during the configuration step.

4 Performance

4.1 Timing measurement

This section presents a summary of CPU timing measurements for various steps of data preparation and track reconstruction in L2 ID and EF ID algorithms.

4.1.1 L2 ID

The L2 ID timing has been measured on a quad-core 3GHz Woodcrest CPU machine. Timers provided by the standard trigger monitoring framework have been used for these measurements.

The data preparation timing measurements obtained on $t\bar{t}$ data for the Pixel and SCT are shown in Table 1 for e/γ , muon, and tau triggers. The average cluster collection and space point multiplicities per RoI are presented in Table 2.

Data preparation step	Mean time [ms]		
	μ	e/γ	τ
RegionSelector	0.23	0.26	0.32
RobDataProvSvc	0.02	0.02	0.02
Cluster IDCs retrieval	0.03	0.03	0.02
BS-to-clusters, Pixel	0.63	0.73	1.02
BS-to-clusters, SCT	0.60	0.69	0.95
Pixel space point formation	0.24	0.27	0.35
SCT space point formation	0.31	0.37	0.40
Total time	2.11	2.45	3.23

Table 1: The timing measurements of the data formation per RoI.

The track reconstruction timing measurements are summarized in Table 3 (note that SiTrack fit timing is included in the pattern recognition time). The average track multiplicity is presented in Table 4.

Tables 5,6, 7, and 8 present a comparison between RoI-based and FullScan running for B -physics triggers on $b\bar{b} \rightarrow \mu X$ data. These show, respectively, the time of data formation, average space point multiplicity, time of L2 ID track reconstruction, and average track multiplicity for the two cases.

Data preparation step	Multiplicity		
	μ	e/γ	τ
Pixel cluster coll.	46.2	45.6	104.4
SCT cluster coll.	82.8	83.5	245.8
Pixel space points	66.4	65.9	131.6
SCT space points	36.5	38.8	87.1

Table 2: Average cluster/space point multiplicities per RoI.

Track reconstruction step	Mean time [ms]		
	μ	e/γ	τ
IDScan : Pattern recognition	0.60	0.70	1.45
IDScan : Track Fit	0.28	0.29	0.49
IDScan : TRT data preparation	1.14	1.32	–
IDScan : TRT tracking	2.96	2.87	–
SiTrack : Pattern recognition	0.62	0.66	–
SiTrack : TRT data preparation	1.15	1.39	–
SiTrack : TRT tracking	2.85	3.66	–

Table 3: The timing measurements of the L2 ID track reconstruction with IDScan and SiTrack.

4.1.2 EF ID

The EF ID timing measurements are shown in Table 9. They were done on a 3 GHz Intel Xeon 5160 CPU with 8GB of memory. The EF ID software was run in the emulator of the event filter processing and execution times for the algorithms were obtained as mean values from the timing histograms provided by the HLT framework. The results are representative in terms of algorithm execution times but do not account for data collection times, which are platform dependent. Two event samples were used in the tests, one corresponding to a single electrons with 100GeV, the other was a simulation of $t\bar{t}$ production.

The most CPU demanding operation is the track finding in the silicon detectors with typical execution times of 30 ms per RoI (300 ms in FullScan mode). Next is the processing of TRT extensions and resolution of ambiguities each with a CPU cost of about 20 ms per RoI (approximately 170 ms in FullScan). Another important contribution to the processing time is the data preparation step which takes about 20 ms per RoI for all detectors (and about 300 ms in FullScan mode). The total time to process $t\bar{t}$ events in full-scan mode is about 1s.

L2 ID algorithm	Tracks/RoI		
	μ	e/γ	τ
IDScan	1.92	1.63	4.50
SiTrack	2.05	2.66	–

Table 4: The average track multiplicity per RoI for L2 ID track reconstruction with IDScan and SiTrack.

Data preparation step	Mean time [ms]	
	RoI 0.75×0.75	FullScan
RegionSelector	0.67	4.76
RobDataProvSvc	0.02	0.41
Cluster IDCs retrieval	0.03	0.06
BS-to-clusters, Pixel	2.13	11.7
BS-to-clusters, SCT	1.89	13.1
Pixel space point formation	0.78	5.27
SCT space point formation	0.68	5.65
Total time	6.67	44.5

Table 5: The timing measurements of the data formation for RoI-based and FullScan running of B -physics triggers.

Data preparation step	Multiplicity	
	RoI 0.75×0.75	FullScan
Pixel cluster coll.	266.2	1744
SCT cluster coll.	824.7	8176
Pixel space points	246.2	1591.5
SCT space points	119.6	910.3

Table 6: The average cluster/space point multiplicities for RoI-based and FullScan running of B -physics triggers.

4.2 Efficiency and resolution definition

In order to evaluate the performance of any tracking algorithm, three kinds of information have to be provided: track reconstruction efficiency, the percentage of fake reconstructed track candidates, and the resolution of the track parameters.

Track reconstruction step	Mean time [ms]	
	RoI 0.75×0.75	FullScan
IDScan : Pattern recognition	3.00	34.6
IDScan : Track Fit	0.55	2.19
IDScan : TRT data preparation	2.05	9.50
IDScan : TRT tracking	5.43	19.5

Table 7: The timing measurements of the L2 ID track reconstruction for RoI-based and FullScan running of B -physics triggers.

L2 ID algorithm	Tracks/RoI	
	RoI 0.75×0.75	FullScan
IDScan	4.26	16.9

Table 8: The average track multiplicity for RoI-based and FullScan running of the B -physics triggers.

Algorithm		Mean time [ms]				
		single e 100GeV	$t\bar{t}$			
		Electron	Electron	Muon	Tau	FullScan
Data preparation	Pix	2	2	2	3	61
	SCT	8	8	10	10	85
	TRT	8	6	8	9	140
Space point finder		1	1	1	2	30
Track finding in Si		6	29	6	31	310
Ambiguity solving		4	15	5	11	135
TRT track extensions		1	3	1	1	31
TRT extension processing		5	19	8	13	170
Vertex finding		0.1	1	1	-	23
Particle creation		0.4	2	1	1	24
Total		35	86	43	81	1009

Table 9: Timing of EF reconstruction steps per RoI in two samples of events, single e^- of 100 GeV and $t\bar{t}$ events. Electron, muon, and tau times were measured in the corresponding triggers, FullScan mode comes from the execution of B-physics triggers. The RoI sizes were 0.1×0.1 ($\Delta\phi \times \Delta\eta$) for the electron and muon triggers, 0.2×0.2 for tau triggers.

All the results shown in this section refer to reconstructed Monte Carlo (MC) simulated data samples, where the track parameters for both the reconstructed and the simulated tracks are available. In addition, each space point used to build a given track can be traced back to the MC particle that generated the corresponding charge deposit. This information is used by the ID trigger analysis packages to evaluate two additional quantities for each reconstructed track, which prove fundamental for the efficiency definition:

- the number of space points which trace back to the same MC track;
- the link between a reconstructed track and the MC track that generated most of its space points.

In this context we define:

- reconstructible MC particle: a particle passing a set of geometrical selection cuts (being contained in one of the processed RoIs, pointing to the primary vertex) and a p_T cut;
- good track: a reconstructed track linked to a reconstructible particle by the majority of its space points; the geometrical and p_T cuts used for the MC particles are applied to the reconstructed tracks too;
- best track: for each reconstructible particle, more than one good track can be available; the best track is defined as the one sharing the largest percentage of space points with the linked particle;
- fake track: a reconstructed track (passing geometrical and p_T cuts) which is not a good track;

The most natural choice for the definitions of efficiency, fake fraction and resolution is then:

- efficiency: ratio between the best tracks and reconstructible particles;
- fake fraction: ratio between fake tracks and all the reconstructed tracks passing the applied cuts;
- resolution: difference between the track parameter of a good track and that of the linked reconstructible MC particle;

These definitions are used to produce the set of plots used in the following subsections to summarize the performance of a given tracking algorithm.

The track reconstruction efficiency is shown as a function of the absolute value of η and p_T of the reconstructible MC particle while the fake fraction is shown as a function of the absolute value of η and p_T of the reconstructed track. The track parameter resolutions are shown as a function of the absolute value of η and p_T of the reconstructible MC particle: the resolutions on ϕ , $1/p_T$, transverse impact parameter (d_0) and longitudinal impact parameter (z_0) are shown as a function of the absolute value of η while resolutions on d_0 and $1/p_T$ are shown as a function of p_T . The track parameter resolutions are evaluated with a gaussian fit to the resolution distribution for each parameter.

4.3 Results with isolated electrons

The tracking performance for electrons was evaluated with a data sample of single electrons uniformly distributed over a transverse momentum range of 7 to 80 GeV. Figure 2 shows the reconstruction efficiency and the fake fraction as a function of η and p_T for SiTrack, IDScan and EF tracking algorithms. Figure 3 summarizes, for the same algorithms, the track parameters resolutions. As shown in the figures, the efficiency is typically 95% or greater, except for low p_T or high η where it drops to about 90%. The fake fractions tend to be below 2% except at low p_T or high η where they can exceed 6%. Resolutions are observed to be quite good and are also somewhat degraded at low p_T or high η .

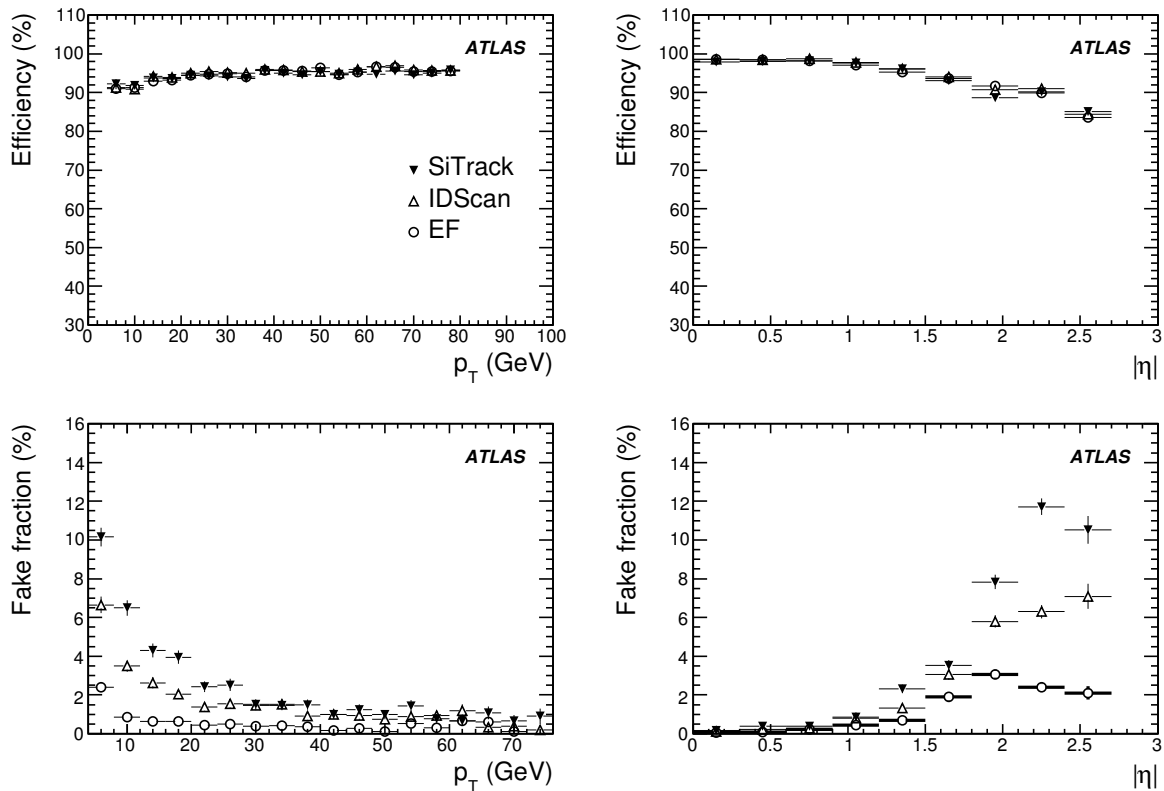


Figure 2: Electron track reconstruction efficiency (top) for single electrons as functions of p_T and η for SiTrack (full triangles), IDScan (empty triangles) and EF tracking (empty circles). Bottom plots show fake fraction as a function of the reconstructed p_T and η .

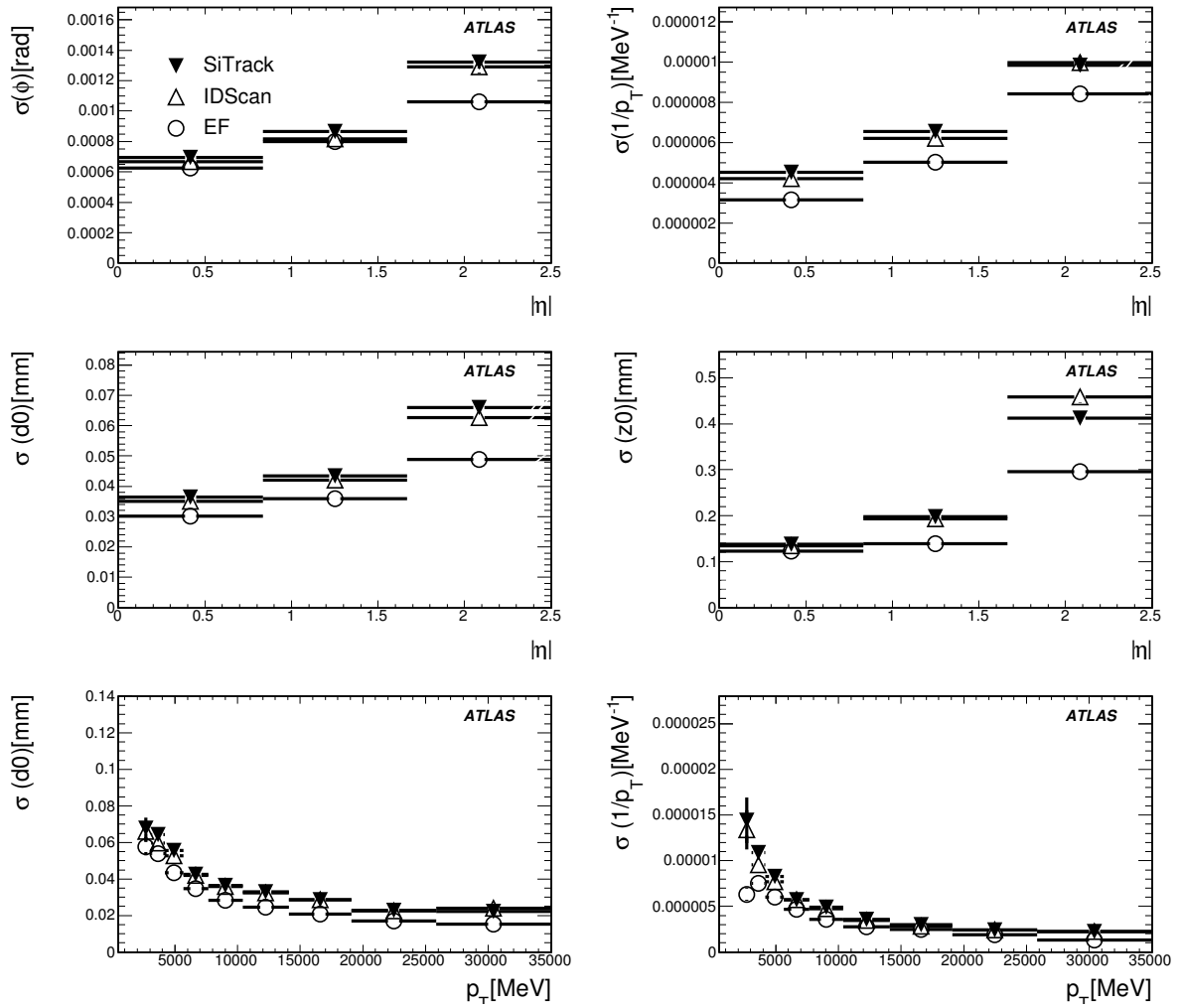


Figure 3: Track parameter resolutions for single electrons as a function of η and p_T for SiTrack (full triangles), IDScan (empty triangles) and EF tracking (empty circles).

4.4 Results with isolated muons

The tracking performance for muons was evaluated with a data sample of single muons with transverse momenta of 6, 9, 21 and 30 GeV. Figure 4 shows the reconstruction efficiency and the fake fraction as a function of η and p_T for SiTrack, IDScan and EF tracking algorithms. Figure 5 summarizes, for the same algorithms, the track parameter resolutions. As shown in the figures, the muon performance is even better than for electrons, with close to 100% efficiency throughout the kinematic range and an extremely low fake rate. Resolutions are slightly better than for electrons, and show similar p_T and angular dependence.

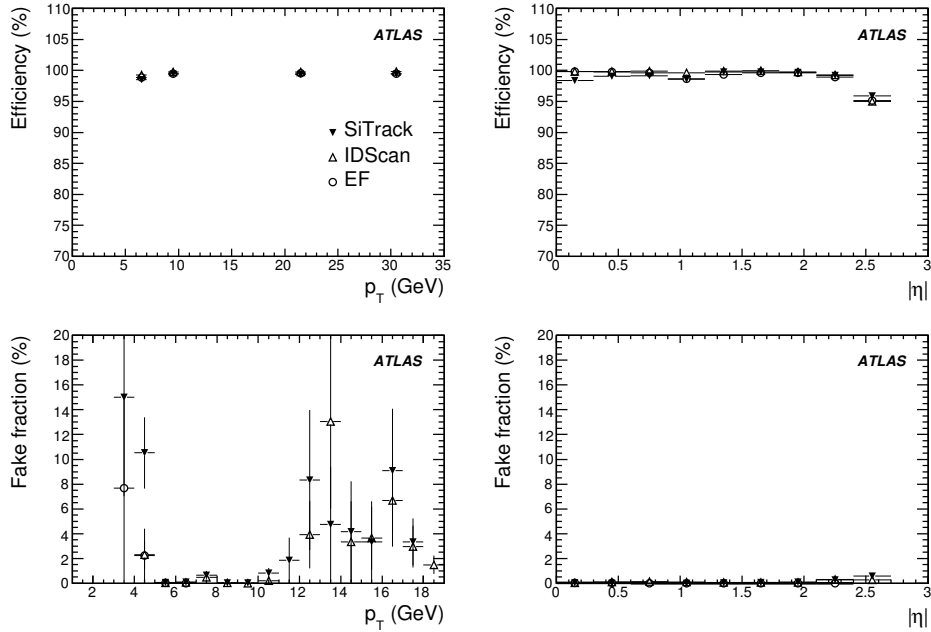


Figure 4: Muon track reconstruction efficiency (top) for single muons as functions of p_T and η for SiTrack (full triangles), IDScan (empty triangles) and EF tracking (empty circles). Bottom plots show fake fraction as a function of the reconstructed p_T and η .

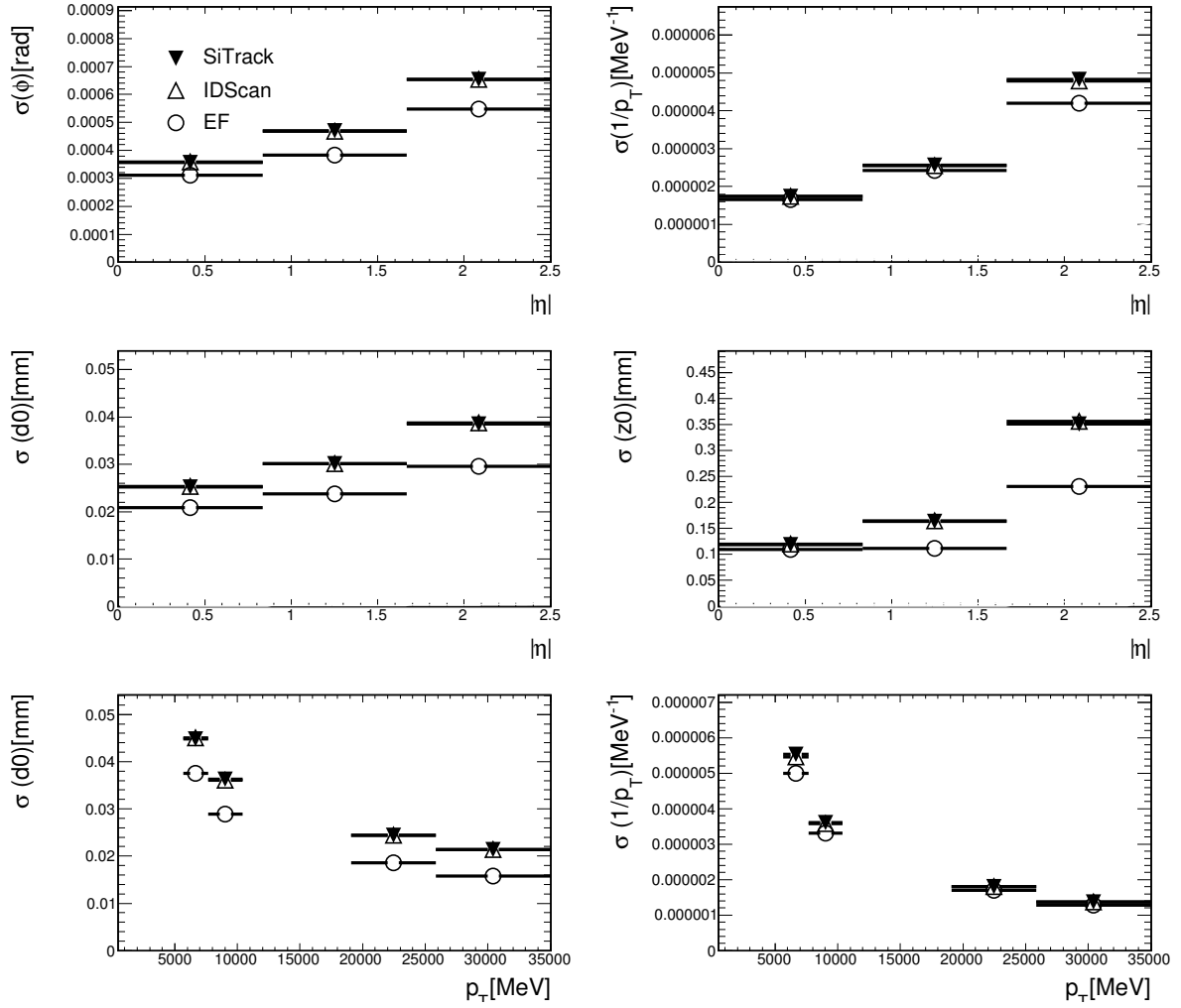


Figure 5: Track parameter resolutions for single muons as a function of η and p_T for SiTrack (full triangles), IDScan (empty triangles) and EF tracking (empty circles).

4.5 Results with jets

The tracking performance has also been evaluated using a sample of b -jets produced in the decay of a Higgs boson (mass 120 GeV) produced in association with a leptonically-decaying W boson. This is the benchmark sample for b -tagging selection. Only tracks formed with at least four space points are considered.

Figure 6 shows the reconstruction efficiency and the fake fraction as a function of η and p_T for SiTrack, IDScan and EF tracking algorithms. Figure 7 summarizes, for the same algorithms, the track parameter resolutions. Not surprisingly, the efficiency for this sample is a bit lower (80 to 90 %) than the single electron and single muon samples, and has a bit higher fake rates. Resolutions are comparable or slightly worse.

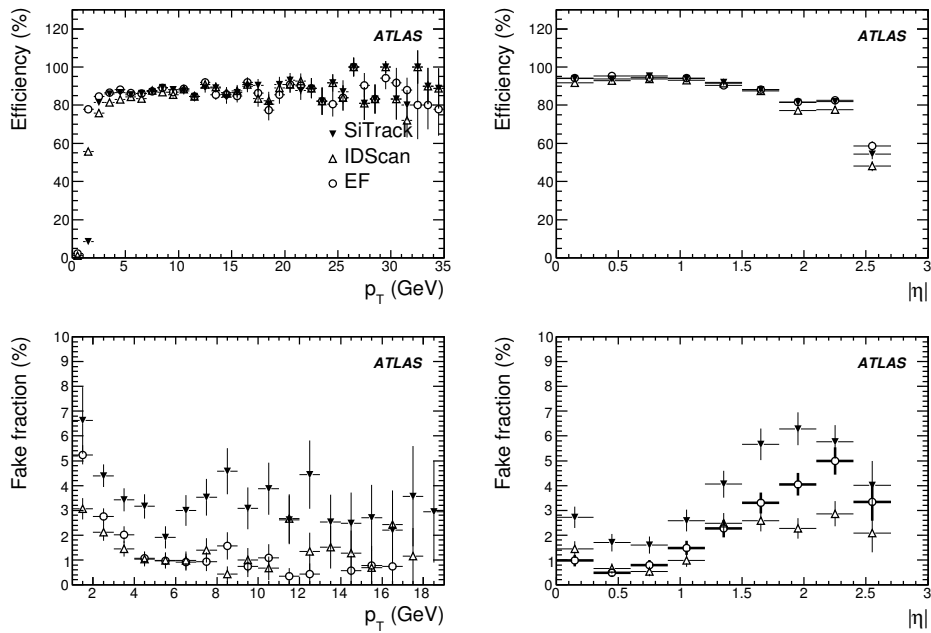


Figure 6: Reconstruction efficiency (top) for tracks in b -jets yielded by Higgs decay as functions of p_T and η for SiTrack (full triangles), IDScan (empty triangles) and EF tracking (empty circles). The efficiency as a function of η is computed for tracks with $p_T > 3$ GeV. Bottom plots show fake fraction as a function of the reconstructed p_T and η .

4.6 Results with π and K in B-physics events

Using B_s decay to $\phi\pi \rightarrow KK\pi$ the reconstruction efficiency for both kaons and pions have been evaluated.

Figure 8 shows the reconstruction efficiency as a function of η and p_T for SiTrack, IDScan and EF tracking algorithms. The efficiency is typically close to 95% dropping somewhat at low p_T or high η .

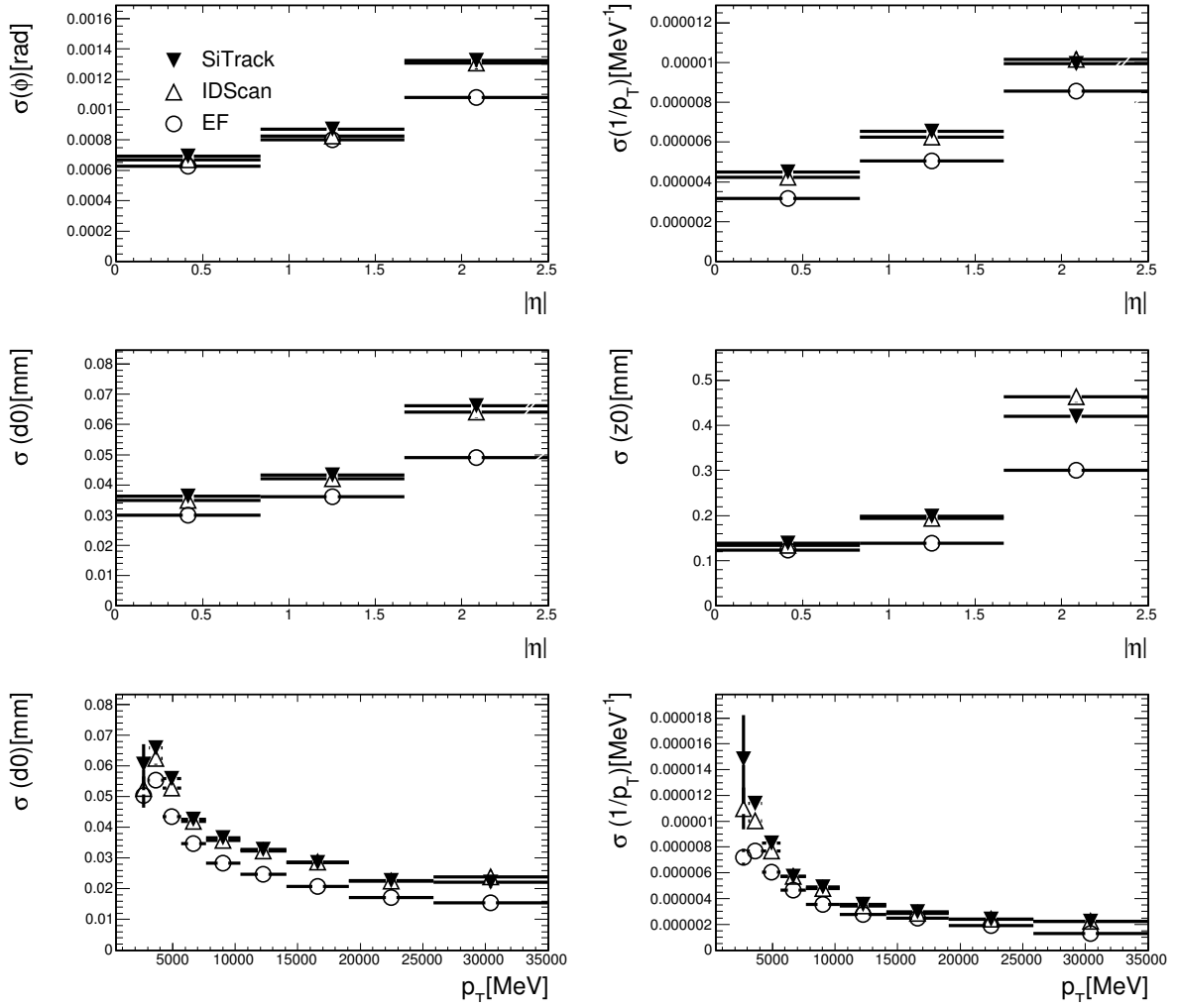


Figure 7: Track parameter resolutions for tracks in b -jets yielded by Higgs decay as a function of η and p_T for SiTrack (full triangles), IDScan (empty triangles) and EF tracking (empty circles).

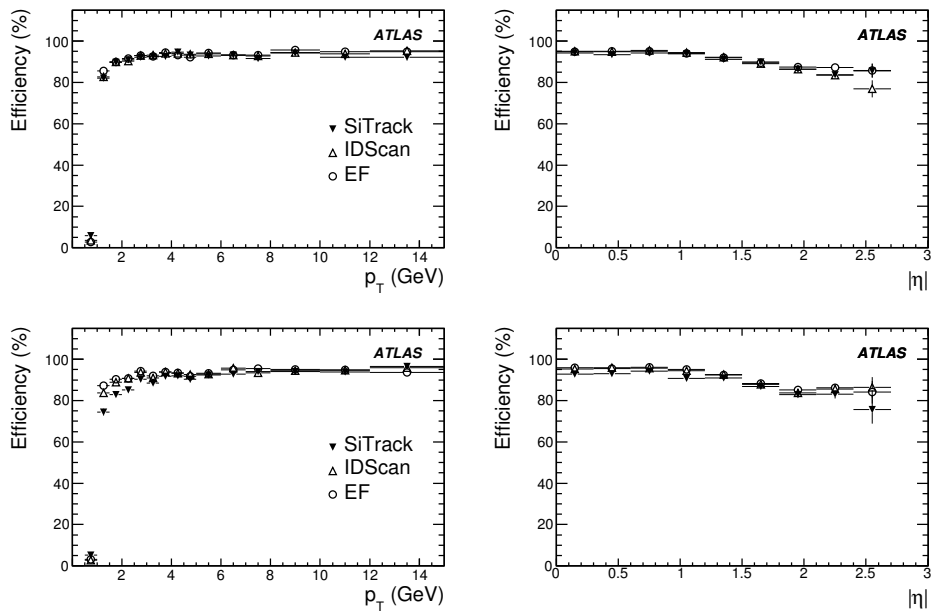


Figure 8: Reconstruction efficiency for kaons (top plot) and pions (bottom plot) yielded from a decay of the B_s meson to $\phi\pi \rightarrow KK\pi$ (sample 16701 with misaligned geometry), as functions of p_T and η for SiTrack (full triangles), IDScan (empty triangles) and EF tracking (empty circles).

4.7 Comparison between the EF and the offline tracking performance

Since the track reconstruction in the EF is based on the same software as used for offline reconstruction, the EF tracking performance was studied under conditions equivalent to those used offline. This study was done using release 13.0.30.4 with perfectly aligned detector geometry.

To factorize out the bare EF performance, fake L1 RoIs were produced and passed through the L2 to seed the EF tracking, based on the MC truth information from particles with a $p_T > 1$ GeV and a $|\eta| < 3$.

The same selection as normally used for offline studies was applied, where only particles within $|\eta| < 2.5$, $|d_0| < 2$ mm and $|z_0 - z_v| \times \sin \theta < 10$ mm were taken into account. The accepted tracks had to pass the requirement that at least 80% of their hits were caused by the matched MC particle as well as the quality requirement of having at least 7 hits in the Pixel or SCT detectors together.

Figure 9 (left) shows the reconstruction efficiency as a function of η , for single electrons and muons with a $p_T = 5$ or 100 GeV. The efficiency is defined as the fraction of particles that produce an accepted track and the average efficiencies based on all electrons (muons) were found to be $83.9 \pm 0.2\%$ (99.50 ± 0.03) for a $p_T = 5$ GeV and $92.9 \pm 0.1\%$ (99.52 ± 0.03) for a $p_T = 100$ GeV. Figure 9 (right) together with Fig. 10 shows the muon resolution of $1/p_T$, ϕ and d_0 for accepted tracks as functions of η , where the inverse p_T resolution is scaled by the p_T value for easier comparison. In accordance with the offline studies, the resolution is determined from the root-mean-squared of the tracks within a region containing 99.7% of the distributions, i.e. within 3 standard deviations from the mean of a Gaussian distribution. All results are shown together with the offline results presented in [1], which is represented in the plots by the superimposed lines. The results agree well, however, small deviations are seen due to slightly different software setup with respect to Ref. [1].

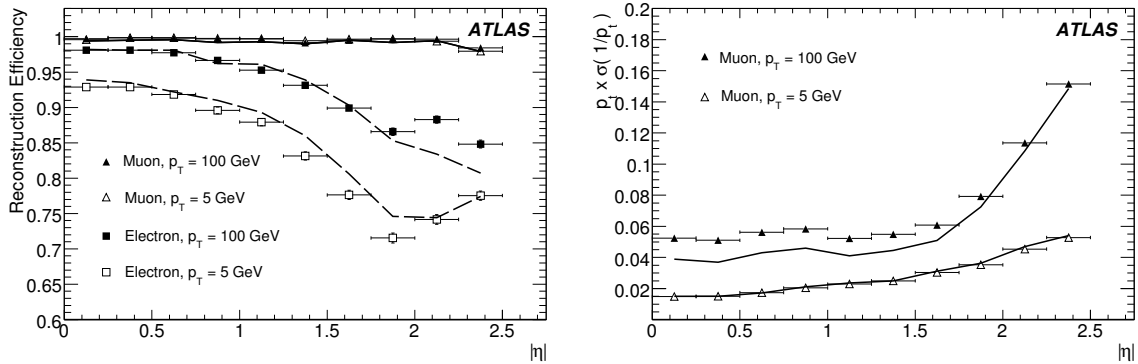


Figure 9: Reconstruction efficiency (left) and scaled $1/p_T$ resolution (right) as functions of $|\eta|$. The superimposed lines represent results from the offline track reconstruction.

5 Summary and conclusions

We have reviewed the track reconstruction algorithms used at the L2 and Event Filter stages of the High Level Trigger of ATLAS.

The algorithms performance and timing have been studied running on simulated data different chains of trigger selection.

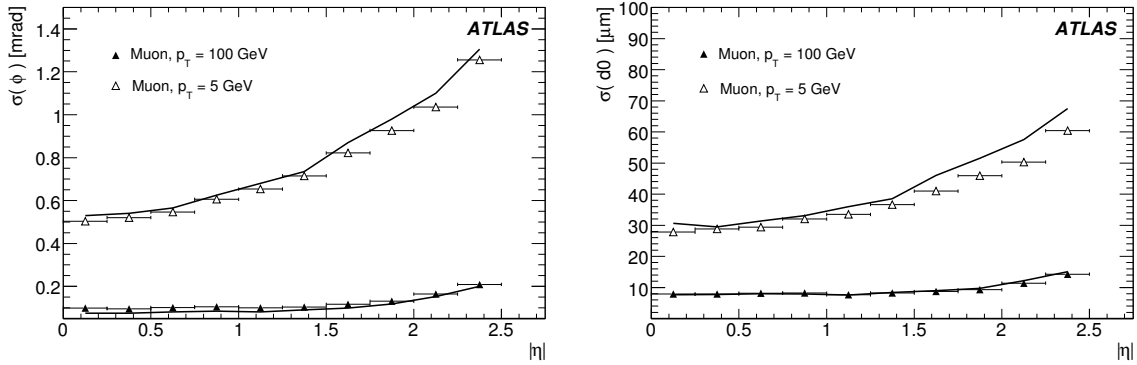


Figure 10: ϕ (left) and d_0 (right) resolution as functions of $|\eta|$. The superimposed lines represent results from the offline track reconstruction.

References

- [1] ATLAS Collaboration, The Atlas Experiment at the CERN Large Hadron Collider.
- [2] I. Gavrilenko, Description of Global Pattern Recognition Program (XKALMAN), ATL-INDET-97-165 (1997).
- [3] D. Emeliyanov, Nucl. Instrum. Meth. **A566** (2006) 50–53.
- [4] D. Emeliyanov, A fast vertex fitting algorithm for ATLAS Level 2 Trigger, XI International Workshop on Advanced Computing and Analysis Techniques in Physics Research ACAT07 (2007).
- [5] T. Cornelissen et al., Concepts, Design and Implementation of the ATLAS New Tracking, CERN-ATL-COM-SOFT-2007-002 (2007).
- [6] P.Billoir, S.Qian, Nucl. Instrum. Meth. **A311** (1992) 139.
- [7] R. Fruhwirth, Nucl. Instrum. Meth. **A262** (1987) 444–450.
- [8] V. Kartvelishvili, Electron bremsstrahlung recovery in ATLAS, Proceedings of the 10th Topical Seminar on Innovative Particle and Radiation Detectors, IPRD06 (2006).
- [9] R. Fruhwirth et al., Nucl. Instrum. Meth. **A502** (2003) 702–704.
- [10] R. Fruhwirth, A. Strandlie, Track finding and fitting with the Gaussian-sum Filter, CHEP (1998).
- [11] R. Fruhwirth et al., J. Phys. **G29** (2003) 561–574.
- [12] L. Bugge, J. Myrheim, Nucl. Instrum. Meth. **179** (1981) 365–381.