DEUTSCHES ELEKTRONEN-SYNCHROTRON
Ein Forschungszentrum der Helmholtz-Gemeinschaft

DESY 21-218
arXiv:2112.09470
December 2021

A Non-Linear Kalman Filter for Track Parameters Estimation in High Energy Physics

X. Ai, N. Styles

Deutsches Elektronen-Synchrotron DESY, Hamburg

H. M. Gray

Department of Physics, University of California, Berkeley, USA

and

Physics Division, Lawrence Berkeley National Laboratory, Berkeley, USA

A. Salzburger

CERN, Geneva, Switzerland

ISSN 0418-9833

NOTKESTRASSE 85 - 22607 HAMBURG

Herausgeber und Vertrieb:

Verlag Deutsches Elektronen-Synchrotron DESY

DESY Bibliothek
Notkestr. 85
22607 Hamburg
Germany

A Non-Linear Kalman Filter for track parameters estimation in High Energy Physics

Xiaocong Ai[a,*], Heather M. Gray[b,c], Andreas Salzburger[d], Nicholas Styles[a]

[a] *Deutsches Elektronen-Synchrotron DESY, Notkestr. 85, 22607 Hamburg, Germany*
[b] *Department of Physics, University of California, 425 Physics South MC 7300 Berkeley, CA, 94720, USA*
[c] *Physics Division, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, USA*
[d] *CERN, Espl. des Particules 1, 1217 Meyrin, Switzerland*

Abstract

The Kalman Filter is a widely used approach for the linear estimation of dynamical systems and is frequently employed within nuclear and particle physics experiments for the reconstruction of charged particle trajectories, known as tracks. Implementations of this formalism often make assumptions on the linearity of the underlying dynamic system and the Gaussian nature of the process noise, which is violated in many track reconstruction applications. This paper introduces an implementation of a Non-Linear Kalman Filter (NLKF) within the ACTS track reconstruction toolkit. The NLKF addresses the issue of non-linearity by using a set of representative sample points during its track state propagation. In a typical use case, the NLKF outperforms an Extended Kalman Filter in the accuracy and precision of the track parameter estimates obtained, with the increase in CPU time below a factor of two. It is therefore a promising approach for use in applications where precise estimation of track parameters is a key concern.

Keywords: Non-linear system, Non-linear Kalman filter, Nuclear and particle physics experiment, Track parameter estimates
2021 MSC: 00-01, 99-00

1. Introduction

The reconstruction of the trajectories of charged particles requires the identification of the set of hits corresponding to a single particle and the determination of the kinematic properties of the particle's trajectory by fitting that set of hits. The most commonly used algorithm for the reconstruction of charged particle trajectories, or tracks, in nuclear and particle physics is the Kalman Filter (KF). The KF was introduced approximately 70 years ago [1] and is used in many fields including navigation, aerospace engineering, space engineering, remote surveillance, telecommunications, physics, audio signal processing and control engineering.

The KF processes a set of discrete measurements to determine the internal state of a linear dynamical system. Both the measurements and the system can be subjected to independent random perturbations or noise. By combining predictions based on the previous state estimates with subsequent measurements, the impact of these perturbations on the following state estimates can be minimized. The Kalman filter is known to be the optimal linear estimator for such linear systems.

The KF for track reconstruction was introduced to particle physics by the DELPHI experiment [2] at the Large Electron Positron (LEP) collider at the European Council for Nuclear Research (CERN). In track reconstruction [3], the description of the system incorporates the impact of magnetic fields and detector material on charged particle trajectories[1]. KF algorithms are used both in track finding, where the collection of measurements correspond-

*Corresponding author
Email addresses: xiaocong.ai@desy.de (Xiaocong Ai), heather.gray@berkeley.edu (Heather M. Gray), andreas.salzburger@cern.ch (Andreas Salzburger), nicholas.styles@desy.de (Nicholas Styles)

[1] Magnetic fields are used to deflect the trajectory to allow

ing to a single charged particle trajectory are identified, and in track fitting, where the parameters describing the trajectory of the charged particle are determined from a set of measurements. To date, the KF remains the method with the best overall performance for most track reconstruction applications. See Ref [4] for a review.

KF algorithms for track reconstruction typically proceed in two steps. The starting point is the track seed, which is an initial coarse trajectory estimate for a candidate track, based on a small number of measurements, typically three or four. Subsequent measurements are added progressively to the track seed following a track propagation to reachable detection elements. Once all the measurements have been added, a second smoothing step [5] is performed, which runs a second filtering sequence in the opposite direction. This means that information from all measurements are included in the track parameter estimates at all measurement points. Without the smoothing step, only the parameters estimated at the final measurement point would include the information from all measurement points due to the progressive nature of the KF procedure. An extension of the KF is the Combinatorial Kalman Filter (CKF) [6, 7, 8], which can be used to perform track finding and track fitting simultaneously and allows branching of track candidates.

Despite the success of the KF, a key limitation for many applications is the assumption of linear models for the system and measurement and Gaussian distributions for the system state, process and measurement noise. This has motivated the development of a number of extensions. One such extension is the Extended Kalman filter (EKF) [9] which linearizes a model using a first-order Taylor expansion. This improved description is insufficient in particular when the incidence angle of the charged particle on the measurement surface is large. The EKF assumes that the contribution from the noise is described by a Gaussian distribution, which is not necessarily appropriate.

The Gaussian Sum Filter (GSF) [10] relaxes the assumption of Gaussian process noise by assuming that the noise distribution can be described by a sum of Gaussian distributions [11]. In the domain of nuclear and particle physics, this is particularly important when modelling radiative energy loss such as is common when electrons lose energy through bremsstrahlung when passing through tracking detectors [12, 13]. The application of the GSF procedure is typically restricted to track candidates which have been identified as being a potential electron candidate (e.g. by combining track information with calorimeter information, or other forms of particle identification such as transition radiation). The GSF does not address non-linear effects in tracking fitting.

This paper will explore a non-linear Kalman filter (NLKF) based on the Unscented Kalman filter (UKF) [14, 15], which uses a set of discretely sampled points to parameterize the mean and covariance to account for non-linearities of the system and measurement models. It has been shown to have comparable performance to a second-order Gaussian filter. We investigate the application of the UKF to charged particle reconstruction for high-energy nuclear and particle physics experiments.

The manuscript is organized as follows. Section 2 provides a brief introduction to track reconstruction and A Common Tracking Software Toolkit (ACTS) [16]. The formalism for the EKF is discussed in Section 3 and the extension to the NLKF in Section 4. Section 5 compares the performance of the EKF and the NLKF using a typical detector geometry. Brief conclusions are presented in Section 6.

2. Track reconstruction and the ACTS toolkit

A Common Tracking Software (ACTS) is a toolkit providing a set of encapsulated track reconstruction components that can be used by a wide range of experiments. ACTS features an internal geometry and navigation model, including a minimal Event Data Model (EDM) implementation that allows client applications to augment and extend the data with information specific to the target experiment. It imposes minimal dependencies. ACTS is written in C++17 using modern programming best-practises and follows a component level design that provides encapsulated, stateless modules. These modules perform well-defined tasks for track reconstruction (e.g. track propagation or track fitting) and are designed to be executed in parallel call paths if desired, in compliance with modern multi-core CPU architectures. ACTS is currently used within a number of nuclear and particle physics

the charged particle momentum to be measured and material effects cause random fluctuations due to elastic scattering and energy loss

experiments, e.g. ATLAS [17], sPHENIX [18] and FASER [19], and is being investigated as a potential track reconstruction library by a number of others [20, 21, 22, 23, 24].

Based on its internal geometry and navigation model, ACTS provides a fast[2] track simulation engine, based on the concept of the ATLAS Fast Track Simulation (Fatras) [26]. The internal navigation model of the ACTS geometry is used to predict the particle trajectories through the detector. Hits are created at the intersection points of the trajectory with sensitive detector elements, and the interaction of particles with detector material is modelled using approximate electromagnetic and hadronic physics models. The recorded hits are processed by a digitization module that emulates the detector readout and provides an estimate for the detector resolution.

In ACTS, candidate tracks are created from input measurements by a series of track reconstruction algorithms, and are represented by a series of track states, representing the trajectory at various points. A track state can be expressed in either a free (also called global) or a local representation. Local representations are constrained to a surface description within the detector.

The free (global) track parameters, g, are 8-dimensional and represented as:

$$g = (x, y, z, t, d_x, d_y, d_z, q/p). \tag{1}$$

The first four parameters are the space-time (x, y and z for position and t for time) coordinates of the track state, d_x, d_y and d_z represent the direction of the track at that point, and q/p is the ratio of the charge, q, and momentum, p. The local track parameters, l, are 6-dimensional and represented as[3]:

$$l = (loc_0, loc_1, \phi, \theta, q/p, t). \tag{2}$$

Here, loc_0 and loc_1 are the coordinates of the track in the local coordinate frame of a reference surface, the ϕ and θ are angles representing the track direction in the polar frame, and the q/p and t are the same as in the global track parameters. The reference surface can consist of different types or shapes, including cylindrical or planar surfaces, or surfaces describing straw-like detector or virtual lines. An example of a line surface is the perigee surface used to describe the track parameters near the vertex[4]. The track parameters on a perigee surface are called the perigee track parameters and, in this case, the loc_0 and loc_1 are often denoted as d_0 and z_0, which are the transverse and longitudinal impact parameters. The perigee parameters are often used when the track is described by a single set of parameters at its estimated point of production, which is typically of most relevance for physics analyses. See Ref. [16] for more details of the track parametrization.

In the ACTS Kalman filtering algorithm, the track state is represented by the local track parameters expressed at measurement planes. Measurements are represented by a subset of the local track parameters, as explained in Section 4.3.

3. Track fitting with Extended Kalman filter

Track fitting with a Kalman filter requires evolving the track state and its associated covariance matrix, as it is propagated through a discrete dynamical system. This can be described by a track state propagation model:

$$x_k = f_{k-1}(x_{k-1}) + \eta_{k-1}. \tag{3}$$

Here,

- x_{k-1} and x_k are the track state vector at the states $k-1$ and k, respectively.
- η_{k-1} is the vector representing the noise when propagating from state $k-1$ to state k, i.e. process noise.
- f_{k-1} is the track state propagation model from $k-1$ to state k, which describes the motion of the particle. It depends on the kinematics of the particle and the magnetic field.

The track state is projected onto the measurement using the measurement projection model:

$$y_k = h_k(x_k) + \epsilon_k. \tag{4}$$

Here,

[2] i.e. Fatras is significantly simplified with respect to a physics-based simulation such as Geant4 [25], resulting in orders-of-magnitude faster processing times

[3] We assume a right-handed coordinate system, with the polar angle θ measured from the positive z-axis in an interval of $[0, \pi]$, and the azimuthal angle $\phi \in [-\pi, \pi)$ defined in the transverse x-y plane, with $\phi = 0$ denoting the x-axis

[4] The vertex is assumed to be the common point where particles from a single interaction or decay originated. ACTS also includes algorithms for reconstructing the positions of such vertices from a set of input tracks

- y_k is the measurement vector at state k.
- ϵ_k is the measurement noise vector at state k.
- h_k is the measurement projection function from track state to measurement, which depends on the kinematics of the particles and detector geometry.

Both the track state propagation model, f, and the measurement projection model, h, are often non-linear functions. The process noise η_{k-1} and measurement noise ϵ_k are assumed to be Gaussian distributions with zero means, and variances Q_k and V_k respectively, however they may not necessarily follow Gaussian distributions.

For the EKF, the f and h are approximated with linear models as follows:

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + \eta_{k-1}, \\ y_k &= H_k x_k + \epsilon_k, \end{aligned} \tag{5}$$

where F_{k-1} is the first-order Taylor expansion of the track propagation function f at state $k-1$, and H_k is the first-order Taylor expansion of the measurement projection function h at state k. As before, η and ϵ are the process and measurement noise vectors.

In nuclear and particle experiments which often have inhomogeneous magnetic fields, F_{k-1} is calculated using the Runge-Kutta method [27] to numerically solve the second-order differential equations describing charged particles moving through magnetic fields. For example, the ATLAS experiment uses an adaptive Runge-Kutta-Nyström approach [28], which adapts the step size to minimize computational costs while ensuring that the estimation error remains below a set threshold. The H_k matrix is obtained by analytically calculating the derivative of h with respect to the track state vector and accounting for the angle at which the track intersects the detector module.

The EKF includes three steps: the *prediction* of the track state at state k based on previous $k-1$ measurements, the *filtering* of predicted track state at state k taking into account the measurement at state k, and the *smoothing* of the filtered track state with all measurements taken into account. A full description can be found in Ref [3]. Here we briefly outline the formulae used to update the track state vector, x and its covariance, C.

- Prediction:

$$\begin{aligned} x_k^{k-1} &= F_{k-1}x_{k-1}, \\ C_k^{k-1} &= F_{k-1}C_{k-1}F_{k-1}^T, \end{aligned} \tag{6}$$

  where the upper index $k-1$ indicates the estimate prior to the filtering, i.e. with only the previous $k-1$ measurements taken into account.

- Filtering:

$$\begin{aligned} x_k &= x_k^{k-1} + K_k(m_k - H_k x_k^{k-1}), \\ C_k &= (1 - K_k H_k)C_k^{k-1}, \end{aligned} \tag{7}$$

  where m_k is the measurement on state k, and the K_k is the Kalman gain matrix:

$$K_k = C_k^{k-1}H_k^T(V_k + H_k C_k^{k-1}H_k^T)^{-1}. \tag{8}$$

- Smoothing:

$$\begin{aligned} x_k^n &= x_k + A_k(x_{k+1}^n - x_{k+1}^k), \\ C_k^n &= C_k + A_k(C_{k+1}^n - C_{k+1}^k)A_k^T, \end{aligned} \tag{9}$$

  where the upper index n indicates the smoothed estimation with all n measurements taken into account, and the A_k is the smoother gain matrix:

$$A_k = C_k F_k^T (C_{k+1}^k)^{-1}. \tag{10}$$

4. The Non-linear Kalman filter

4.1. Non-linear effects in track reconstruction

Tracking detectors at particle colliders follow a cylindrical or layered approach. A cylindrical detector typically consists of concentric cylindrical layers, which are oriented parallel to the beam direction, in the barrel, and disk layers, which are oriented normal to the beam direction, in the forward regions. This guarantees a close to hermetic coverage of the phase space of the particles produced in the collisions, while complying with mechanical constraints and minimizing detector material. When a track from the beam interaction point intersects with a detector module, the dependence of the intersection position on the incident track direction is non-linear. Fig. 1 demonstrates an example of such non-linearity for simplified detector

consisting of two parallel detector planes. The local coordinates of the state k are shown as a function of the polar and azimuthal angles of the track direction at the previous state $k-1$. In this example the functions are closest to linear when the azimuthal angle and polar angle are zero, which corresponds to the case when the track intersects the detector module at a perpendicular angle, or zero incidence angle, and become increasingly non-linear when the absolute angles get larger. This effect is particularly significant for the polar angle, which is highly correlated with the track incidence angle. These non-linear effects can be addressed by the NLKF.

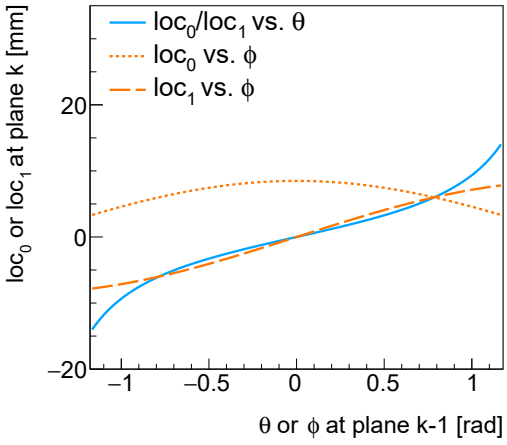

Figure 1: Example of non-linear dependence of the local coordinate of the intersection of a track on detector plane k on the track direction on detector plane $k-1$ for two parallel detector planes oriented normal to the beam direction. The loc_0 and loc_1 have the same dependence on the θ, which is shown by the solid blue line. The dependence of the loc_0 and loc_1 on the ϕ are shown by the short dashed and long dashed orange lines, respectively.

4.2. NLKF formalism

The NLKF calculates the propagated or projected track state and covariance using a set of sample points around the mean of the track state being propagated or projected, with each point assigned a weight. This is analogous to the random sampling of a distribution function in Monte Carlo simulation, which is the method typically used to generate events corresponding to different physics processes. For a N-dimensional track state vector x_{k-1} with covariance C_{k-1} at state $k-1$, $2N+1$ sample points are generated [14, 15]. This comprises the nominal track state vector plus $2N$ vectors generated by varying the nominal track state vector along the direction of the eigenvectors of the covariance matrix. The magnitudes of the variations are given by the eigenvalues of the covariance matrix. The eigenvectors and eigenvalues of the covariance matrix are obtained via Singular Value Decomposition (SVD) [29]. C_{k-1} is a real symmetric matrix and therefore can be expressed through SVD as,

$$C_{k-1} = U_{k-1} S_{k-1} U_{k-1}^T, \tag{11}$$

where U_{k-1} is a unitary matrix whose columns are the eigenvectors of C_{k-1}, and S_{k-1} is a diagonal matrix whose non-zero diagonal elements are the corresponding eigenvalues of C_{k-1}. Denoting the i-th column of U_{k-1} as u^i_{k-1} and the i-th diagonal element of S_{k-1} as s^i_{k-1}, N sets of orthogonal shifting vectors δ^i_{k-1} are,

$$\delta^i_{k-1} = \sqrt{s^i_{k-1}}\, u^i_{k-1}, \quad i = 1, ..., N, \tag{12}$$

where $\sqrt{s^i_{k-1}}$ is the magnitude of the variation in the direction of u^i_{k-1}.

The $2N+1$ sample points for x_{k-1} are:

$$\mathrm{X}^{(i)}_{k-1} = \begin{cases} x_{k-1}, & i = 0; \\ x_{k-1} + \gamma\delta^i_{k-1}, & i = 1, ..., N; \\ x_{k-1} - \gamma\delta^{i-N}_{k-1}, & i = N+1, ..., 2N, \end{cases} \tag{13}$$

where γ is a scaling parameter,

$$\gamma = \sqrt{N + \lambda}, \quad \lambda = \alpha^2 N - N, \tag{14}$$

and α is a tuning parameter used to control the deviation of the sample point from the nominal point, in the range $0 < \alpha \leq 1$.

In principle, the sample points could be propagated using the track model using Eq. 3,

$$\mathrm{X}^{(i)}_k = f_{k-1}(\mathrm{X}^{(i)}_{k-1}) + \eta_{k-1}, \quad i = 0, ..., 2N, \tag{15}$$

and projected to a measurement point using the measurement model with Eq. 4,

$$\mathrm{Y}^{(i)}_k = h_k(\mathrm{X}^{(i)}_k) + \epsilon_k, \quad i = 0, ..., 2N. \tag{16}$$

However, because the Runge-Kutta method already accounts for the non-linearity of the track model, we apply only the second of these non-linearity corrections: the projection of the track state to the measurement point.

The mean, y_k, and covariance, P_k, of the projected track state are calculated as,

$$\begin{aligned} y_k &= \sum_{i=0}^{2N} w_m^{(i)} \mathrm{Y}_k^{(i)}, \\ P_k &= \sum_{i=0}^{2N} w_c^{(i)} (\mathrm{Y}_k^{(i)} - y_k)(\mathrm{Y}_k^{(i)} - y_k)^T + V_k, \end{aligned} \tag{17}$$

and the covariance between the track state and the measurement, T_k is calculated as

$$T_k = \sum_{i=0}^{2N} w_c^{(i)} (\mathrm{X}_k^{(i)} - x_k^{k-1})(\mathrm{Y}_k^{(i)} - y_k)^T. \tag{18}$$

In Eq. 17 and Eq. 18, the weights $w_m^{(i)}$ and $w_c^{(i)}$ are defined as,

$$\begin{aligned} w_m^{(0)} &= \tfrac{\lambda}{N+\lambda}, \quad i = 0, \\ w_c^{(0)} &= \tfrac{\lambda}{N+\lambda} + (1 - \alpha^2 + \beta), \quad i = 0, \\ w_m^{(i)} &= w_c^{(i)} = \tfrac{1}{2(N+\lambda)}, \quad i = 1, ..., 2N, \end{aligned} \tag{19}$$

where β is a non-negative weighting parameter used to tune the weight of the $\mathrm{Y}^{(0)}$ when calculating P_k. A value of $\beta = 2$ as suggested in Ref. [30] is used.

The Kalman gain is calculated as,

$$K_k = T_k P_k^{-1}, \tag{20}$$

and used, with the mean and covariance, to update the track state and its covariance in the Kalman filtering step,

$$\begin{aligned} x_k &= x_k^{k-1} + K_k(m_k - y_k), \\ C_k &= C_k^{k-1} - K_k P_k K_k^T. \end{aligned} \tag{21}$$

4.3. Implementation of NLKF in ACTS

As described in Section 2, a measurement is described by a subset of the local track parameters in ACTS. Therefore, projecting a track state to a measurement is equivalent to transforming the global track parameters to the local track parameters and projecting the local track parameters to the measurement by an identity projection matrix. The track state is represented by global track parameters during its propagation between detector planes and transformed to local track parameters at the detector plane where a material effect needs to be taken into account or a measurement is present. In the latter case, the measurement is used to update the predicted track state x_k^{k-1} and its covariance C_k^{k-1} represented by the local track parameters at state k using Eq. 7.

If the incidence angle of track on a detector plane is larger than a certain value, the transformation of a single set of global track parameters to local track parameters is replaced by the transformation of the 17 sets[5] of global track parameters to the local track parameters. Eq. 16 is used and the corrected local track parameters and associated covariance are calculated using Eq. 17. In ACTS, the covariance in Eq. 18 between the track state and the measurement is part of the covariance matrix of the local track parameters, and therefore no additional calculation of this term is needed and the Kalman gain formulas for the EKF and NLKF are identical. Therefore, the same Kalman filtering and smoothing formulae for the EKF are used for NLKF with the predicted local track parameters x_k^{k-1} and its covariance C_k^{k-1} in Eq. 7 replaced by the corresponding corrected local track parameters.

4.4. Comparison of the EKF and NLKF for track fitting

Figure. 2 illustrates the impact of the non-linearity effects on track parameter propagation using the configuration shown in Figure 1. Given the configuration of the track parameters at plane $k-1$, the local coordinate of the intersection of the track on plane k will have a true covariance indicated by the dashed green shape. The two arc sides, and two radial sides of the true covariance are due to variations in the polar and azimuthal angles of the track direction. If the EKF is used to transport the track from plane $k-1$ to plane k, the local coordinate of the track on plane k will have the error denoted by the blue ellipse. If the NLKF, is used, the local coordinate of the track on plane k will have the error denoted by the orange ellipse.

Such non-linear effects will impact the Kalman filtering procedure. In particular, the Kalman gain matrix in Eq. 8 tends to either over- or under- estimate the polar angle of the track. This effect is demonstrated in Fig. 3 using the configuration from

[5] As discussed in Sec. 4.2, the NLKF uses $2N+1$ samples points and N is 8 for the global track parameters

Fig. 2 and showing the pull distribution of the filtered polar angle. The pull value for track parameter v is defined as,

$$pull_v = \frac{v^{fit} - v^{truth}}{\sigma_v}. \quad (22)$$

Here v^{fit} and σ_v are the estimated value and uncertainty of the track parameter v respectively, and v^{truth} is the true simulated value of the v. If both the values and uncertainties of the track parameters are estimated correctly, the pull distributions are expected to follow normal distributions. To avoid any bias from assuming a Gaussian distribution, the mean and the *Root-Mean-Square* (RMS) values of the pulls are compared between the EKF and the NLKF. For the EKF, the filtered polar angles are biased to larger values than their true values with a large RMS. For the NLKF, the mean of the polar angles is biased to negative values, but the RMS is significantly improved. The impact of the non-linear effects on the azimuthal angle is smaller. Both implementations have mean at zero and RMS at 1.2.

5. Performance studies

The performance of EKF is evaluated using the Open Data Detector (ODD) [31]. The layout of the ODD is shown in Fig. 4. It consists of a pixel detector and two strip detectors with differing instrinsic resolution and it uses a realistic material model using the DD4hep [32] detector description tool. The ODD is immersed in a solenoidal magnetic field of 2 Tesla centered on the beam line.

A sample of 1 million simulated muons is used to study the performance, as muons are insensitive to the detector material. The muons are generated with transverse momentum[6] p_T uniformly distributed in the range of $0.4 < p_T < 100\,\text{GeV}$ and pseudorapidity[7] η uniformly distributed in the range of $|\eta| < 3.0$. The range in p_T allows us to study the impact of multiple scattering, which varies with p_t and the range in η allows us to study muons that intersect the detector modules at a range of angles. The intersection points of

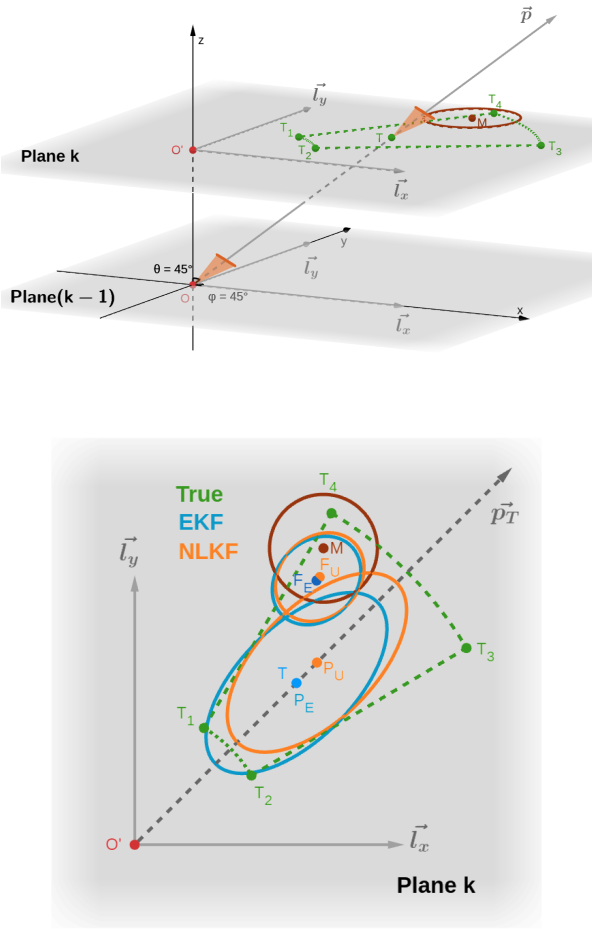

Figure 2: Illustration of the impact of non-linear effects during track parameter transformation for a two layer detector with no magnetic field. (Top) A track intersects planes $k-1$ and k at points O and T. The local coordinates of point O has zero covariance. The track direction has the azimuthal and polar angles of $\pi/4$ and a covariance denoted by the orange cones. (Bottom) The true covariance of the predicted local coordinates on plane k is shown by the green dashed curve, a measurement at M with its covariance is denoted by the red circle, and the predicted (filtered) local coordinates of the track using the EKF and the NLKF are shown with the ellipses centered P_E (F_E) and P_U (F_U), respectively.

the muons with the detectors, the simulated hits, are generated with the Fatras fast simulation engine within the ACTS toolkit. The input measurements to the Kalman filter algorithm are created by applying Gaussian smearing to the positions of the simulated hits to emulated the impact of detector resolution. One- and two-dimensional measurements in the local coordinate frames of the detector

[6]Transverse momentum is the momentum in the transverse x-y plane, $p_T = \sqrt{p_x^2 + p_y^2}$

[7]Pseudorapidity is an angular quantity calculated from the polar angle θ as follows $\eta = -\ln\left[\tan\left(\frac{\theta}{2}\right)\right]$. $\eta = +\infty$ corresponds to the direction of the beam.

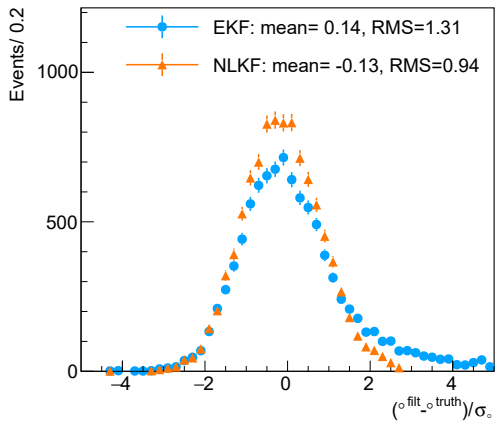

Figure 3: Comparison of pull of the filtered momentum direction polar angle θ using the EKF (blue) and NLKF (orange) with the setup in Fig. 2. Ten thousand tracks and their corresponding measurements are simulated.

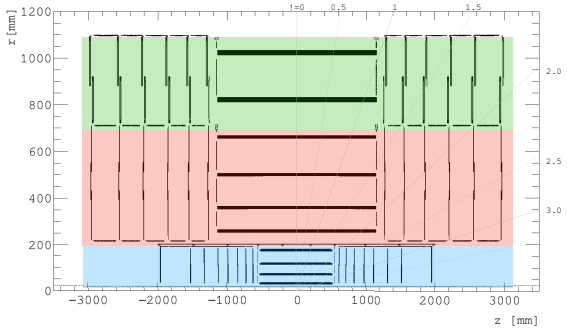

Figure 4: Schematic layout of the ODD silicon tracking detector projected into the z-r plane. The beam interaction would occur at $z = 0, r = 0$. The location of the pixel detector is shown in blue, the two strip detectors with different intrinsic resolution are shown in red and green, where the inner strip detector (red) has better resolution than the outer strip detector (green).

planes are created in the strip and pixel detectors of ODD, respectively, by smearing with Gaussian distributions with zero mean and different width (σ) as in Table 1.

The reconstructed seed of the track fit is emulated by smearing the vertex position, momentum and time of the generated muons using Gaussian distributions with zero mean and either momentum-dependent or constant width. The production vertex is smeared to obtain the local coordinates d_0 and z_0 using Gaussian distributions with $\sigma = a_0 + a_1 e^{-a_2 p_T}$, q/p is smeared using a Gaussian distribution with $\sigma = a_0/p$, and ϕ, θ and t are smeared using a Gaussian distribution with constant σ. Table 2 provides the parameters used to construct the width of the Gaussian used for the smearing, which are of similar order to the resolution of the tracking detectors at current nuclear and particle physics experiments.

Subdetectors	σ_x $[\mu m]$	σ_y $[\mu m]$
Pixel	15	15
Inner strip	43	-
Outer strip	72	-

Table 1: The width of Gaussian (with zero mean) used to smear the x and y coordinates of the truth hits in different sub-detectors of the ODD.

Track parameters	Smearing parameters
	$a_0 = 20$ μm
d_0, z_0	$a_1 = 30$ μm
	$a_2 = 0.3$ GeV^{-1}
ϕ, θ	$\sigma = 1°$
q/p	$a_0 = 0.01$ GeV^{-1}
t	σ =1 ns

Table 2: The parameters used used to construct the width of the Gaussian for smearing the generated vertex, momentum and t to obtain the seed of the track fit.

The physics and the computational performance of the EKF and NLKF are studied. The non-linear correction for the NLKF is only performed when the incidence angle of a track with a detector plane is larger than 0.1. The NLKF performance is found to be insensitive to the tuning parameter α so a fixed value of $\alpha = 0.1$ is used.

5.1. Track parameter estimation

The mean and RMS of the residuals, defined as $v^{fit} - v^{truth}$, and the pulls, defined in Eq. 22, of the perigee track parameters are used to evaluate the performance. The pull depends on the central value of the track parameter and its uncertainty, but the residual depends only on the central value. Ideally, the pulls would have means of zero and and RMSs of one and the residuals would have means of zero and the RMS of the detector resolution.

The mean and the RMS of the residuals and pulls are studied in bins as a function of η. The degree of non-linear effects, the number of detector layers and the amount of material that a charged particle

passes through vary with η. The results are presented with and without a magnetic field and with and without the impact of the particle interactions with the detector material. The results without the magnetic field are equivalent to a track fitting scenario without non-linear effects in the track state propagation model in which case non-linear effects are only due to the measurement model, which is directly addressed by this implementation. The impact of the non-linear effects on t is negligible and therefore only the RMS of its pull as a function of η is shown.

The mean of the residuals of the impact parameters, d_0 and z_0, as a function of η for simulated particles with $p_T > 20\,\text{GeV}$ are shown in the upper panel of Fig. 5. The mean estimated using the EKF is biased from zero at higher $|\eta|$ bins due to more pronounced non-linear effects in this region. However, such biases are absent when the NLKF is used. As there is a strong correlation between the residuals and pulls, the mean of the pulls show similar biases to the residual means of the perigee track parameters.

The resolution of the impact parameters as a function of η for simulated particles with $p_T > 20\,\text{GeV}$ are shown in the lower panel of Fig. 5. The NLKF improves their resolution by up to 50% at higher $|\eta|$ bins compared to the EKF. There is similar improvement for ϕ and θ. All track parameters are studied, and no improvement in the resolution of q/p is observed.

Fig. 6 shows the RMS of the pulls of all perigee track parameters as a function of η for simulated particles with $p_T > 20\,\text{GeV}$. The parameter t is unaffected by the non-linear effects and hence the RMS of its pulls is approximately one. Non-linear effects cause the RMS to deviate from one at higher $|\eta|$ for d_0, z_0, ϕ, θ and q/p when using the EKF. The deviation is largest for z_0 and θ where the RMS can reach up to 2.6 and smallest for q/p. The deviation is significantly reduced using the NLKF, i.e. the RMS for all track parameters is below 1.7 in the whole η range being studied and below 1.3 in the central region. No deviation for ϕ, θ and q/p is observed with the NLKF for tracks in such p_T range when magnetic field and material effects are present.

The dependence of the pulls on track p_T is studied in Fig. 7, which shows the RMS of the pulls of the impact parameters for simulated particles in the range of $1.0 < |\eta| < 2.5$. This η range was selected because non-linear effects are significant for these values. The RMS of the pulls is smaller for lower p_T tracks using the EKF. When there is no material and magnetic field, the deviation with the NLKF is always significantly smaller than that with the EKF for all values of p_T. When there is material and magnetic field, the NLKF achieves significantly better performance than the EKF for z_0. For d_0, the NLKF improves the RMS of the pulls for tracks with $p_T > 2\,\text{GeV}$ despite the fact that it corrects the bias of the mean of residual and pull for tracks in all values of p_T. The performance differences observed between the NLKF and the EKF for ϕ are similar to d_0 and for θ are similar to z_0. This is expected due to the correlations between the pairs of track parameters.

5.2. Computational Performance

Additional computational cost with the NLKF is expected due to the additional evaluation points, which are key to improving the precision. An estimate of this cost is obtained by comparing the track fitting time of the NLKF to that of the EKF as a function of η and p_T. In each η or p_T bin, track fitting is performed five times per sample with 1k tracks. The mean of the track fitting time per track from the five tests is shown as the nominal value, and the RMS is shown as the uncertainty bar. The tests are performed in a single thread using the Intel Core i7-8559U CPU @2.70 GHz processor.

Fig. 8 shows the track fitting time in HS06 [33] $\times$ ms per track as a function of η or p_T of the simulated particles with EKF and NLKF. The average fitting time per track with EKF is approximately 4.8 HS06 $\times$ ms and with NLKF it increases by a factor ranging from ~ 1.6 in the barrel region to ~ 1.8 at higher η. In general, track parameter estimation is not the most timing consuming step during track reconstruction, therefore this can be expected to have a negligible impact on the total time for track reconstruction in most applications.

6. Conclusion

The reconstruction of charged particle trajectories is a challenging computational task for nuclear and particle physics experiments today and in the future. The Kalman Filter algorithm is currently widely used due to its excellent performance, however, it is limited by its assumption of linear models for the system and measurements as well as Gaussian distributions for the noise. We have introduced

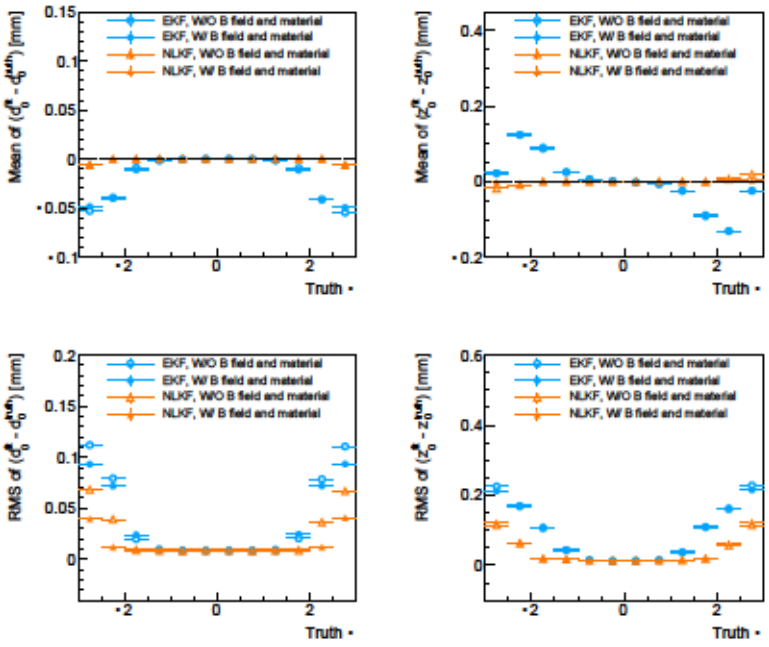

Figure 5: The mean (top) and RMS (bottom) of the residual of fitted perigee track parameters d_0 (left) and z_0 (right) parameterized as a function of simulated particle η ($20 < p_T < 100\,\text{GeV}$) for the ODD without (blue circles for EKF, orange hollow triangles for NLKF) and with (blue dots for EKF, orange filled triangles for EKF) the presence of a solenoidal magnetic field of 2 T and material effects. The dashed horizontal lines in the upper panel denote the expected mean of the residuals.

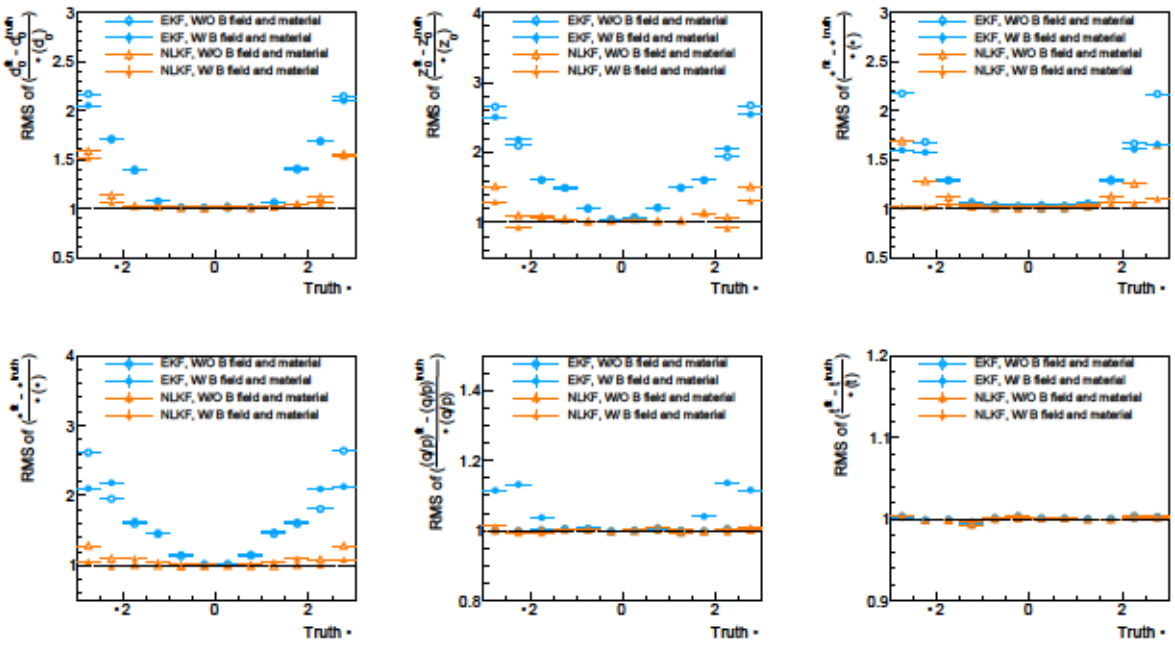

Figure 6: The RMS of the pull of fitted perigee track parameters d_0, z_0, ϕ, θ, q/p and t parameterized as a function of the simulated particle η ($20 < p_T < 100\,\text{GeV}$) for the ODD without (blue circles for EKF, orange hollow triangles for NLKF) and with (blue dots for EKF, orange filled triangles for EKF) the presence of a solenoidal magnetic field of 2 T and material effects. The dashed horizontal lines denote the expected RMS of the pulls.

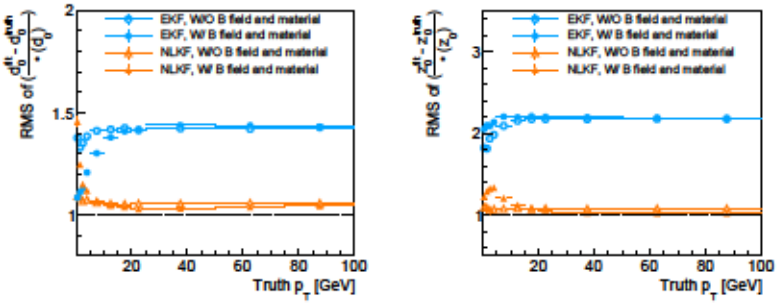

Figure 7: The RMS of the pull of fitted perigee track parameters d_0 and z_0 parameterized as a function of simulated particle p_T ($1.0 < |\eta| < 2.5$) for the ODD without (blue circles for EKF, orange hollow triangles for NLKF) and with (blue dots for EKF, orange filled triangles for EKF) the presence of a solenoidal magnetic field of 2 T and material effects for the track parameters d_0, z_0, ϕ, θ and q/p, respectively. The dashed horizontal lines denote the expected RMS of the pulls.

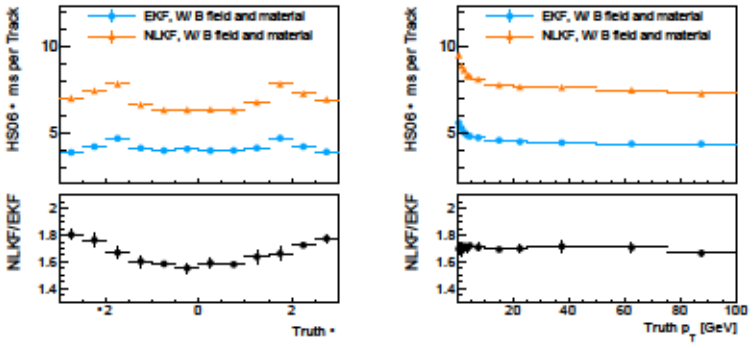

Figure 8: (Left) A comparison of the track fitting time per track as a function of simulated particle η ($20 < p_T < 100\,\text{GeV}$) between EKF and NLKF for the ODD at ATLAS B field. (Top) The track fitting time in HS06 × ms per track. The blue dots and orange triangles shown the results obtained using EKF and NLKF, respectively. (Bottom) The ratio of fitting time per track between NLKF and EKF. (Right) A comparison of the track fitting time per track as a function of simulated particle p_T ($1.0 < |\eta| < 2.5$) between EKF and NLKF for the ODD at a solenoidal magnetic field of 2 T. (Top) The track fitting time per track. The blue dots and orange triangles shown the results obtained using EKF and NLKF, respectively. (Bottom) The ratio of fitting time per track between NLKF and EKF.

the non-linear Kalman filter for charged particle reconstruction, which uses a set of discretely sampled points to account for non-linear effects.

We tested the performance of our NLKF algorithm using the ODD. The NLKF yields residuals for all track parameters with a mean of zero throughout η. In addition, the RMS of the residuals are reduced for most track parameters, by up to a factor of two. The effect is most pronounced in regions with larger incidence angle of the tracks on the measurement planes, which are located at large values of $|\eta|$ in the detector geometry we studied. Compared to the EKF, the NLKF also provides a more accurate estimation of the uncertainty of the parameters, which results in the RMS of the pulls being more consistent with one for a larger range of η. The improvement is more pronounced for tracks with larger p_T.

The computational requirements for the NLKF increase due to the additional evaluation points. We found that the time for track fitting increases from a factor of 1.6 to 1.8 depending on the p_T and η of the particle. However, track fitting is typically a small fraction of the total track reconstruction time in most applications.

In conclusion, the NLKF shows promising performance in improving the estimation of the track parameters corresponding to charged particle trajectories by accounting for non-linear effects. In particular, its use can be warranted in applications where the precision of the track parameters is particularly important.

Acknowledgments

Xiaocong Ai, Nicholas Styles acknowledge support from DESY (Hamburg, Germany), a member of the Helmholtz Association HGF.

Declarations

Funding. This work was funded by the NSF under Cooperative Agreement OAC-1836650.

Conflict of interest. The authors declare that they have no conflict of interest.

Availability of data and material. Not applicable. No associated data except for code.

Code availability. The code used for this research is available open source [34].

References

[1] R. E. Kalman, A New Approach to Linear Filtering and Prediction Problems, Journal of Basic Engineering 82 (1) (1960) 35–45. doi:10.1115/1.3662552.

[2] P. Abreu, et al., Performance of the DELPHI detector, Nucl. Instrum. Meth. A 378 (1996) 57–100. doi:10.1016/0168-9002(96)00463-9.

[3] R. Fruhwirth, Application of Kalman filtering to track and vertex fitting, Nucl. Instrum. Meth. A262 (1987) 444–450. doi:10.1016/0168-9002(87)90887-4.
URL https://doi.org/10.1016/0168-9002(87)90887-4

[4] A. Strandlie, R. Frühwirth, Track and vertex reconstruction: From classical to adaptive methods, Rev. Mod. Phys. 82 (2010) 1419–1458. doi:10.1103/RevModPhys.82.1419.
URL https://link.aps.org/doi/10.1103/RevModPhys.82.1419

[5] H. Rauch, F. Tung, C. Striebel, Maximum likelihood estimates of linear dynamical systems, AIAA 3 (1965) 1445. doi:https://doi.org/10.2514/3.3166.

[6] P. Billoir, Progressive track recognition with a Kalman-like fitting procedure, Comput. Phys. Commun. 57 (1) (1989) 390–394. doi:10.1016/0010-4655(89)90249-X.

[7] P. Billoir, S. Qian, Simultaneous pattern recognition and track fitting by the Kalman filtering method, Nucl. Instrum. Methods. Phys. Res. A 294 (1) (1990) 219–228. doi:10.1016/0168-9002(90)91835-Y.

[8] R. Mankel, A concurrent track evolution algorithm for pattern recognition in the HERA-B main tracking system, Nucl. Instrum. Methods. Phys. Res. A 395 (2) (1997) 169–184. doi:10.1016/S0168-9002(97)00705-5.

[9] F. E. Daum, Extended Kalman Filters, Springer London, London, 2015, pp. 411–413. doi:10.1007/978-1-4471-5058-9_62.
URL https://doi.org/10.1007/978-1-4471-5058-9_62

[10] R. Frühwirth, S. Frühwirth-Schnatter, On the treatment of energy loss in track fitting, Computer Physics Communications 110 (1) (1998) 80–86. doi:https://doi.org/10.1016/S0010-4655(97)00157-4.
URL https://www.sciencedirect.com/science/article/pii/S0010465597001574

[11] R. Frühwirth, A Gaussian-mixture approximation of the Bethe–Heitler model of electron energy loss by bremsstrahlung, Computer Physics Communications 154 (2) (2003) 131–142. doi:https://doi.org/10.1016/S0010-4655(03)00292-3.
URL https://www.sciencedirect.com/science/article/pii/S0010465503002923

[12] ATLAS Collaboration, Electron reconstruction and identification in the ATLAS experiment using the 2015 and 2016 LHC proton–proton collision data at $\sqrt{s} = 13\,\text{TeV}$, Eur. Phys. J. C 79 (2019) 639. arXiv:1902.04655, doi:10.1140/epjc/s10052-019-7140-6.

[13] CMS Collaboration, Performance of electron reconstruction and selection with the CMS detector in proton–proton collisions at $\sqrt{s} = 8\,\text{TeV}$, JINST 10 (2015) P06005. arXiv:1502.02701, doi:10.1088/1748-0221/10/06/P06005.

[14] S. J. Julier, J. K. Uhlmann, New extension of the Kalman filter to nonlinear systems, in: I. Kadar (Ed.), Signal Processing, Sensor Fusion, and Target Recognition VI, Vol. 3068, International Society for Optics and Photonics, SPIE, 1997, pp. 182 – 193. doi:10.1117/12.280797.
URL https://doi.org/10.1117/12.280797

[15] S. Julier, J. Uhlmann, Unscented filtering and nonlinear estimation, Vol. 92, 2004, pp. 401–422. doi:10.1109/JPROC.2003.823141.

[16] X. Ai, C. Allaire, N. Calace, A. Czirkos, I. Ene, M. Elsing, R. Farkas, L.-G. Gagnon, R. Garg, P. Gessinger, H. Grasland, H. M. Gray, C. Gumpert, J. Hrdinka, B. Huth, M. Kiehn, F. Klimpel, A. Krasznahorkay, R. Langenberg, C. Leggett, J. Niermann, J. D. Osborn, A. Salzburger, B. Schlag, L. Tompkins, T. Yamazaki, B. Yeo, J. Zhang, G. Mania, B. Kolbinger, E. Moyse, D. Rousseau, A Common Tracking Software Project (2021). arXiv:2106.13593.

[17] G. Aad, et al., The ATLAS Experiment at the CERN Large Hadron Collider, JINST 3 (2008) S08003. doi:10.1088/1748-0221/3/08/S08003.

[18] J. D. Osborn, A. D. Frawley, J. Huang, S. Lee, H. P. D. Costa, M. Peters, C. Pinkenburg, C. Roland, H. Yu, Implementation of ACTS into sPHENIX Track Reconstruction, Computing and Software for Big Science 5 (1) (2021) 23. doi:10.1007/s41781-021-00068-w.
URL https://doi.org/10.1007/s41781-021-00068-w

[19] A. Ariga, et al., FASER: ForwArd Search ExpeRiment at the LHC (1 2019). arXiv:1901.04468.

[20] T. Abe, et al., Belle II Technical Design Report (2010). arXiv:1011.0352.

[21] The CEPC Study Group, CEPC Conceptual Design Report: Volume 2 - Physics & Detector (2018). arXiv:1811.10545.

[22] J. Smyrski, Overview of the PANDA Experiment, Physics Procedia 37 (2012) 85–95, Proceedings of the 2nd International Conference on Technology and Instrumentation in Particle Physics (TIPP 2011). doi:https://doi.org/10.1016/j.phpro.2012.02.359.
URL https://www.sciencedirect.com/science/article/pii/S1875389212016690

[23] A. Accardi, et al., Electron Ion Collider: The Next QCD Frontier - Understanding the glue that binds us all (2014). arXiv:1212.1701.

[24] T. Åkesson, A. Berlin, N. Blinov, O. Colegrove, G. Collura, V. Dutta, B. Echenard, J. Hiltbrand, D. G. Hitlin, J. Incandela, J. Jaros, R. Johnson, G. Krnjaic, J. Mans, T. Maruyama, J. McCormick, O. Moreno, T. Nelson, G. Niendorf, R. Petersen, R. Pöttgen, P. Schuster, N. Toro, N. Tran, A. Whitbeck, Light Dark Matter eXperiment (LDMX) (2018). arXiv:1808.05219.

[25] S. Agostinelli, et al., GEANT4: A Simulation toolkit, Nucl. Instrum. Meth. A 506 (2003) 250–303. doi:10.1016/S0168-9002(03)01368-8.

[26] K. Edmonds, S. Fleischmann, T. Lenz, C. Magass, J. Mechnich, A. Salzburger, The Fast ATLAS Track Simulation (FATRAS), Tech. rep., CERN, Geneva (Mar 2008).
URL https://cds.cern.ch/record/1091969

[27] J. Myrheim, L. Bugge, A fast Runge-Kutta method for fitting tracks in a magnetic field, Nucl. Instrum. Meth. 160 (1) (1979) 43–48. doi:10.1016/0029-554X(79)90163-0.

[28] E. Lund, L. Bugge, I. Gavrilenko, A. Strandlie, Transport of covariance matrices in the inhomogeneous magnetic field of the ATLAS experiment by the application of a semi-analytical method, Journal of Instrumentation 4 (04) (2009) P04016–P04016.

doi:10.1088/1748-0221/4/04/p04016.
URL https://doi.org/10.1088/1748-0221/4/04/p04016

[29] M. Roth, F. Gustafsson, An efficient implementation of the second order extended Kalman filter, in: 14th International Conference on Information Fusion, 2011, pp. 1–6.

[30] R. V. D. Merwe, E. Wan, Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models, in: In Proceedings of the Workshop on Advances in Machine Learning, 2003.

[31] C. Allaire, P. Gessinger, J. Hdrinka, M. Kiehn, F. Kimpel, J. Niermann, A. Salzburger, S. Sevova, OpenDataDetector (Apr. 2021). doi:10.5281/zenodo.4674401.

[32] M. Petrič, M. Frank, F. Gaede, S. Lu, N. Nikiforou, A. Sailer, Detector Simulations with DD4hep, Journal of Physics: Conference Series 898 (2017) 042015. doi:10.1088/1742-6596/898/4/042015.

[33] Hep-spec06 benchmark, https://w3.hepix.org/benchmarking.html.

[34] Acts on github, https://github.com/acts-project/acts.