Track reconstruction in the LHC experiments

A. Strandlie[a,b]
for the ALICE, ATLAS, CMS and LHCb Collaborations

[a] *Department of Engineering, Gjøvik University College, Gjøvik, Norway*
[b] *Department of Physics, University of Oslo, Oslo, Norway*

Abstract
This paper is an overview of track reconstruction strategies and algorithms in the four LHC experiments. Similarities and differences between the experiments are highlighted, and a description of the strategies of each individual experiment is given. The emphasis is on algorithms analyzing data from main tracking systems, software technicalities and tracking in muon systems are briefly mentioned also. Finally, some examples of recent algorithmic developments implemented in the LHC experiments are given.

10.1 Introduction

Track reconstruction is traditionally divided into two separate subtasks, track finding and track fitting. Track finding deals with the division of a set of measurements in a tracking detector into subsets, each subset containing measurements believed to originate from the same particle. The track fit procedure starts out with the measurements inside one subset as provided by the track finder and aims to optimally estimate a set of track parameters from the information from the measurements. The track fit also evaluates the quality of the track candidate and decides whether it eventually should be accepted as a real particle track.

The traditional procedure described above was to a large extent followed from the beginning of the era of experimental high-energy physics up to the mid 1980s. A multitude of different track finding strategies existed, depending on the specific challenges of the various experiments, and usually the tracks were subsequently fitted by a global least-squares technique [1]. With the invention of the Kalman filter [2, 3], boundaries between track finding and track fitting became more fuzzy. Due to the recursive structure of the Kalman filter, it was soon realized that the algorithm also could be used for track finding [4]. As originally proposed by Billoir, the full knowledge of the track parameters at a detector layer—resulting from a Kalman fit using all measurements already included in the track candidate—is used in the search for compatible measurements in the next detector layer. The measurement closest to the predicted track is included in the track candidate and the rest discarded. The most advanced and popular approach today is the combinatorial Kalman filter (CKF) [5]. The CKF works by building up a combinatorial tree of track candidates, creating new branches each time several measurements in a detector layer are compatible with the prediction from the previous layer. The tree is trimmed by requiring branches to pass a set of quality criteria, e.g., having a value of the track χ^2 below some cut value and not traversing too many detector layers without finding any compatible measurements. An example of this procedure is shown in Fig. 10.1.

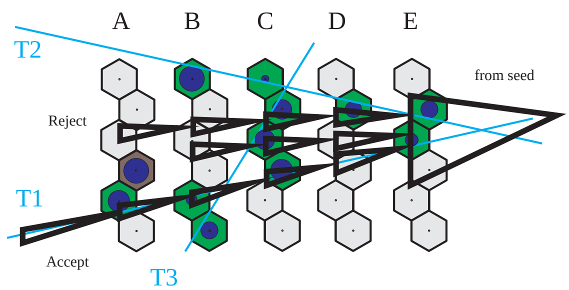

Fig. 10.1: An example of how the CKF works, taken from the original experimental context (the HERA-B experiment at DESY) of the development of the algorithm [5]

Developments having taken place after the Kalman filter can to a large extent be regarded as extensions or generalizations of the Kalman filter. Two main classes of algorithms of this kind exist: Gaussian-sum filters [6] and adaptive methods [7]. Gaussian-sum filters model non-Gaussian effects by Gaussian mixtures, leading to an approach where several weighted Kalman filters are propagated in parallel. Applications up to now include the treatment of bremsstrahlung energy loss of electrons [8, 9], the treatment of non-Gaussian tails of multiple Coulomb scattering [10,11] and the discrimination between different types of material in track reconstruction [12]. Adaptive methods were originally developed for resolving ambiguous and noisy measurements in track fitting but have found applications also in resolving ambiguities in vertex reconstruction [13].

10.2 Overall strategies

All LHC experiments have implemented several tracking strategies, and it seems to be the consensus that there is no single algorithm optimal for all use cases. A typical LHC experiment scenario is to have one default approach as well as various alternative approaches, e.g., second-pass track finding, track fitting in dense jets and some kind of special treatment of electrons.

The following overall decomposition seems to be valid for all experiments:

- Seed generation
- Local track finding (trajectory building) starting from a seed
- Track fitting
- Post-processing, i.e., refitting, ambiguity resolution etc.

Seed generation

Track finding is very often initiated in some part of the tracking detector by creating a *seed*. A seed is typically a few measurements (sometimes including a vertex constraint) plus an initial set of track parameters. The actual strategy of seed generation, however, differs from experiment to experiment:

- **ALICE** generates seeds in the outer part of their Time Projection Chamber (TPC), but there is also an alternative starting in the Inner Tracking System (ITS) quite close to the beam.
- **ATLAS** generates seeds in the inner part of the Inner Detector. An alternative procedure starts out in the Transition Radiation Tracker (TRT).
- **CMS** starts out in the inner part of the tracker. A recent alternative uses measurements also at the outside for generating seeds.
- **LHCb** generates seeds close to the beam in the Vertex Locator (VELO). An alternative approach starts in the T-stations further out.

Local track finding starting from seed

The common denominator for local track finding is the CKF. This approach is the default for track finding in all experiments except LHCb, which bases the track finding on a search for peaks in a histogram of distances from measurements to a parameterized trajectory.

Global approaches are more or less absent, except e.g.,

- **ALICE**, which has implemented a Hough transform in the TPC. Also, a Hopfield neural network is used for stand-alone track finding in the ITS.
- **ATLAS**, which uses a Hough transform in the TRT.
- **CMS**, which tried out and abandoned a Hopfield net several years ago.

None of the approaches above are default.

Track fitting

The Kalman filter is by far the most common track fitting algorithm in all LHC experiments. A global least-squares fit is still used as an alternative in the ATLAS Inner Detector and as default in the ATLAS muon system. Generalizations of the Kalman filter are also used in ATLAS and CMS, in particular the Deterministic Annealing Filter (DAF) [7] for high-luminosity track fitting in the ATLAS TRT [14] and for track fitting in dense jets in CMS [15]. The GSF is used for electron track fitting in both experiments [8, 9].

Post-processing

Some examples (not exhaustive) follow below:

- **CMS** goes through a procedure of *trajectory cleaning*, i.e., removing track candidates which have too many measurements in common.
- **ATLAS** carries out outlier rejection at various stages during the track reconstruction procedure.
- **ALICE+LHCb**: second-pass track finding and refitting.

Muon tracking

In general, muon systems contain more material, have less well-behaved magnetic fields and necessitate propagations over longer distances than the main tracking systems. This creates a need for dedicated propagators, which are able to handle these challenges in a consistent way. The muon tracking strategies in the LHC experiments can briefly be summarized as follows:

- **ALICE+CMS**: combinatorial Kalman filter.
- **ATLAS**: local track finding in regions of interest, followed by a procedure of matching track segments, ending with a global track fit.
- **LHCb**: local track finding, momentum estimated by vertex constraint and measured kink through the magnetic field.

Software

The main programming language is beyond any doubt C++, with some (very few) pieces of residual `FORTRAN77`. An exception is ATLAS, which has implemented important parts of the muon reconstruction software in `FORTRAN90`. There is a clear trend of decomposing the code into components with implementation details hidden behind abstract interfaces. The various reconstruction algorithms therefore mainly differ by putting these basic components together in different ways. As an example, both the ATLAS and CMS muon and inner tracking systems share vital parts of the implementation code. In general, the experiments are moving away from large, monolithic packages, maintainable by only a very few persons, in order to make the implementation strategies more transparent and, hence, making it easier for a larger number of people to contribute to code development.

10.3 Specific strategies for the different experiments

ALICE

An illustration of the ALICE detector can be found in Fig. 10.2.

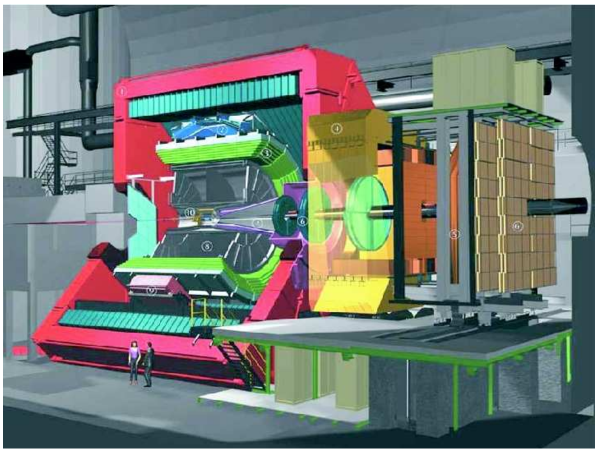

Fig. 10.2: The ALICE detector

ALICE follows a multi-pass procedure for the reconstruction of primary tracks, as illustrated schematically in Fig. 10.3.

Seeds are generated in the outer parts of the TPC, and trajectories are built inwards through the detector with the CKF. Surviving track candidates are extrapolated into the ITS and further built towards the beam pipe area. Tracks are back-propagated through the ITS, the TPC, the Transition Radiation Detector and the Time-Of-Flight detector. Finally, tracks are refitted towards the vertex area. In addition, local track segment finding is performed in each subdetector individually, used to generate additional seeds if these segments are not part of track candidates originating from seeds in the TPC.

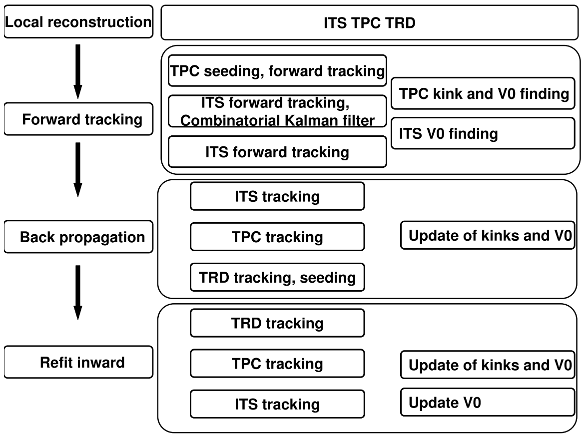

Fig. 10.3: An overview of the different components of the track reconstruction strategy of primary tracks in ALICE

ATLAS

An illustration of the ATLAS detector is shown in Fig. 10.4.

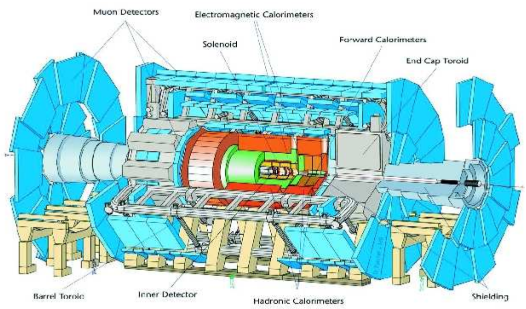

Fig. 10.4: The ATLAS detector

The default strategy for track reconstruction in ATLAS is to generate seeds close to the vertex by combinations of measurements in the pixel detector layers. Starting from the seeds, trajectories are built by the CKF through the silicon strip detector layers. The full track segments from the pixel and silicon detector layers are fitted, bad track candidates removed and ambiguities resolved. Subsequently, tracks are extended into the TRT, initiating the combined track finding and track fitting procedure in the straw tube detector. Finally, extensions are combined with the track segments from the semiconductor layers and the entire tracks refitted.

CMS

The CMS detector is shown in Fig. 10.5.

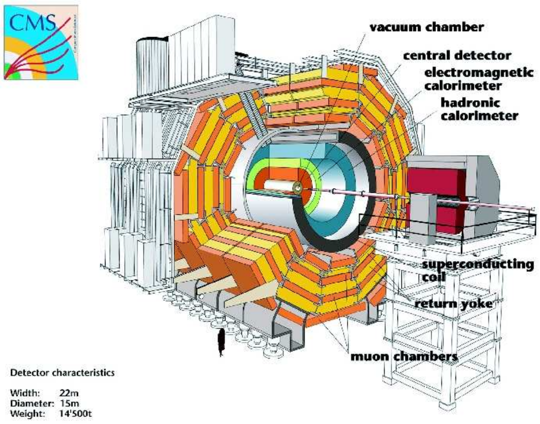

Fig. 10.5: The CMS detector

The CMS strategy is quite similar to the one followed by ATLAS. Seeds are by default generated in the pixel layers close to the beam, and trajectories are built — again with the CKF — outwards from these seeds towards the end of the tracker. Ambiguities, i.e., tracks sharing too many measurements, are then resolved, and finally tracks are fitted.

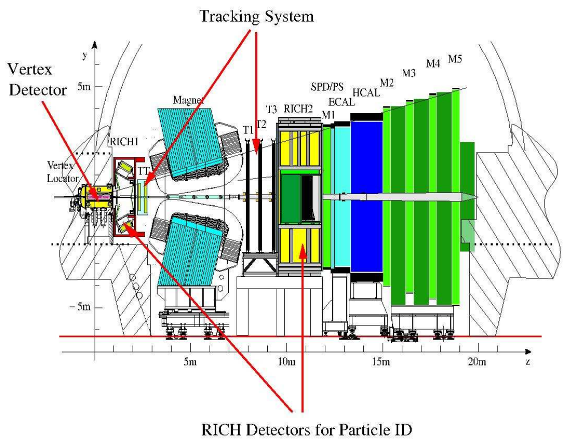

Fig. 10.6: The LHCb detector

LHCb

The LHCb detector is shown in Fig. 10.6. The LHCb follows a multi-pass track finding strategy, using many different algorithms to find many types of tracks. Referring to Fig. 10.7, tracks are first found in the VELO close to the beam. These tracks are extrapolated to the T stations downstream, initiating the finding of track extensions in these stations. Used measurements are cleaned, new seeds are generated in the T stations and matched with remaining VELO tracks. Long-lived particles are looked for by finding tracks in the TT stations and combining these with measurements in the downstream T stations. Low-momentum tracks are found by an upstream search, starting in the TT stations. Finally, tracks are fitted and clones are killed.

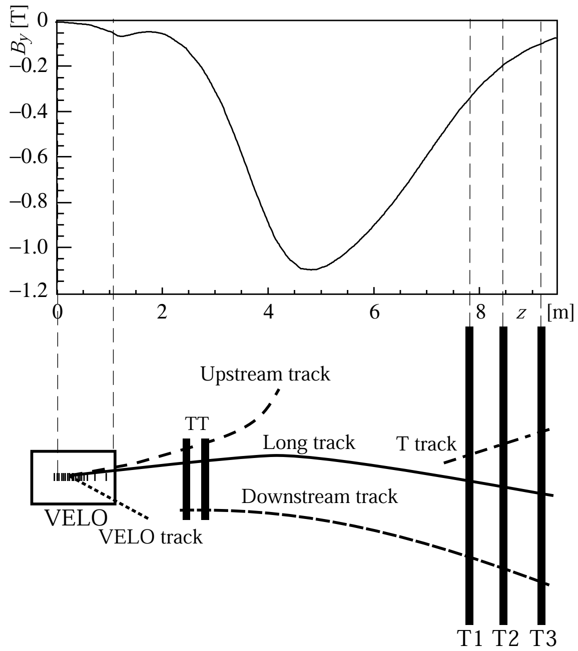

Fig. 10.7: Examples of different types of tracks in the LHCb detector

10.4 Tracking beyond the Kalman filter

Track reconstruction algorithms developed after the Kalman filter have been implemented in ATLAS and CMS. Recent examples are the Deterministic Annealing Filter, intended for track fitting at high luminosity in the ATLAS TRT [14] , and the Gaussian-sum filter for electron track reconstruction in the ATLAS Inner Detector [9].

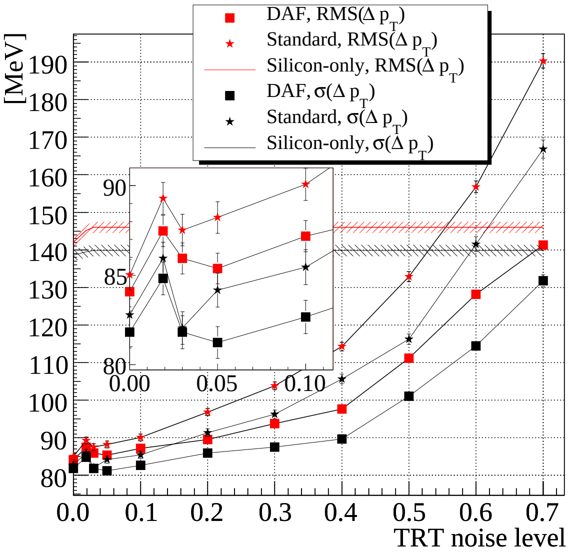

Fig. 10.8: Transverse momentum resolution for 10 GeV/c muon tracks in the ATLAS Inner Detector as a function of noise level in the TRT. The figure is taken from [14].

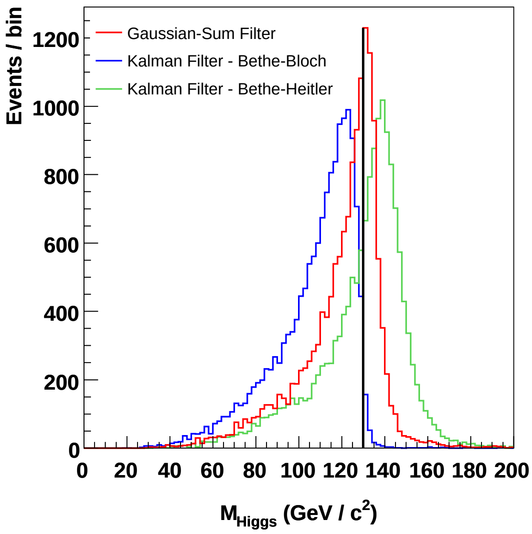

Fig. 10.9: Invariant mass of four reconstructed electrons and positrons, coming from the decay of a simulated Higgs boson of mass 130 GeV/c^2. The figure is taken from [9].

In Fig. 10.8, the p_t resolution for 10 GeV/c muon tracks, simulated by the ATLAS fast simulation program [16], as a function of the noise level in the TRT is shown. In particular, at high noise levels it can be seen that the DAF is more precise than the default algorithm.

In Fig. 10.9, the invariant mass of four reconstructed electrons and positrons, coming from the decay of a simulated Higgs boson of mass 130 GeV/c^2, is shown. The peak of the invariant mass distribution from the GSF agrees very well with the generated Higgs mass. The GSF also creates a distribution with a more narrow core than the Kalman filter does.

10.5 Conclusions

This paper constitutes an overview of the different strategies for track reconstruction in the LHC experiments. There are several common strategies across the experiments but also differences, due to differences in e.g., detector designs and manpower situations.

In an overall perspective, significant changes have taken place since the LEP era. At the beginning of LEP, the Kalman filter was new and exotic, while traditional techniques based on global least-squares estimation played a major role. Now, shortly before LHC start-up, the Kalman filter has taken over as the common choice of algorithm for track reconstruction, while some examples of more recent developments are starting to show up.

Acknowledgements

This paper would have been impossible to write without information and support from Jochen Schieck, Markus Elsing, Teddy Todorov, Thomas Speer, Marian Ivanov, Matt Needham, Gerhard Raven, Jeroen van Tilburg, Tom Atkinson, Sebastian Fleischmann, Wolfgang Liebig, and Rainer Mankel.

References

[1] R. Frühwirth *et al.*, *Data analysis techniques for high-energy physics* (Cambridge University Press, Cambridge, 2000).

[2] P. Billoir, *Nucl. Instrum. Methods* **225** (1984) 352.

[3] R. Frühwirth, *Nucl. Instrum. Methods* **A262** (1987) 444.

[4] P. Billoir, *Comput. Phys. Commun.* **57** (1989) 390.

[5] R. Mankel, *Nucl. Instrum. Methods* **A395** (1997) 169.

[6] R. Frühwirth, *Comput. Phys. Commun.* **100** (1997) 1.

[7] R. Frühwirth and A. Strandlie, *Comput. Phys. Commun.* **120** (1999) 197.

[8] W. Adam, R. Frühwirth, A. Strandlie and T. Todorov, *Journ. Phys. G: Nucl. Part. Phys.* **31** (2005) N9.

[9] T. Atkinson, PhD thesis, University of Melbourne, Australia (2006).

[10] R. Frühwirth and M. Regler, *Nucl. Instrum. Methods* **A456** (2001) 369.

[11] A. Strandlie and J. Wroldsen, *Nucl. Instrum. Methods* **A559** (2006) 158.

[12] A. Strandlie and R. Frühwirth, *IEEE Trans. Nucl. Sci.* **53** (2006) 3842.

[13] T. Speer, R. Frühwirth, P. Vanlaer *et al.*, *Nucl. Instrum. Methods* **A566** (2006) 149.

[14] S. Fleischmann, MSc thesis, University of Wuppertal, Germany (2006).

[15] M. Winkler, CMS Thesis 2002-015/Tracking, PhD thesis, Vienna University of Technology and CERN (2002).

[16] A. Salzburger *et al.*, *The new fast ATLAS track simulation engine (Fatras)*, Proceedings of CHEP06, Mumbai, India (2006).