Linearized Track Fitting

Marco De Mattia

January 6, 2017

Abstract

We present a method to perform a fast estimate of track parameters. We discuss the effect of a realistic detector considering the phase 2 tracker of the CMS experiment and we show how to correct for them. We conclude that a single linear approximation can be utilized over the full barrel region of the CMS tracker improving significantly over previous attempts that relied on segmenting the tracker in thousands of independent regions.

0.1 Introduction

The task of fitting tracks from a multitude of hits in the CMS silicon tracker is achieved offline by the CMS tracking software. The software uses a Kalman Filter approach to refine the information from hits in successive layers. Even in the offline environment, where time is in principle not a constraint, it is not feasible to explore all possible hit combinations. Seeds are instead built using layers that give 3D information (pixels, or double sided strip layers) to guide the search for hits in a narrower window. Even with this approach it is not feasible to perform track fitting within the 12.5 μs latency that will be available in the L1 trigger of CMS for the HL-LHC. New solutions must be found to reduce combinatorics and speed up the track fitting. One possible approach is to utilize associative memories (AM) to quickly group together subsets of the hits in roads that are potentially consistent with tracks. The coarsness of this step leads to multiple fake roads and an additional step is needed to refine the information, remove fakes, and estimate track parameters.

In this document we explore a method of fitting tracks out of a set of hits that is simple enough to be fast and implementable in FPGAs. This approach is based on a linear approximation of the dependence of the track parameters on the hit coordinates. The linear approximation will be acceptable only in a limited region of the phase space, i.e. an expansion of the function around a specific point. This method was utilized in the SVT at CDF and a review can be found in [1] and [4].

Other methods for track fitting at L1 trigger are being explored for CMS, such as Hough Transform and Retina Fit. They are not described in this document.

0.2 Linear Approximation

Each candidate track is represented as a list of n hit coordinates, and can be thought of as a point $\mathbf{x}$ within a set $\mathcal{C} \subset R^n$. This subset is limited by the physical boundaries on the hit coordinates. Of all the points in $\mathcal{C}$ only a subset $\mathcal{T} \subset \mathcal{C}$ can be reasonably consistent with tracks coordinates. The concrete subset $\mathcal{T}$ is defined by specific criteria, such as a χ^2 cut. Track finding is the process of deciding whether a set of coordinates corresponds to a track in $\mathcal{T}$.

Each track is fully defined by a set of $m < n$ parameters $\mathbf{p}$ (e.g. p_T, ϕ, η, d_0, z_0). In the case of perfect detector resolution, the hit coordinates are uniquely determined by the parameters $\mathbf{p}$, and the set $\mathcal{T}$ reduces to an m-dimensional surface contained in $\mathcal{C}$, described by n parametric equations,

$$\mathbf{x} = \mathbf{x}(p_1, \ldots, p_m)\,. \tag{1}$$

By eliminating the parameters equation 1 can be rewritten as $n - m$ constraint equations,

$$f_i(\mathbf{x}) = 0;\ i = 1, \ldots, n - m\,. \tag{2}$$

A set of coordinates $\mathbf{x}$ corresponds to a track in $\mathcal{T}$ if and only if it satisfies the constraints equations.

In the case of finite detector resolution each hit coordinate has an associated uncertainty. The constraint functions f_i then become random variables that take values slightly different from zero. Geometrically, the surface $\mathcal{T}$ acquires some thickness. In this case, there is no general exact solution to the constraint equations.

0.2.1 Linearized Constraints

The starting system of equations is generally not diagonal. However, given that the f_i are random variables we can write their covariance matrix

$$F_{ij} = E[\ (f_i(\mathbf{x}) - E[f_i(\mathbf{x})])(f_j(\mathbf{x}) - E[f_j(\mathbf{x})]\ ]\,, \tag{3}$$

where $E[\ ]$ denotes the expected value. By expanding the constraint functions to first order around a point $\mathbf{x_0}$, and defining $\Delta x_r = x_r - x_{r0}$,

$$f_i(\mathbf{x}) \simeq \frac{\partial f_i}{\partial \mathbf{x}}(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0) = \sum_{r=1}^{n} \frac{\partial f_i}{\partial x_r}(\mathbf{x}_0)\Delta x_r\,, \tag{4}$$

we can express, to first order, the covariance matrix of the constraint functions f as a function of the covariance matrix of the deltas of the coordinates:

$$\begin{aligned} F_{ij} &= E\left[\frac{\partial f_i}{\partial \mathbf{x}}(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0) - E\left[\frac{\partial f_j}{\partial \mathbf{x}}(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0)\right]\right] = \\ &= \sum_{rs} \frac{\partial f_i}{\partial x_r}\frac{\partial f_j}{\partial x_s}\ E\left[(\Delta x_r - E[\Delta x_r])(\Delta x_s - E[\Delta x_s])\right] = \\ &= \sum_{rs} \frac{\partial f_i}{\partial x_r}\frac{\partial f_j}{\partial x_s}\mathrm{Cov}(\Delta x_r, \Delta x_s) \end{aligned} \tag{5}$$

The matrix F_{ij} is a covariance matrix, therefore it is symmetric and it can be diagonalized through a rotation in the space of the f_is. The new coordinates $\tilde{f}_i$

can be scaled to unit variance so that the sum $\sum_i \tilde{f}_i^2$, with $\langle \tilde{f}_i^2 \rangle = 1$, follows a χ^2 distribution as long as the PDFs of the $\tilde{f}_i$s are gaussian. The linear expansion of the diagonalized base of $\tilde{f}_i$s can be written as

$$\tilde{f}_i \simeq \frac{\partial \tilde{f}_i}{\partial \mathbf{x}} \cdot (\mathbf{x} - \mathbf{x}_0) = \mathbf{v_i} \cdot \mathbf{x} + c_i \,. \tag{6}$$

Geometrically, this corresponds to approximating $\mathcal{T}$ with its tangent hyperplane in $\mathbf{x}_0$. The approximation will only work in a small region around $\mathbf{x}_0$, therefore, to apply this method over a large region and retain good accuracy the region must be split in smaller parts where the linearization is performed.

In principle, one can derive the constraint functions analytically given the detector geometry, magnetic field, and material. In practice, however, these functions can be arbitrarily complicated and determining their derivatives from first principles might not be practical. A numerical method can be used to estimate the linearized constraints via the empirical covariance matrix. If we sample directly the phase space of $\mathcal{T}$ we can build this covariance matrix and study its principal components, some of which are the $\mathbf{v_i}$s of equation 6. We can do this by taking a sample of tracks (either from MC or from data), which by construction are sets of hits that belong to $\mathcal{T}$, and by computing the covariance matrix of their hit coordinates

$$\mathrm{M}_{ij} \simeq E[(\Delta x_i - E[\Delta x_i])(\Delta x_j - E[\Delta x_j])] \,. \tag{7}$$

Diagonalizing this matrix, i.e. finding the principal components, will allow to estimate the $\mathbf{v_i}$s. Note that the full correlation matrix of the coordinates contains more information than the constraints, since it has a higher dimensionality (n) than the constraints covariance matrix ($n - m$). In general, what one obtains are n eigenvectors and n corresponding eigenvalues. Of these eigenvalues m will be bigger than the others as their associated variances correspond to the dependence of the track parameters on the hit coordinates. Geometrically, their associated eigenvectors form a base on the hyperplane tangent to the surface $\mathcal{T}$ in $\mathbf{x}_0$. The remaining $n - m$ eigenvalues are smaller and correspond to the linearized constraints. Geometrically, their associated eigenvectors correspond to the coordinates perpendicular to the tangent hyperplane associated to the thickness of the surface $\mathcal{T}$ due to the finite detector resolution (uncertainties on the hit coordinates). We can take the $n - m$ eigenvectors with smaller eigenvalues, scale them to unit variance, and identify them with the $\tilde{f}_i$s. The advantages of this procedure are in its simplicity and in the possibility to perform it directly on data samples, thereby automatically accounting for possible detector misalignements and avoiding uncertainties that might affect the simulation.

With this method we can derive the coefficients $\mathbf{v_i}$ of the linearized constraints through which we can evaluate a χ^2 for each set of hits.

0.2.2 Linearized Track Parameters

The method described in the previous subsection naturally yields a subset of m eigenvectors that are associated to bigger eigenvalues and that can be related to the track parameters. However, these track parameters are not in a readily usable form (e.g. p_T, η, ϕ, d_0 and z_0). We want to find the transformation from the principal components of the hit coordinates to the track parameters

in a usable form. We can invert, at least locally, equation 1 to express the parameters as a function of the hit coordinates

$$p_i = p_i(\mathbf{x})\,. \tag{8}$$

We consider a linear approximation of this equation in the form

$$p_i \simeq \mathbf{w_i} \cdot (\mathbf{x} - \mathbf{x}_0) + p_i(\mathbf{x}_0) \tag{9}$$

or

$$\Delta p_i \simeq \mathbf{w_i} \cdot \mathbf{\Delta x} = \sum_{j=1}^{n} D_{ij} \Delta x_j \tag{10}$$

where we defined $\Delta p_i = p_i - p_i(\mathbf{x}_0)$ and where D_{ij} for $j \in [1, n]$ are the components of $\mathbf{w_i}$. The best coefficients D_{ij} are those that minimize the variance. We can estimate them on a sample of tracks using the least squares method and minimizing the sum of the squares of the deviations

$$\chi^2 = \sum_{ik} \left(\sum_j D_{ij} \Delta x_j^{(k)} - \Delta p_i^{(k)} \right)^2 , \tag{11}$$

where the index (k) denotes iteration on the tracks. Note that, unlike for the linearized constraints coefficients that could be derived also on data, this method can only be applied on MC because we need to know the true values of $\Delta p_i^{(k)}$

Equation 11 presents a global $\chi^2 = \sum_i \chi_i^2$ computed using the deviations for all track parameters. Each parameter is estimated through a separate linear equation and these equations are independent. Therefore, minimizing the χ^2 in equation 11 is equivalent to minimizing the χ_i^2 independently for each track parameter, and we only write it as a single value for convenience. This is important because the uncertainties between different tracks (k) can be assumed to be the same ($\sigma_{(k)} = \sigma$ for all k) and in this case the least squares expression above is equivalent to a χ^2 (if the errors are also gaussian) apart from a multiplicative factor $1/\sigma^2$ which does not affect the result of the minimization. Furthermore, the errors have expectation zero, are uncorrelated and have equal variances as they all have the same σ, therefore from the Gauss-Markov theorem [2], we know that the ordinary least squares estimator is the best linear unbiased estimator of the coefficients. In this case best means giving the lowest variance of the estimate.

To find the coefficients that minimize the χ^2 let us consider the derivative

$$\begin{aligned} \frac{1}{2}\frac{\partial \chi^2}{\partial D_{rs}} &= \sum_{ik} \left(\sum_j D_{ij} \Delta x_j^{(k)} - \Delta p_i^{(k)} \right) \sum_q \delta_{ir} \delta_{qs} \Delta x_q^{(k)} = \\ &= \sum_k \left(\sum_j D_{rj} \Delta x_j^{(k)} - \Delta p_r^{(k)} \right) \Delta x_s^{(k)}\,. \end{aligned} \tag{12}$$

By solving the equation obtained by setting the derivative equal to zero we can find the best coefficients. We have

$$\sum_j D_{rj} \sum_k \Delta x_j^{(k)} \Delta x_s^{(k)} = \sum_k \Delta p_r^{(k)} \Delta x_s^{(k)}\,. \tag{13}$$

Given N tracks, the second half of equation 13 is $N-1$-times the element $C_{rs}^{(p)}$ in the empirical correlation matrix between track parameters and hit coordinates (where x_0 and p_0 are taken as the expected values, or the means) since

$$C_{rs}^{(p)} = \sum_{k=1}^{N} \frac{\Delta p_r^{(k)} \Delta x_s^{(k)}}{N-1}\,, \tag{14}$$

and part of the left term is $N-1$-times the empirical covariance matrix of the hit coordinates $C_{js}^{(v)}$ since

$$C_{js}^{(v)} = \sum_{k=1}^{N} \frac{\Delta x_j^{(k)} \Delta x_s^{(k)}}{N-1}\,. \tag{15}$$

We can finally write

$$\sum_j D_{rj} C_{js}^{(v)} = C_{rs}^{(p)}\,, \tag{16}$$

or in matrix form

$$D \cdot C^{(v)} = C^{(p)} \tag{17}$$

We conclude that the matrix D that performs the linear transformation from the hit coordinates to the track parameters and with coefficients that minimize the variances can be obtained from the covariance matrix of the hit coordinates and from the correlation matrix between track parameters and hit coordinates as

$$D = C^{(p)} \cdot C^{(v)-1}\,. \tag{18}$$

This result is valid independently of the form of the correlation matrix $C^{(v)}$. However, if it is diagonal, i.e. if we are in the base of the principal components, then the inverse matrix $C^{(v)-1}$ is still diagonal with elements $1/\lambda_{ii}$ on the main diagonal, where λ_{ii} are the eigenvalues of $C^{(v)}$. In this case equation 18 simplifies to

$$D_{ij} = \sum_r C_{ij}^{(p)}/\lambda_{jj}\,,\ \text{for } i,j \in [1,n]\,. \tag{19}$$

0.3 Application to the CMS Detector

We apply the method described in the previous section to the silicon strip tracker for the phase 2 upgrade of the CMS detector. In this application we will focus on the barrel region which consists of six layers. For a 3D fit we have three coordinates per layer and with six layers we have a total of 18 coordinates. The barrel region has a cylindrical symmetry, we therefore utilize the cylindrical coordinates (ϕ, R, z) to describe the positions of the hits. The trajectory of a charged particle in a uniform magnetic field can be described by a helix (neglecting energy loss effects) and requires 5 parameters. We take these parameters to be

- c/p_T: the charge over the transverse momentum of the particle.
- phi_0: the *phi* angle of the momentum vector at the point of closest approach to the origin.

- d_0: the transverse impact parameter, defined as the minimum distance between the trajectory and the origin in the transverse plane.
- $cot(\theta)$: the cotangent of the angle between the momentum vector and the z axis (along which the magnetic field is directed) at the point where d_0 is evaluated.
- z_0: the z coordinate of the trajectory at the point where d_0 is evaluated.

For a trajectory passing through the origin in the transverse plane ($d_0 = 0$) we can write the coordinate ϕ of a point on the trajectory as a function of its radius R and of the track parameters c/p_T and ϕ_0 as:

$$\phi = \phi_0 - \arcsin\left(\frac{R}{2\rho}\right) , \tag{20}$$

where

$$\rho = \frac{p_T}{0.003 \cdot B \cdot c} \tag{21}$$

is the curvature radius of the trajectory (radius of the circle deriving from the projection of the trajectory in the transverse plane) and is expressed in cm.

For a perfectly cylindrical detector the radial coordinates do not provide information useful for a principal component analysis of the hit correlations since it is a fixed value for each layer. It is sufficient to consider hits from different layers as separate inputs. In a real detector, however, the hit coordinates for a given layer will not be at the same radius because of two effects: the modules are flat, and they are staggered to allow for overlaps and detector hermeticity. In the following we will show that these effects are not negligible in the upgraded CMS tracker and we will provide a solution that allows to reduce the realistic detector to the ideal case of a perfect cylinder to a very good approximation.

0.3.1 Effect of the Variation of the Module Radius Within a Layer

Let us consider the first order expansion of equation 20[1]

$$\phi \simeq \phi_0 - \frac{R}{2\rho} . \tag{22}$$

If we want to estimate the track parameters ϕ_0 and c/p_T with a linear combination of the ϕ coordinates of the hits in the detector layers we can write

$$\phi_0 = \sum_i A_i \phi_i , \tag{23}$$

$$\frac{c}{p_T} = \sum_i B_i \phi_i , \tag{24}$$

[1]This approximation is valid for small values of the $R/(2\rho)$. For a 2 GeV/c track $\rho \simeq 175$ cm and for the outermost layer the biggest radius is $\simeq 110$ cm. Therefore, $R/(2\rho) \simeq 0.31$. The second order term in the taylor expansion is $(R/(2\rho))^3/6 \simeq 0.005$ which is about 1.6% of the first order term and should be corrected for low p_T tracks. For a 3 GeV/c track the effect is about 0.15%.

and we know that the principal component analysis (PCA) can be used to derive optimal coefficients for these expressions (in the sense that they minimize the χ^2). Substituting equation 22 we obtain

$$\phi_0 \simeq \phi_0 \sum_i A_i + \frac{1}{2\rho} \sum_i A_i R_i \,, \tag{25}$$

$$\frac{c}{p_T} \simeq \phi_0 \sum_i B_i + \frac{1}{2\rho} \sum_i B_i R_i \,, \tag{26}$$

which yield the following constraints on the coefficients

$$\sum_i A_i = 1 \,, \sum_i A_i R_i = 0 \,, \tag{27}$$

$$\sum_i B_i = 0 \,, \sum_i B_i R_i = -\frac{2}{0.003 \cdot B} \,. \tag{28}$$

In a perfectly cylindrical detector the R_i are constant for each layer and it is possible to perfectly satisfy the constraints. In this case the PCA would provide the exact values of the coefficients that minimize the χ^2. In a real detector, however, there are two effects that make it not possible to perfectly satisfy the constraints. The first effect is due to the modules being staggered to provide hermeticity to the detector. In the barrel the staggering is both along ϕ and along z and it causes the radius of the hits to change when moving between adjacent modules. The second effect is due to the flatness of the modules which causes the distance from the hits to the origin to change along the surface of the module. These two effects are not negligible in the planned CMS tracker for the phase 2 upgrade and must be accounted for to achieve the best possible resolution. In the following we will focus our discussion on the p_T resolution. Similar considerations apply to the ϕ_0 and d_0 resolutions.

Figure 1 shows the structure of the transverse projection of a quarter of the barrel of the silicon strip tracker for the phase 2 upgrade of the CMS detector as simulated in the CMS software version CMSSW_6_1_0_SLHC20_patch1. For each of the six layers it is possible to see pairs of modules at the same angle $\phi = \arctan(y/x)$ which correspond to different z positions. It is immediately clear that there is a significant variation in radius among adjacent modules and, to first order, that there are four different sets of radii for each layer. Considering all layers there are 4^6 unique combinations of radii for the full silicon strip tracker. If one were to neglect the variations along z there would be 2^6 combinations.

Figure 2 shows the variation in the ϕ coordinate of the hits in the outermost layer produced by a 2 GeV/c track from the innermost to the outermost module in that layer. This variation is about 2.4%. From equation 20 we can compute the p_T of a track that would produce, on the outermost module, a hit with the same ϕ of the hit produced by the 2 GeV/c track in the innermost module. This relative variation in p_T is approximately independent of p_T and is approximately inversely proportional to the variation of the radius of the hit. The biggest variation is in the innermost layer where it can big as big as about 11% while in the outermost layer it is the smallest and it is about 2.4%.

An additional effect that causes biases and reduces the precision of the measured parameters is the module flatness, which causes a variation of the radius

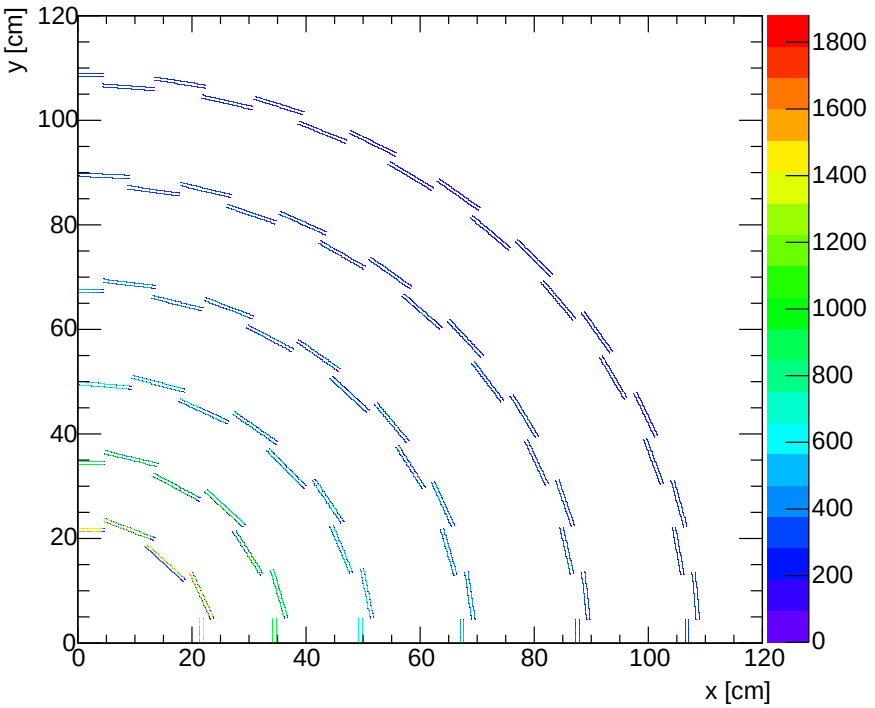

Figure 1: A view of the modules in the transverse projection of a quarter of the barrel of the CMS silicon tracker for the phase 2 upgrade. The geometry utilized is the default in the CMS software version CMSSW_6_1_0_SLHC20_patch1. The plot is obtained by generating single muons and showing the position of the stubs reconstructed by the emulation of the front-end electronic of the strip tracker. The color bar shows the number of entries.

of the hit inside a single module. This effect is most important for the innermost layer, where 16 modules cover the full ϕ angle with each module covering approximately 0.4 radians. Assuming the center of the module to be its closest point to the origin and having a radius of approximately 23 cm, the radius at the edge is approximately 23.5 cm, a variation of about 2%. For a 2 GeV/c track this translates in a similar effect on the p_T of about 2%. For the second innermost layer this effect is reduced to less than 1% by the increase in the number of modules (24) and it is reduced further as the number of modules per layer increases.

The expected offline p_T resolution in the upgraded tracker is about 0.5% for a p_T of 2 GeV/c and better than 2% at high p_T (around 100 GeV/c). It is clear that the effects discussed above would preclude the possibility of reaching this level of precision and they need to be corrected. Simulation studies show that the impact of the module flatness on the p_T resolution is small. This can be understood from the fact that the effect is already smaller than 1% in the second layer and decreases further in the outermost layers. Correcting for it can improve the p_T resolution from 0.6% to about 0.5% (a 10-15% relative increase). The effect of the module staggering, on the other hand, is as big as 11% in the innermost layer and not smaller than 2% in the outermost. Even assigning a bigger weight to the outermost layers the p_T resolution would be limited to about 2% at low p_T (2 GeV/c) and it would be limited to about 3% at high p_T (100 GeV/c) (estimated as the sum in quadrature of the intrinsic resolution of the detector (about 2%) and the effect of module staggering (about 2%)). These numbers were verified with a full simulation of the upgraded CMS tracker using samples of single muons and anti-muons. In what follows we will describe a simple method to correct these effects that allows to achieve close to offline-like resolutions with a linearized fit.

0.3.2 Correction of the ϕ Coordinates

We describe a simple method to correct the ϕ coordinates for each layer for the variation of the radii. For each layer we take the average radius of all the modules as the radius of an ideal layer and we correct all the ϕ coordinates as if the hits were on this ideal layer. From equation 20 it is clear that the shift of the ϕ coordinate due to the change in radius is p_T-dependent. Therefore, we need to know the p_T to be able to perform the correction. If the radius of the ideal layer is R' we can write, to first order, the corrected ϕ' coordinate at the intersection of the track with this ideal layer as a function of the original ϕ, R and ρ as

$$\phi' = \phi + \frac{R - R'}{2\rho} \,. \tag{29}$$

The correction to the ϕ for the change in radius is generally of 1-2%, as discussed in the previous section. Therefore, the value of the c/p_T utilized for the correction can be obtained from a simpler, first order estimate and does not need to be extremely accurate, as will be explained below. To obtain this estimate we perform a first PCA analysis of the ϕ coordinates of the hits in a sample of tracks and derive a set of coefficients to estimate the c/p_T from the ϕ of all the hits for a given track. From what we discussed in the previous section we know that the resolution of this preliminary estimate is about 2% at

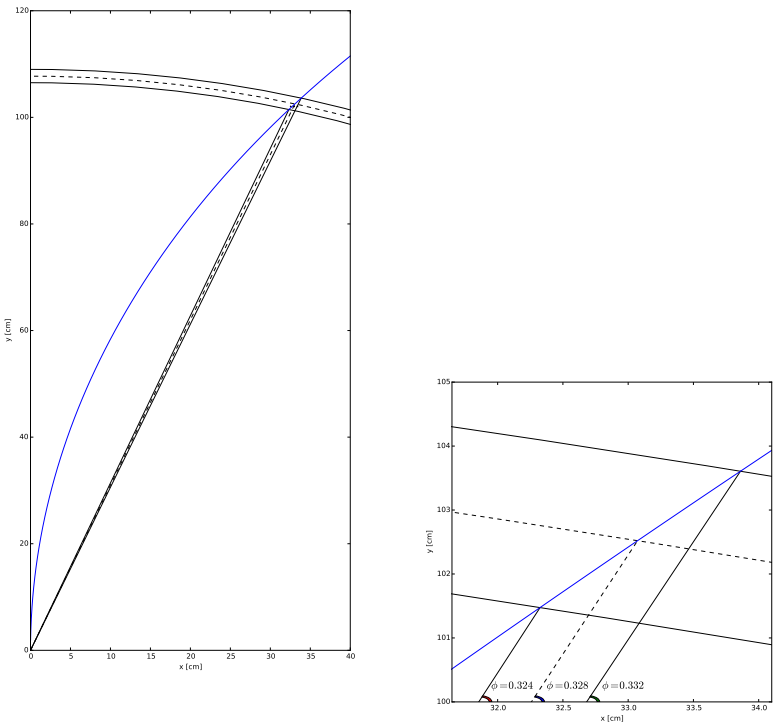

Figure 2: (Left) Variation of the ϕ angle for the hits produced by a 2 GeV/c track crossing the outermost layer of the silicon strip tracker for the phase 2 upgrade of CMS. The internal and external curved lines show the average position of the innermost and outermost modules in the layer, while the dotted curved line shows the average position of all modules in the layer. (Right) Zoomed view of the left figure in the area of the hits. The ϕ angles show the ϕ coordinates of the hits that would be produced, from left to right, in the innermost module, a module in the average radius of all layer modules, and outermost module.

low p_T and order of 3% at higher p_T (around 100 GeV/c). This is enough as a pre-estimate because it allows to perform a correction on an approximately 2% shift of the ϕ coordinates with a precision better than 3%. We obtain a new set of corrected ϕ coordinates which we utilize for a second PCA analysis. The new set of coefficients allows for a much improved estimate of the track parameters. Full detector simulation studies show that it is possible to achieve close-to-offline levels of p_T resolution (0.5% at 2 GeV/c and better than 2% at 100 GeV/c).

To perform the first estimate of the c/p_T it is sufficient to store six coefficients as shown by equation 24 and it would require six multiplications and five additions to compute. Applying the correction to each coordinate would require six more multiplications by the radii and the addition of the result to the ϕ coordinates. Furthermore, performing the correction restores cylindrical symmetry to the detector allowing to use a single set of PCA coefficients to estimate the χ^2 and the final track parameters for the full ϕ and η range in the barrel. This is a distinct advantage over other approaches that consider each possible combination of modules as a separate entity with a dedicated set of coefficients.

For low p_T tracks a first order correction might not be enough since the non-linearity in the arcsin of equation 20 leads to a 1.6% effect for a 2 GeV/c track (and a 0.15% for a 3 GeV/c track). In this case there are two possible alternatives. The first one is to compute a dedicated set of constants for the 2-3 GeV/c p_T range. The initial c/p_T estimate allows to select the correct set of constants for this p_T range and the limited range allows the PCA to optimize the corrections to compensate for the non-linearities. The second possibility is to correct the ϕ coordinates with the second order term in the taylor expansion of the arcsin, which would require a few additional operations per track.

0.3.3 Correction of the z Coordinates

The same approach can be applied to the calculation of the z_0 and $\cot\theta$ parameters by considering the equation expressing the z coordinate of the trajectory as a function of the track parameters and of the radius

$$z = z_0 + 2\rho \cdot \arcsin\left(\frac{R}{2\rho}\right)\cot\theta\,, \tag{30}$$

and noting that its first order expansion of the arcsin around $R/(2\rho) \simeq 0$ is[2]

$$z \simeq z_0 + R \cdot \cot\theta\,. \tag{31}$$

The effect of the variation of the radii on the $\cot\theta$ is similar to the effect on p_T in the case of the ϕ coordiantes. The effect is maximum in the outermost layer where it is about 11% and minimum in the innermost layer where it is about 2.5%. Again, this is a non-negligible effect and should be corrected to achieve a good performance with the linearized track fit. The same approach used for the ϕ coordinates can be applied to the z coordinates by building a first

[2]The second order term is $\frac{1}{24}\frac{R^3}{\rho^2}\cot\theta$. The effect of this term is biggest for the lowest p_T and for the larger radius. For a 2 GeV/c track the effect is 1.5% at the outermost layer and 0.07% at the innermost layer. For a 3 GeV/c track the effect is 0.6% at the outermost layer and 0.03% at the inmermost layer.

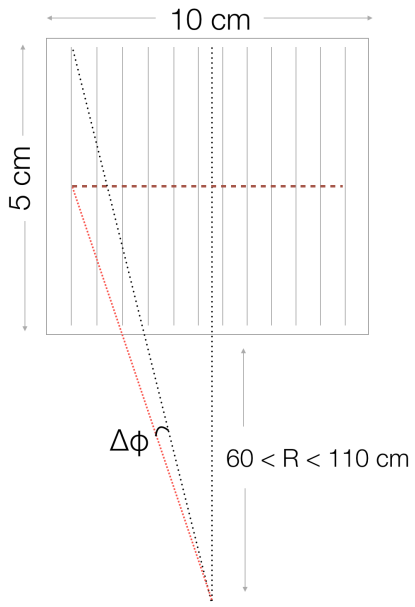

Figure 3: Variation of the ϕ coordinate along the strip length due to the non-radial arrangement of the strips. This sizes shown are for strips in 2S modules.

order estimate of $\cot\theta$ utilizing only the z coordinates and using it to compute corrected-z coordinates referred to the ideal layers.

0.4 The Disks

The geometry of the CMS tracker is that of concentric cylindrical layers of modules in the so-called barrel and of modules arranged in disks in the so-called endcaps. So far we focused our discussion on the barrel component of the tracker. However, the algorithm described here can be applied with not change to a disk-based geometry, since it will perform an intermediate transformation that changes that geometry to a smooth barrel. The modules used in both barrel and endcaps have strip that are parallel to each other. In the endcaps the modules are arranged so that the strips roughly point back to the z axis. However, since they are parallel to each other, if one strip points to the z axis, all others will not. Since we do not have information on the position of the charge deposit along the strip, each stub on each strip in the disks is assigned the R coordinate of the center of the strip. Figure 3 illustrates how the non-radial strips arrangement causes the correct R of the stub to deviate from the assigned R if the particle did not pass through the center of the strip. This deviation is roughly proportional to the distance from the center of the strip. As a consequence, the ϕ coordinate is also biased as shown in the figure.

While this effect is present for every strip in the disks that does not point back to the z axis, the effect is most important for 2S modules where the strips are 5 cm long. In the PS modules, the reduced strip length of 2.5 mm makes

the effect negligible.

To correct for this effect we need to have a better estimate of the position of the stub along the strip. Since in the innermost modules are from PS modules that have a much better resolution along R than the 2S modules, we extrapolate the trajectory linearly from the outermost PS module to the 2S modules in the disks. We then correct the ϕ coordinate with (to first order):

$$\Delta\phi \approx \text{pitch} \cdot (\text{strip\_index} - \text{central\_index}) \cdot \frac{R_{extrapolated} - R}{R^2}\,, \qquad (32)$$

where

- pitch = 90 μm;
- strip_index is the index of the strip from 0 to 1015;
- central_index = 507.5. This is the index a strip would have if placed at the center of the module.

The $R_{extrapolated}$ is computed to second order approximation as

$$R_{extrapolated} \approx R + \Delta z \tan\theta + \left(\frac{1}{2\rho}\right)^2 \cdot \left(-\frac{1}{2}R^2 \cdot \Delta z \tan\theta - \frac{1}{2}R \cdot (\Delta z \tan\theta)^2 - \frac{1}{6}(\Delta z \tan\theta)^3\right)\,, \qquad (33)$$

where R is the coordinate of the stub in the outermost PS module and $\Delta z = z_{2S} - z_{PS}$, with z_{PS} the coordinate z of the stub of the outermost PS module and z_{2S} the coordinate of the stub in the 2S module we are extrapolating to.

0.5 Firmware Implementation

The algorithm described in this document was developed to be FPGA friendly. Most of the operations involve scalar products among vectors of numbers or simple additions and multiplications. Furthermore, the memory required for the constants (correlation and transformation matrices and mean values) is significantly smaller than previous approaches. This is because the intermediate transformation step produces a smooth geometry where the linear approximation of the second PCA is very accurate over the full detector. It is therefore possible to utilize a single set of coefficients for each combination of layers and disks over the full detector.

The algorithm was implemented and tested in KCU040 and KCU060 Xilinx ultrascale FPGAs. The firmware implementation utilizes DSPs for the scalar products and for the intermediate correction step. The constants are stored in distributed memories (DRAMs), thus freeing the larger block rams (BRAMs) for other possible uses outside this algorithm. Since each DSP is performing one operation in the scalar product chain (such as always multiplying the first element by its corresponding coefficient), it only needs access to a subset of the coefficients. This justifies and encourages the use of DRAMs instead of BRAMs.

In a concrete implementation we divided the detector in 14 unique possible combinations of six between layers and disks. An example of such combination includes the six barrel layers and a different combination would be the five innermost barrel layers and the first disk. For each of those combinations we store

a different set of constants since the relative resolutions of stubs in each layer are different and therefore the PCA yields a different set of optimal coefficients. In total, when including also combinations of with only 5 stubs out of those 6, we are able to cover the full detector with 89 unique combinations. We decouple the estimates of transverse plane track parameters (p_T, ϕ_0 and possibly d_0) from those of the $R-z$ plane track parameters ($\cot\theta$ and z_0) since we found little correlation between the two after the intermediate correction step. This helps in reducing the size of the coefficient matrices. The decoupling means that we only need to use the six (or five) ϕ coordinates to estimate the transverse plane parameters and the six (or five) z coordinates to estimate the $R-z$ plane track parameters. Furthermore, to optimize the performance for low p_T where energy loss effects become more important, we utilize two independent set of constants for the transverse plane estimates for low p_T and high p_T tracks. This selection is enabled by the knowledge of the first estimate of the track p_T (accurate to a few percent) before we perform the final estimate of the track parameters.

With this setup, each DSP needs to access at most 178 unique coefficients. In the concrete firmware implementation we implemented memories able to store up to 256 unique coefficients so that extra combinations can be included as needed, to cover specific cases (such as some combinations with only 4 stubs). This memory requirement is very small compared to the typical size of modern FPGA memories and can comfortably be implemented with DRAMs.

The firmware implementation contains coefficients for the full detector. The resource utilization is small enough that in principle up to 17 copies can be included in a single KCU060 FPGA, with the major limiting factor being the number of DSPs (162 or less than 6% in a KCU060 FPGA) utilized by each instance. This opens up possibilities for parallelization or time multiplexing in a single FPGA. Up to four independent copies of the linearized track fitter were implemented in a KCU040 FPGA and timing requirements were met with a clock of 500 MHz. Figure 4 shows the structure of the firmware implementation for the transverse plane along with the latency of each step. The structure for the $R-z$ plane component is almost identical except for the lack of the "p_T switch" block. The overall fixed latency is of 39 clock cycles and the design is fully pipelined. At 500 MHz a single instance would be able to process 422 track fits within $1\mu s$. The performance of this firmware implementation was found to be consistent with that of the full floating point simulation. Figure **??** shows an example waveform for a single muon event being processed by the fitter.

0.5.1 Conclusions

A method to compute optimal coefficients for a linear estimate of the track parameters using a principal component analysis was presented. It was shown that the effect of the variation of the radii of modules in a layer of the CMS tracker for the phase 2 upgrade has a non-negligible impact on the accuracy of this linear estimate. A simple method to correct this effect in the barrel was also presented. Overall, a single set of PCA coefficients is enough to provide performance close to the offline reconstruction over the full barrel. Splitting the PCA coefficients in few p_T regions can further improve the performance by reducing the effect of non-linearities at low p_T.

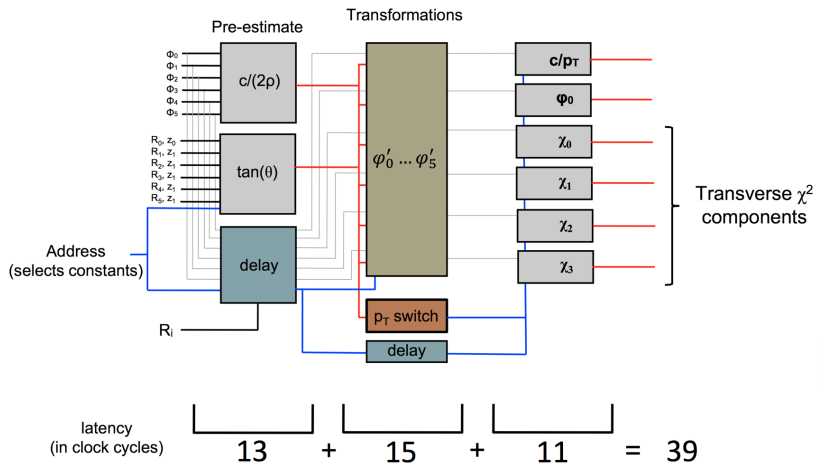

Figure 4: Structure of the firmware implementation for the transverse plane.

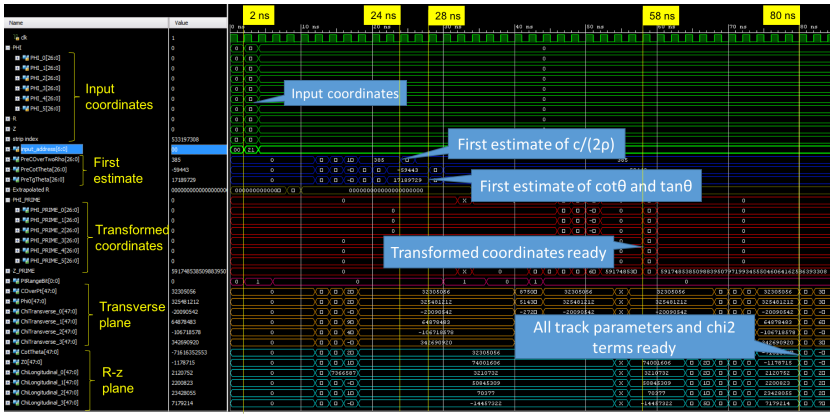

Figure 5: Example waveform.

.1 One-Pass Evaluation of Mean and Covariance

The mean of a random variable x over a sample of size N can be defined as

$$\bar{x}_N = \sum_{i=1}^{N} \frac{x_i}{N}\,, \tag{34}$$

and the covariance for two random variables x and y can be estimated with the empirical covariance matrix in a sample of size N

$$\mathrm{Cov}(x,y) = \sum_{k=1}^{N} \frac{(x_i - \bar{x}_N)(y_i - \bar{y}_N)}{N-1}\,, \tag{35}$$

where the $N-1$ at denominator makes the estimator unbiased given that we evaluate the mean from the same sample.

Evaluating the mean and covariance of a large sample using these formulas is not practical since one would need to know the total size of the sample for computing the mean and the mean itself for computing the covariance. This implies having to iterate over the sample more than once. Furthermore, the formula for the mean is not numerically stable, since the sum at numerator can be arbitrarily big. Iterative formulas for evaluating both mean and covariance will be derived in this section. For a reference to these, and more general formulas which allow to split the calculation in arbitrary number of parts and combine the results, useful for parallelizing the evaluation, see [3].

First of all, let us derive an iterative formula for computing the mean in one pass. From equation 34 we can write

$$\bar{x}_N = \frac{N-1}{N}\sum_{i=1}^{N-1}\frac{x_i}{N-1} + \frac{x_N}{N} = \frac{N-1}{N}\bar{x}_{N-1} + \frac{x_N}{N}\,, \tag{36}$$

or, in a format convenient for C++,

$$\bar{x}_{N-1} \mathrel{+}= \frac{x_N - \bar{x}_{N-1}}{N}\,. \tag{37}$$

We can use the result of equation 36 for deriving the iterative formula for the empirical covariance matrix. It is more terse to derive a recursive formula for $\mathcal{M}(x,y)$ defined as

$$\mathcal{M}_N(x,y) = (N-1)\cdot \mathrm{C}_N(x,y) = \sum_{k=1}^{N}(x_i - \bar{x}_N)(y_i - \bar{y}_N)\,, \tag{38}$$

where C_N denotes that the empirical covariance matrix is evaluated on a sample

of size N. Utilizing equation 36 in equation 38 we have

$$\begin{aligned}
\mathcal{M}_N(x,y) &= \sum_{i=1}^{N-1}(x_i-\bar{x}_N)(y_i-\bar{y}_N)+(x_N-\bar{x}_N)(y_i-\bar{y}_N) = \\
&= \sum_{i=1}^{N-1}\left(x_i-\bar{x}_{N-1}-\frac{x_N-\bar{x}_{N-1}}{N}\right)\left(y_i-\bar{y}_{N-1}-\frac{y_N-\bar{y}_{N-1}}{N}\right)+ \\
&\quad + \frac{N-1}{N}(x_N-\bar{x}_{N-1})(y_N-\bar{y}_N) = \\
&= \mathcal{M}_{N-1}(x,y)+\sum_{i=1}^{N-1}(x_i-\bar{x}_{N-1})\left(\frac{\bar{y}_{N-1}-y_N}{N}\right)+ \\
&\quad + \sum_{i=1}^{N-1}(y_i-\bar{y}_{N-1})\left(\frac{\bar{x}_{N-1}-x_N}{N}\right)+ \\
&\quad + \frac{N-1}{N^2}(x_N-\bar{x}_{N-1})(y_N-\bar{y}_{N-1})+\frac{N-1}{N}(x_N-\bar{x}_{N-1})(y_N-\bar{y}_N)\,.
\end{aligned} \tag{39}$$

We note that

$$\sum_{i=1}^{N-1}(x_i-\bar{x}_{N-1})\left(\frac{\bar{y}_{N-1}-y_N}{N}\right)=0\,, \tag{40}$$

(and the same exchanging x and y) because the first term is

$$\sum_{i=1}^{N-1}x_i-(\bar{x}_{N-1})\cdot(N-1)=\sum_{i=1}^{N-1}x_i-\sum_{i=1}^{N-1}\frac{x_i}{N-1}\cdot(N-1)=0\,. \tag{41}$$

We also note that the next to last term can be rewritten, using again equation 36, as

$$\frac{N-1}{N^2}(x_N-\bar{x}_{N-1})(y_N-\bar{y}_{N-1})=\frac{1}{N}(x_N-\bar{x}_{N-1})(y_N-\bar{y}_N)\,. \tag{42}$$

Therefore,

$$\mathcal{M}_N(x,y)=\mathcal{M}_{N-1}(x,y)+(x_N-\bar{x}_{N-1})(y_N-\bar{y}_N)\,. \tag{43}$$

Expressing equation 43 in terms of the covariance results in

$$\mathrm{C}_N(x,y)=\mathrm{C}_{N-1}(x,y)+\frac{1}{N-1}\left[\frac{N}{N-1}(x_N-\bar{x}_N)(y_N-\bar{y}_N)-\mathrm{C}_{N-1}(x,y)\right]\,. \tag{44}$$

If the mean was known a priori (i.e. if it is not estimated from the sample) the definition of the unbiased covariance estimator over a sample of size N has N, instead of $N-1$, at denominator. In this case equation 44 becomes

$$\mathrm{C}_N(x,y)=\mathrm{C}_{N-1}(x,y)+\frac{(x_N-\bar{x}_N)(y_N-\bar{y}_N)}{N-1}-\frac{\mathrm{C}_{N-1}(x,y)}{N}\,. \tag{45}$$

Finally, we express equations 44 and 45 in a format more convenient for a C++ implementation as

$$\mathrm{C}_{N-1}(x,y)\mathrel{+}=\frac{1}{N-1}\left[\frac{N}{N-1}(x_N-\bar{x}_N)(y_N-\bar{y}_N)-\mathrm{C}_{N-1}(x,y)\right]\,, \tag{46}$$

and

$$\mathrm{C}_{N-1}(x,y) \mathrel{+}= \frac{(x_N - \bar{x}_N)(y_N - \bar{y}_N)}{N-1} - \frac{\mathrm{C}_{N-1}(x,y)}{N}\,, \tag{47}$$

respectively.

Bibliography

[1] Ashmanskas B et al. The cdf silicon vertex trigger. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 518(1–2):532 – 536, 2004. Frontier Detectors for Frontier Physics: Proceedings.

[2] Gauss and Markov. Gauss-Markov Theorem. `http://en.wikipedia.org/wiki/GaussMarkov_theorem`. [Online; accessed Nov-10-2014].

[3] Philippe Pebay. Formulas for robust, one-pass parallel computation of covariances and arbitrary-order statistical moments. Technical report, Sandia National Laboratories, 2008. SAND2008-6212.

[4] Luciano Ristori and Giovanni Punzi. Triggering on heavy flavors at hadron colliders. *Annual Review of Nuclear and Particle Science*, 60(1):595–614, 2010.