PROCEEDINGS OF SCIENCE

Tracking performance in CMS

Valentina Gori[*][†]
Università degli Studi di Firenze
E-mail: `valentina.gori@cern.ch`

The CMS tracker is the largest silicon detector ever built, covering 200 square meters and providing an average of 14 high-precision measurements per track. Tracking is essential for the reconstruction of objects like jets and tau leptons and is widely used also at trigger level as it improves lepton and jet resolution and allows to pre-identify tau leptons and b-jets. Tracking algorithms used in CMS will be described. The resolution and efficiency of the track, vertex, and beam line reconstruction, as measured in data, are compared to the results from simulation.

11th International Conference on Large Scale Applications and Radiation Hardness of Semiconductor Detectors
3-5 July 2013
Auditorium Cassa di Risparmio di Firenze, via Folco Portinari 5, Florence, Italy

---

[*] Speaker.

[†] On behalf of the CMS collaboration.

http://pos.sissa.it/

PoS(RD13)004

1. Introduction

In 2012 data taking, LHC luminosity has exceeded 10^{33} cm^{-2} s^{-1} , with a maximum of about 30 overlapping proton-proton interactions. In these conditions the CMS [1] tracker is crossed at each bunch crossing by about 1000 charged particles and track reconstruction in such a high occupancy environment is very challenging. Tracking algorithms have to ensure high tracking efficiency and a low fraction of fake tracks. In addition the tracking code must run sufficiently fast to be used not only for offline event reconstruction but also for the CMS High Level Trigger. The physics goals of CMS place strong requirements on the performance of tracking. Searches for high mass dilepton resonances require tracks to have good momentum resolution for transverse momenta p_T of up to 1 TeV/c. At the same time, efficient reconstruction of very soft tracks with $p_T < 1$ GeV/c is needed for studies of hadron production rates and to obtain optimum jet energy resolution with Particle Flow technique. In addition, it must be possible to resolve very close tracks, such as those from 3-prong taus decay. Furthermore, excellent impact parameter resolution is needed for a precise measurement of the primary vertex position and for b-jet identification. The CMS tracker was designed with these requirements in mind and the track finding algorithms are designed to fully exploit its capabilities and deliver the desired performance. The software algorithms used to achieve this goal are described in the following and the performance obtained are shown, in terms of tracking efficiency, fake rate and resolution. The performance of tracking and vertexing at the trigger level are also discussed.

2. The CMS tracker detector

The CMS tracker is made of pixel and strip silicon detectors and is immersed in a 3.8 T magnetic field, parallel to the beam line. It features thirteen concentric cilindrical layers (*barrel*) and fourteen circular layers (*endcaps* and *disks*) to make the detector hermetic up to $|\eta| = 2.5$. The inner part is equipped with pixel detectors for a total of 66 millions readout channels (three layers in the barrel region and two disks in each endcap region) which provide high granularity, allowing a low occupancy and high three-dimensional resolution. The outer part of barrel and endcaps are instrumented with silicon strips detectors, with strip pitches ranging between 80 and 180 μm.
The CMS tracker allows to measure transverse momenta with a precision of about 1% at about 100 GeV/c and impact parameters with a resolution of about 10-20 μm in the momentum range 10-100 GeV/c.

3. Track and vertex reconstruction

The CMS tracking software is known as the Combinatorial Track Finder (CTF) [2] based on the Kalman filter method [3].
The collection of reconstructed tracks is produced by multiple iterations of the CTF track reconstruction sequence, in a process called *iterative tracking*. In the early iterations, tracks with relatively high p_T , produced near the interaction region, are reconstructed. After each iteration, hits associated with tracks already found are removed, reducing the combinatorial complexity and thus allowing later iterations to search for lower p_T or highly displaced tracks. For 2012 data reconstruction, the iterative tracking consisted of 6 iterations. Iteration 0 is the source of most of the

tracks and is designed to reconstruct prompt tracks with $p_T > 0.8$ GeV and which have three pixel hits (*Pixel Tracks*). Iteration 1 is used to recover prompt tracks which have hits in only two pixel layers or slightly lower p_T . Iteration 2 is configured to find low p_T prompt tracks. Iterations 3 to 5 are intended to find tracks which originate outside the beamspot and to recover tracks not found by the previous iterations. At the beginning of each iteration, hits associated with high quality tracks found in previous iterations are discarded.
Each iteration proceeds in four steps:

- The seed generation provides initial track candidates using only a few (2 or 3) hits. A seed defines the initial estimate of the trajectory parameters and their uncertainties.
- The track finding is based on a global Kalman filter. It extrapolates the seed trajectories along the expected flight path of a charged particle, searching for additional hits that can be assigned to the track candidate.
- The track fitting is used to provide the best possible estimate of the parameters of each trajectory by means of a Kalman filter and a "smoothing" stage: a second filter is initialized with the result of the first one and is run backward towards the beamline.
- The track selection sets quality flags and discards tracks that fail certain criteria.

The main differences between the 6 iterations lie in the configuration of the seed generation and final track selection steps.

3.1 Seed generation

The trajectory seeds define the starting trajectory parameters and associated uncertainties of potential tracks. Because of the 3.8 T magnetic field inside the tracker, charged particles follow helicoidal trajectories and therefore five parameters are needed to define a starting trajectory. To obtain these five parameters at least 3 hits are required, or 2 hits and a beam constraint. To limit the number of hits combinations, seeds are required to satisfy loose criteria, for example on their minimum transverse momentum and consistency with originating from the proton-proton interaction region.
Seeds are built in the inner part of the tracker and the track candidates are reconstructed outwards. The reason for this approach is that, although the track density is much higher in the innermost region of the tracker, the high granularity of the pixel detector ensures that the average occupancy of the innermost pixel layer is much lower than the average occupancy of the outermost strip layer. Moreover, the pixel layers provide three-dimensional space-point measurements which allow for more constraints and better parameter estimation.

3.2 Track Finding

The track finding module of the CTF algorithm is based on the Kalman filter method.
The filter begins with a coarse estimate of the track parameters provided by the trajectory seed and then builds track candidates by adding hits from successive layers one by one. The information provided at each layer includes the location and uncertainty of any found hit as well as the amount

of material crossed, which is used to estimate the uncertainty arising from multiple Coulomb scattering. The Kalman filter method is implemented in four steps.

The first step, *navigation*, uses the parameters of the track candidate, evaluated at the current layer, to determine which adjacent layer(s) of the tracking detector are intersected by the extrapolated trajectory. The second step is a search for compatible sensors in the layers returned by the navigation step. A sensor is considered compatible with the trajectory if the position at which the latter intercepts the detector surface is no more than a given number (currently three) of standard deviations outside the sensor boundary. The third step forms groups of hits, obtained by collecting all the hits compatible with a certain trajectory. A configurable parameter also allows the addition of an invalid hit to represent the possibility that the particle failed to produce a hit in the detector group, for example, due to detector inefficiency. A χ^2 test is used to check which of the hits are compatible with the extrapolated track trajectory. The fourth and last step is to update the trajectory state. New track candidates are formed from each of the original ones, by adding to them exactly one of the compatible hits from each detector grouping (where this hit may be an invalid hit). The candidate trajectory parameters are then updated at the new detector surface, by combining the information from the hit with the extrapolated track trajectory of the original candidate.

3.3 Track fitting

For each trajectory, the track finding stage results in a collection of hits and an estimate of the track parameters. However, the full information about the trajectory is only available at the last hit of the trajectory and the estimate can be biased by constraints applied during the seeding stage. Therefore the trajectory is refitted using a Kalman filter and smoother.

The Kalman filter is initialized at the location of the innermost hit with the trajectory estimate obtained during seeding. The fit then proceeds in an iterative way through the full list of hits, updating the track trajectory estimate sequentially with each hit. For each valid hit, the hit position estimate is re-evaluated using the current values of the track parameters.

This first filter is complemented by the *smoothing* stage: a second filter is initialized with the result of the first one and is run backwards towards the beam line. The track parameters at the surface associated with any of its hits can then be obtained from the average of the track parameters of these two filters, evaluated on the surface itself.

4. Tracking and vertexing performance

The performance of the CTF tracking algorithm discussed above has been evaluated on single muon and single pion simulated events an on some data samples. Simulated particles and reconstructed tracks are associated for evaluating the tracking efficiency, fake rate and track parameter resolutions. A simulated track is associated to a reconstructed one if at least 75% of the hits assigned to the reconstructed track were produced by the simulated particle. The tracking efficiency is also measured using a data driven technique.

The tracking efficiency is defined as the fraction of simulated charged particles that can be associated with a reconstructed track. In general, it depends not only on the quality of the track finding algorithm, but also on the intrinsic properties of the tracker, such as its geometrical acceptance and material budget. Only in some specific cases, for simulated samples, it is possible to evaluate the instrinsic efficiency of the tracking algorithm.

The fake rate is defined as the fraction of reconstructed tracks that are not associated with any simulated particle.

4.1 Tracking performance in simulated samples

Muons are reconstructed very well in the tracker. They mainly interact with the silicon detector through ionization and, unlike electrons, their energy loss by Bremsstrahlung is generally negligible, except when the muon momentum exceeds about 100 GeV. Therefore, muons usually cross the whole volume of the tracking system, producing detectable hits on several sensitive layers of the apparatus. Finally, muon trajectories are altered exclusively by Coulomb scattering, whose effects are straightforward to include into the Kalman filter formalism. For isolated muons with a transverse momentum between 1 GeV and 100 GeV, the tracking efficiency is higher than 99% over the full η range of the tracker acceptance and does not depend on the transverse momentum (Fig. 1), while the fake rate is completely negligible.

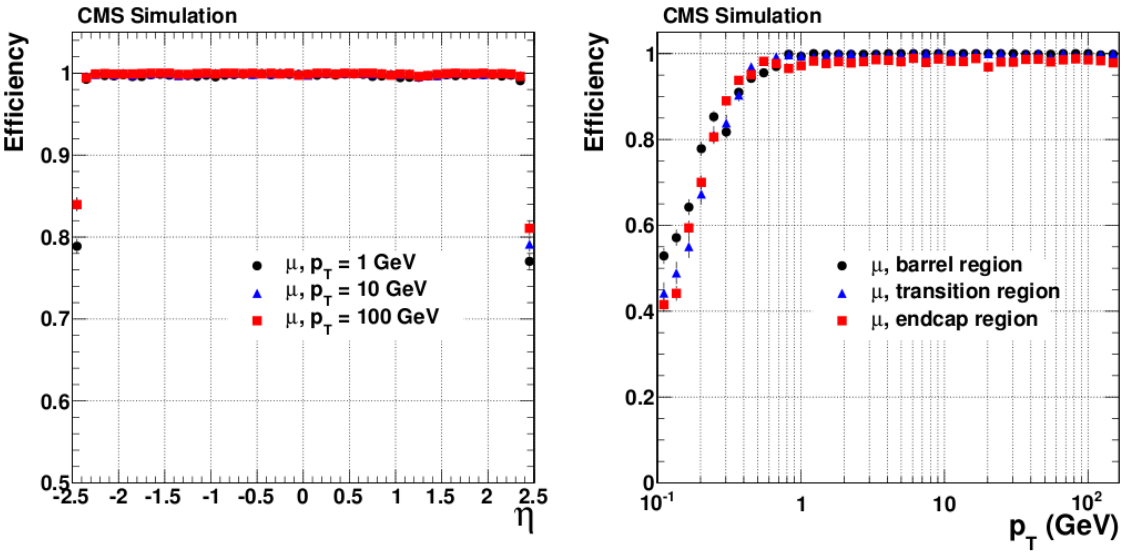

Figure 1: Tracking efficiency for muons, as a function of η and p_T, calculated on simulated samples.

Charged pions undergo multiple scattering and energy loss by ionization, but also elastic and inelastic nuclear interactions with the tracker material as they cross the detector. Nuclear interactions introduce long tails in the distribution of the scattering angle. Inelastic nuclear scattering can lead to the production of secondary particles which, at high momenta, are almost collinear to the pion trajectory: many tracks are produced in a very small cone, so that they cannot be resolved. This is the reason why inelastic nuclear interactions are the main source of tracking inefficiency, especially for the tracker regions where the material budget is larger (for $|\eta| \simeq 1$). Depending on the pseudo-rapidity, up to 20% of the simulated particles are not reconstructed, despite leaving hits in three or more layers of the tracking system (Fig. 2).

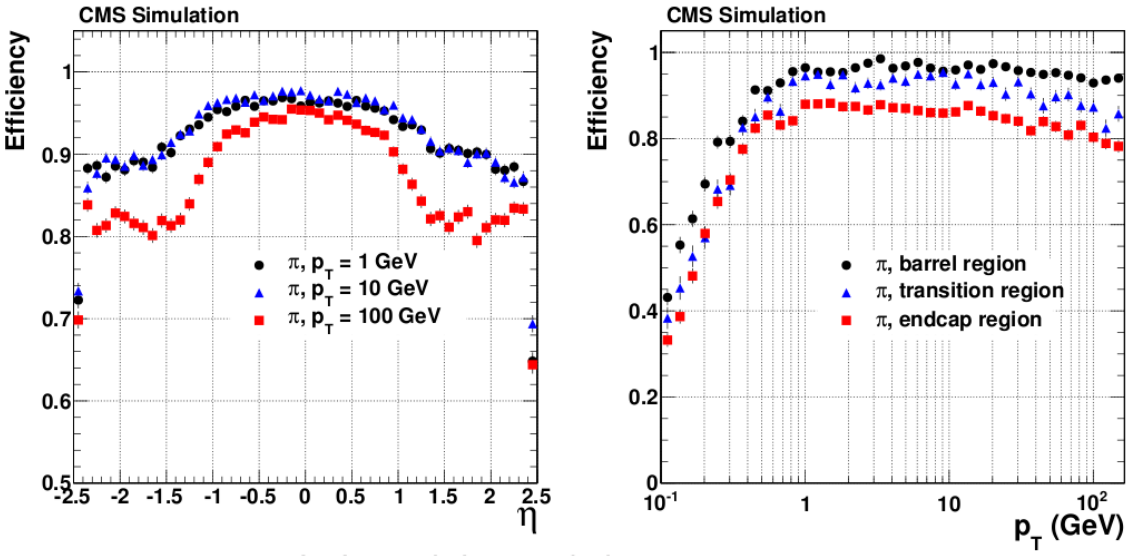

Figure 2: Tracking efficiency for pions, as a function of η and p_T, calculated on simulated samples.

4.2 Tracking performance in data

The *tag-and-probe* method [9] allows to extract the muon tracking efficiency directly from data using resonances.
$Z \to \mu^+\mu^-$ candidates are reconstructed using pairs of oppositely charged particles, identified as muons using the muon chambers [7]. Each candidate must consist of one *tag* muon, that have to be reconstructed in both tracker and muon chambers, and one *probe* muon, reconstructed in the muon chambers only. The invariant mass of each candidate is required to be within the (50,130) GeV range. The tracking efficiency can be estimated from the fraction of the probe muons that can be associated to a reconstructed track in the tracker. A correction must be made for the fact that some of the probe muons are not genuine. This correction is obtained by fitting the dilepton mass spectrum in order to subtract the non-resonant background, (since only genuine dimuons will contribute to the resonance). This must be done separately for candidates in which the probe is or is not associated to a track.
The results of the tag and probe fit are shown in Fig. 3 as a function of the probe η and the number of reconstructed primary vertices in the event. The measured tracking efficiency for muons from Z decay is well above 99% in both data and simulation. Data show a small drop in tracking efficiency, with increasing pile up, which is due to a dynamic inefficiency of the pixel detector, caused by the limited size of the internal buffer of the readout chips, which causes hit loss for events with high occupancy and is not modelled in the simulation.

4.3 Vertexing performance

The goal of the primary vertex (PV) reconstruction is to measure the positions, and the associated uncertainties, of all proton-proton interaction vertices in each event using the available reconstructed tracks. It consists of three steps: selection of the tracks to be used, clustering of the

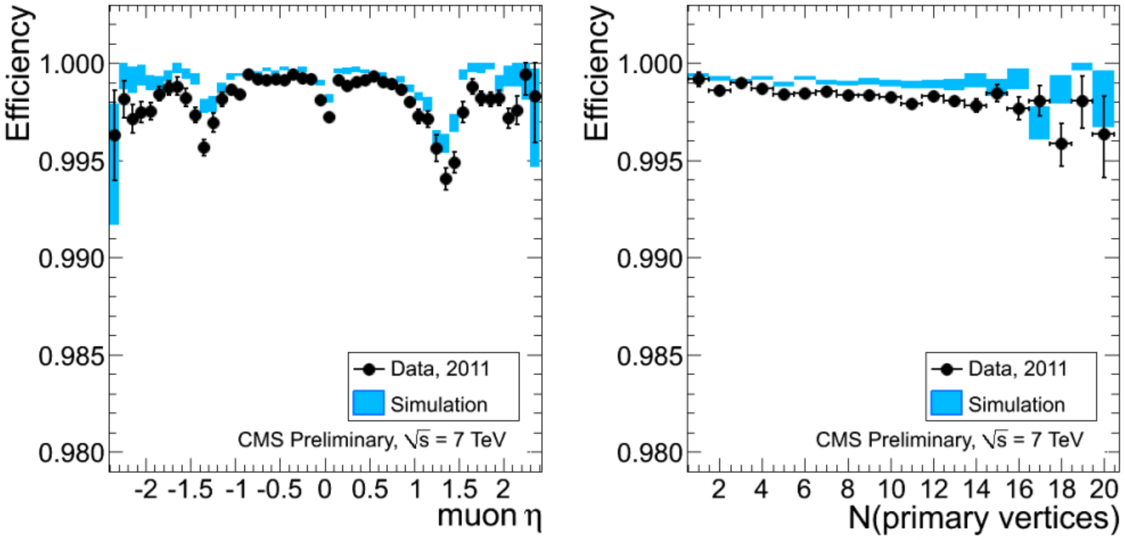

Figure 3: Tracking efficiency for muons measured in $Z \rightarrow \mu^+\mu^-$ events, using the *tag and probe* technique.

tracks (i.e. deciding which ones originate from the same interaction vertex) and fitting the position of each vertex using its associated tracks.

Track selection, which aims to select tracks produced promptly in the primary interaction region, imposes requirements on the maximum allowed transverse impact parameter significance with respect to the beam spot [1], on the number of strip and pixel hits, and on the normalized χ^2 of the track fit. To ensure high reconstruction efficiency, even in minimum bias events, there is no requirement on the minimum allowed track p_T. The selected tracks are then clustered, based on their z coordinates at the point of closest approach to the beamspot. This clustering allows for the possibility of multiple primary interactions in the same LHC bunch crossing. The clustering algorithm must balance the efficiency for resolving nearby vertices in cases of high pileup against the possibility of accidentally splitting a single, genuine interaction vertex into more than one cluster of tracks. The track clustering is performed with a *Deterministic Annealing* (DA) algorithm [4]. Annealing finds the global minimum in a problem with many degrees of freedom analogous to the way a physical system reaches the state of minimal energy through a series of gradual temperature reductions.

To calculate the primary vertex resolution, a data-driven method is used. Tracks belonging to a single vertex are split in two subsets; for each subset, a primary vertex is fitted, independently. The position difference between the two vertices built following this procedure gives the resolution. In Fig. 4 the primary vertex resolution in x and z directions[2] as a function of the number of tracks, for a Minimu Bias[3] data sample (in red) and for a sample enriched in jets, so with a harder p_T

---

[1] The beam spot represents a three-dimensional profile of the luminous region, where the LHC beams collide at CMS. A rough estimate of the beam spot can be given by using pixel vertices, which are recontructed using only Pixel Tracks.

[2] The coordinate frame used in CMS is a right-handed cartesian one, with the x axis pointing towards the LHC centre, the y axis directed upward along the vertical and the z axis along the beam direction with the direction required to complete the right-handed coordinate frame.

[3] A “Minimum Bias” sample is a sample containing collision events with low momentum transfer.

spectrum, in black. A strong dependence of the PV resolution on the number of tracks is shown. For more than about 40 tracks, resolution is about 10-20 μm.

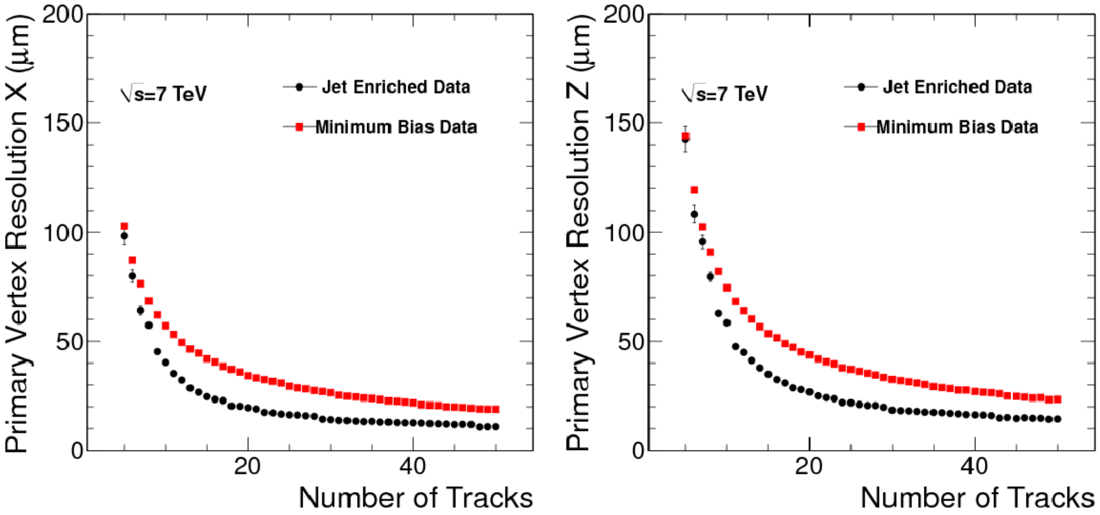

Figure 4: Primary vertex resolution in in x (lefthand side) and z (righthand side) directions, ad a function of the number of tracks, for a Minum Bias data sample.

5. Tracking at HLT

The CMS high-level trigger (HLT, see [5]), implemented in software, runs on events selected by the Level 1 (hardware) trigger [6], at rates of up to 100 kHz, and allows to reduce this rate to about 400 Hz.

The implementation of the track reconstruction in the HLT is important for several reasons. Requiring that the muon/electron candidates reconstructed in the CMS muon chambers or calorimeters [8] should be confirmed by the existence of a corresponding track in the tracker makes it possible to greatly reduce the rate. The background rejection rate of the lepton triggers can be further enhanced by requiring that leptons should be isolated by putting a veto on the presence of (too many) tracks in a cone around the lepton.

It is possible to trigger on jets produced by b-quarks by counting the number of tracks (in the jet) which have a transverse impact parameter which is statistically incompatible with the track originating from the beam line.

The HLT uses the same reconstruction software which is used for offline event reconstruction. However, it has to run much faster, so a dedicated version of the software is achieved by modifying several configuration parameters controlling the tracking algorithms.

Tracking starts with reconstruction of *pixel tracks* from triplets of pixel hits. Pixel tracks are of fundamental importance for the primary vertex reconstruction at HLT. Track reconstruction uses about 20% of the total HLT CPU time. This amount of time is kept low by performing track reconstruction only when necessary and only after some other requirements which reduce the starting rate.

The vertex reconstruction at HLT exploits a simple gap clustering algorithm. All tracks are sorted by the z coordinate of their point of closest approach to the beam spot. Groups of neighbouring tracks separated by a distance in z of at least z_{sep} are assigned to separate vertices, otherwise they are merged, so this algorithm is not an optimal choice for high pile up data taking conditions.

6. Conclusions

The CMS experiment has developed a sophisticated tracking software, based on the Kalman filter technique. It is able to reconstruct tracks over the full rapidity range of the tracker and is sensitive to charged particles with p_T as low as 100 MeV/c and also to those produced as much as 60 cm far from the beam line. For promptly produced charged particles the average tracking efficiency is typically 95 (85)% in the tracker barrel (endcap), with most of the efficiency loss being caused by hadrons undergoing nuclear interactions in the tracker material. The fraction of wrongly reconstructed tracks is at the few percent level. For high p_T tracks (100 GeV), for which multiple scattering is of little importance, the resolution in the central region ($|\eta| < 1$) is approximately 2% in p_T and 10 μm (35 μm) in transverse (longtitudinal) impact parameter. The tracks are used to reconstruct the primary interaction vertices in each event. For the multiparticle vertices typical of interesting physics events, the vertex resolution achieved is 10-12 μm in all three dimensions.

References

[1] CMS Collaboration, J. Inst. 3 (2008) 361.

[2] T. Speer, W. Adam, R. Fruehwirth, A. Strandlie, T. Todorov, M. Winkler, *Track reconstruction in the CMS tracker*, Nuclear Instruments and Methods in Physics Research Section A, Volume 559, Issue 1, p. 143-147

[3] R. Fruhwirth, Nucl.Instrum.Meth. A262 (1987) 444-450.

[4] K. Rose, Proceedings of the IEEE 86 (1998), 11.

[5] CMS Collaboration, *CMS: The Trigger and Data Acquisition Project, Vol. II: Data acquisition and high-level trigger Technical Design Report*, CERN/LHCC 2002-026.

[6] CMS Collaboration, *CMS: The Trigger and Data Acquisition Project, Vol. I: The Level-1 Trigger Technical Design Report*, CMS/LHCC 2000-038, Geneva 2000.

[7] CMS Collaboration, *CMS: The Muon Project Technical Design Report* 1997, CERN/LHCC 97-32, Geneva, 1997. 34

[8] CMS Collaboration, *CMS: The Electromagnetic Calorimeter Technical Design Report*, CERN/LHCC 97-33, Geneva, 1997. 29.

[9] CMS Collaboration, *Measurement of Tracking Efficiency*, CMS Physics Analysis Summary, CMS-PAS-TRK-10-002, (2010).