Nuclear Inst. and Methods in Physics Research, A 1049 (2023) 168041

Contents lists available at ScienceDirect

Nuclear Inst. and Methods in Physics Research, A

journal homepage: www.elsevier.com/locate/nima

Full Length Article

A non-linear Kalman filter for track parameters estimation in high energy physics

Xiaocong Ai [e,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
[e] School of Physics and Microelectronics, Zhengzhou University, Zhengzhou, Henan, 450001, China

ARTICLE INFO

MSC:
00-01
99-00

Keywords:
Non-linear system
Non-linear Kalman filter
Nuclear and particle physics experiment
Track parameters estimation

ABSTRACT

The Kalman Filter is a widely used approach for the linear quadratic 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 and measurement model, which are violated in a number of 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 the projection of the track state to the measurement. In a typical use case, the NLKF outperforms the so-called Extended Kalman Filter in the accuracy and precision of the track parameter estimates obtained, with an 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.

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 using a track model. 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 for track reconstruction was introduced to particle physics by the DELPHI experiment [2] at the Large Electron Positron (LEP) collider at the European Organization for Nuclear Research (CERN).

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 KF is known to be the optimal linear estimator for such linear systems.

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 corresponding 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 smoothing operation is performed by either using

* Corresponding author at: School of Physics and Microelectronics, Zhengzhou University, Zhengzhou, Henan, 450001, China.
E-mail addresses: xiaocongai@zzu.edu.cn (X. Ai), heather.gray@berkeley.edu (H.M. Gray), andreas.salzburger@cern.ch (A. Salzburger), nicholas.styles@desy.de (N. Styles).

[1] Magnetic fields are used to deflect the trajectory to allow the charged particle momentum to be measured and material effects cause random fluctuations due to elastic scattering and energy loss.

https://doi.org/10.1016/j.nima.2023.168041
Received 19 December 2021; Received in revised form 11 January 2023; Accepted 14 January 2023
Available online 16 January 2023
0168-9002/© 2023 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

the Rauch–Tung–Striebel (RTS) smoother formalism [5] or running a second filtering sequence in the opposite direction, i.e. backward filtering. This means that information from all measurements are included in the track parameter estimates at all measurement points. Without the smoothing operation, 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) [6–8], which can account for multiple matching hits during track finding.

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 linearized Extended Kalman filter (EKF) [9] which uses a model that has been linearized via a first-order Taylor expansion. This description, while significantly improved, is insufficient in particular when the incidence angle of the charged particle on the measurement surface is large. The assumption within the EKF that the noise is described by a Gaussian distribution 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 undergo bremsstrahlung as they pass 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 potential effects arising from a non-linear measurement model in track fitting.

This paper will explore a non-linear Kalman filter (NLKF) based on the Unscented Kalman filter (UKF) [14,15], as a method to mitigate some of the previously-discussed issues in charged particle reconstruction for high-energy nuclear and particle physics experiments. The UKF uses a set of discretely sampled points to parameterize the mean and covariance to account for non-linearities of the system and measurement models, and has been shown to have comparable performance to a second-order Gaussian filter. We investigate the performance of the NLKF in high energy physics use cases. We focus on the performance improvements observed through using the NLKF during the projection of the track state to the measurement compared to the EKF.

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) [16] 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) 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-practices 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 experiments, e.g. ATLAS [17], sPHENIX [18], and is being investigated as a potential track reconstruction library by a number of others.

Using its internal geometry and navigation model, ACTS provides a fast track simulation engine, based on the concept of the ATLAS Fast Track Simulation (Fatras) [19].[2] 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 multiple scattering is modelled as a random Gaussian noise and the energy loss is modelled with random numbers drawn from a Landau distribution according to the traversed material. 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 that describe 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 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 coordinates (x, y and z for position and t for time) 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 the momentum, p.[3] The d_x, d_y and d_z are constrained by $d_x^2 + d_y^2 + d_z^2 = 1$.

The local track parameters, l, are 6-dimensional and represented as:

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

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 the azimuthal and polar angles, respectively, describing the track direction in the global coordinate frame, and the q/p and t are the same as the global track parameters. The reference surfaces can include different types and shapes, including cylindrical and planar surfaces, and 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 because they are the 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.

[2] i.e. Fatras is significantly simplified with respect to a physics-based simulation such as Geant4 [20], resulting in orders-of-magnitude faster processing times.

[3] Following the convention of the ATLAS experiment, we use a right-handed Cartesian coordinate system with its origin at the nominal interaction point (IP) in the centre of the detector. The z-axis is along the beam pipe, and the x-axis points from the IP to the centre of the collider ring. Cylindrical coordinates (r, ϕ) are used in the transverse plane, $\phi \in [-\pi, \pi)$ being the azimuthal angle defined in the transverse x–y plane around the beam pipe. The rapidity is defined as $y = (1/2)\ln[(E + p_z)/(E - p_z)]$, while the pseudorapidity is defined in terms of the polar angle θ which is measured from the positive z-axis in an interval of $[0, \pi]$ as $\eta = -\ln\tan(\theta/2)$. $\eta = +\infty$ corresponds to the direction of the beam.

[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.

3. Track fitting with an extended Kalman filter

This section introduces EKF-based track fitting using the mathematical prescription following Ref. [3].

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. If we take the seed of the track fit as the first track state with index 0 and that there are K track states in total, this can be described by a track state propagation model:

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

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. It can be decomposed into two terms, $\eta_{k-1} = \eta^m_{k-1} + \eta^e_{k-1}$, where the former accounts for multiple scattering and the latter for energy loss due to ionization or radiation (in case of electrons).
- f_{k-1} is a function that encodes 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:

$$y_k = h_k(x_k) + \epsilon_k, \quad k = 1, \dots, K-1. \tag{4}$$

Here,

- 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 particle 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^m_{k-1} and measurement noise ϵ_k are assumed to be Gaussian distributions with zero means, and variances Q^m_k and V_k respectively. The process noise η^e_{k-1} is also assumed to follow a Gaussian distribution with non-zero mean (since it will act to reduce the momentum) and variance Q^e_k, even though it is known that it does not typically follow a Gaussian distribution.

As nuclear and particle experiments often have inhomogeneous magnetic fields, the f_{k-1} is evaluated to obtain x_k using the Runge–Kutta method [21] by numerically solving the second-order differential equations describing charged particles moving through magnetic fields. For example, the ATLAS experiment uses an adaptive Runge–Kutta–Nyström approach [22], which adapts the step size to minimize computational costs while ensuring that the estimation error remains below a set threshold. The y_k is obtained analytically by calculating the intersection of the track with the detector plane, which is described by h_k.

The covariance of x_k and y_k are obtained based on the first-order derivative of the track state at k with respect to the track state at $k-1$, F_{k-1}, and that of the measurement at k with respect to the track state at k, H_k, respectively, as follows:

$$\begin{aligned} F_{k-1} &= \partial f_{k-1}/\partial x_{k-1}, \quad k = 1, \dots, K-1 \\ H_k &= \partial h_k/\partial x_k, \quad k = 1, \dots, K-1, \end{aligned} \tag{5}$$

where the F_{k-1} is obtained numerically using the Runge–Kutta method and the H_k is calculated analytically using h_k and x_k.

The EKF has 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 where the k runs from 1 to K−1 used to update the track state vector, x and its covariance, C.

- Prediction:

$$\begin{aligned} x^{k-1}_k &= f_{k-1}(x_{k-1}) + \eta^e_{k-1}, \\ C^{k-1}_k &= F_{k-1} C_{k-1} F^T_{k-1} + Q^m_{k-1} + Q^e_{k-1}, \end{aligned} \tag{6}$$

  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. The F_{k-1} is evaluated at this stage using the filtered track state at $k-1$ and the predicted track state at state k.
- Filtering:

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

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

$$K_k = C^{k-1}_k H^T_k (V_k + H_k C^{k-1}_k H^T_k)^{-1}. \tag{8}$$

- Smoothing:

$$\begin{aligned} x^n_k &= x_k + A_k(x^n_{k+1} - x^k_{k+1}), \\ C^n_k &= C_k + A_k(C^n_{k+1} - C^k_{k+1}) A^T_k, \end{aligned} \tag{9}$$

  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:

$$A_k = C_k F^T_k (C^k_{k+1})^{-1}. \tag{10}$$

4. The non-linear Kalman filter

4.1. Non-linear effects in track reconstruction

A typical tracking detector at a particle collider is a cylindrical detector with concentric cylindrical layers around the collision point, which are oriented parallel to the beam direction, and disc layers on either side, which are oriented normal to the beam direction. This guarantees almost hermetic coverage of the phase space of the particles produced in the collisions, while complying with mechanical constraints and minimizing detector material. However, 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-linear effects using a straight line track model, i.e. the track direction is the same between state $k-1$ and k. A track propagates through a simplified detector (shown in Fig. 2) consisting of two parallel detector planes oriented normal to the beam direction without the presence of magnetic field and material effects. As a simple assumption, at state $k-1$, the track parameters ϕ and θ, i.e. the azimuthal angle ϕ and polar angle θ of the track direction, have non-zero uncertainty while the track parameters loc_0 and loc_1, i.e. the local coordinates of state in the cartesian frame of a plane (loc_0 is along the global x axis and loc_1 is along the global y axis), have zero uncertainty. The dependence of loc_0 and loc_1 at state k on the ϕ and θ at state k can be calculated analytically and is shown in Fig. 1. In this example, the dependence is closest to linear when ϕ and θ 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 as the absolute angles increase. This effect is particularly significant for the θ, which is highly correlated with the track incidence angle. As we will show, these non-linear effects can be addressed by the NLKF.

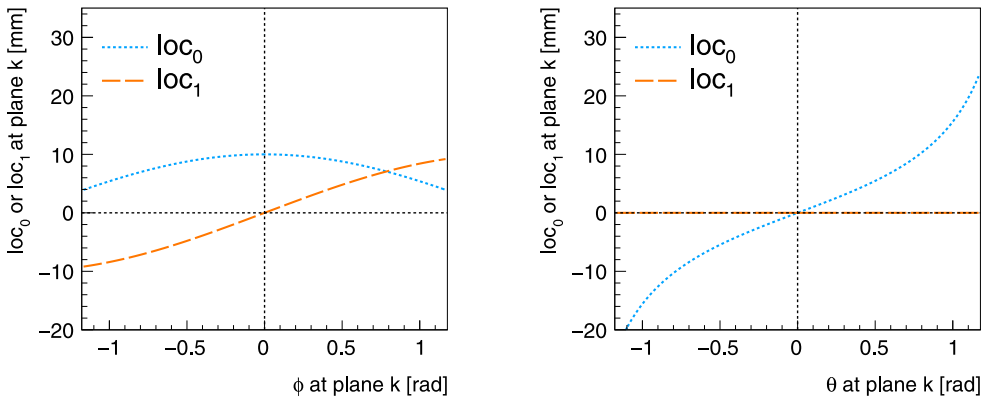

Fig. 1. Example of non-linear dependence of the local coordinate of the intersection of a track on detector plane k on the (left) azimuthal angle ϕ with $\theta = 45°$ and (right) polar angle θ with $\phi = 0°$ of the track on detector plane k for two parallel detector planes oriented normal to the beam direction and placed with a distance of 10 mm. The dependence of the loc_0 and loc_1 on the track direction are shown by the short-dashed blue and long-dashed orange lines, respectively. The vertical dashed line denotes the angle at zero and the horizontal line denotes the local coordinate at zero.

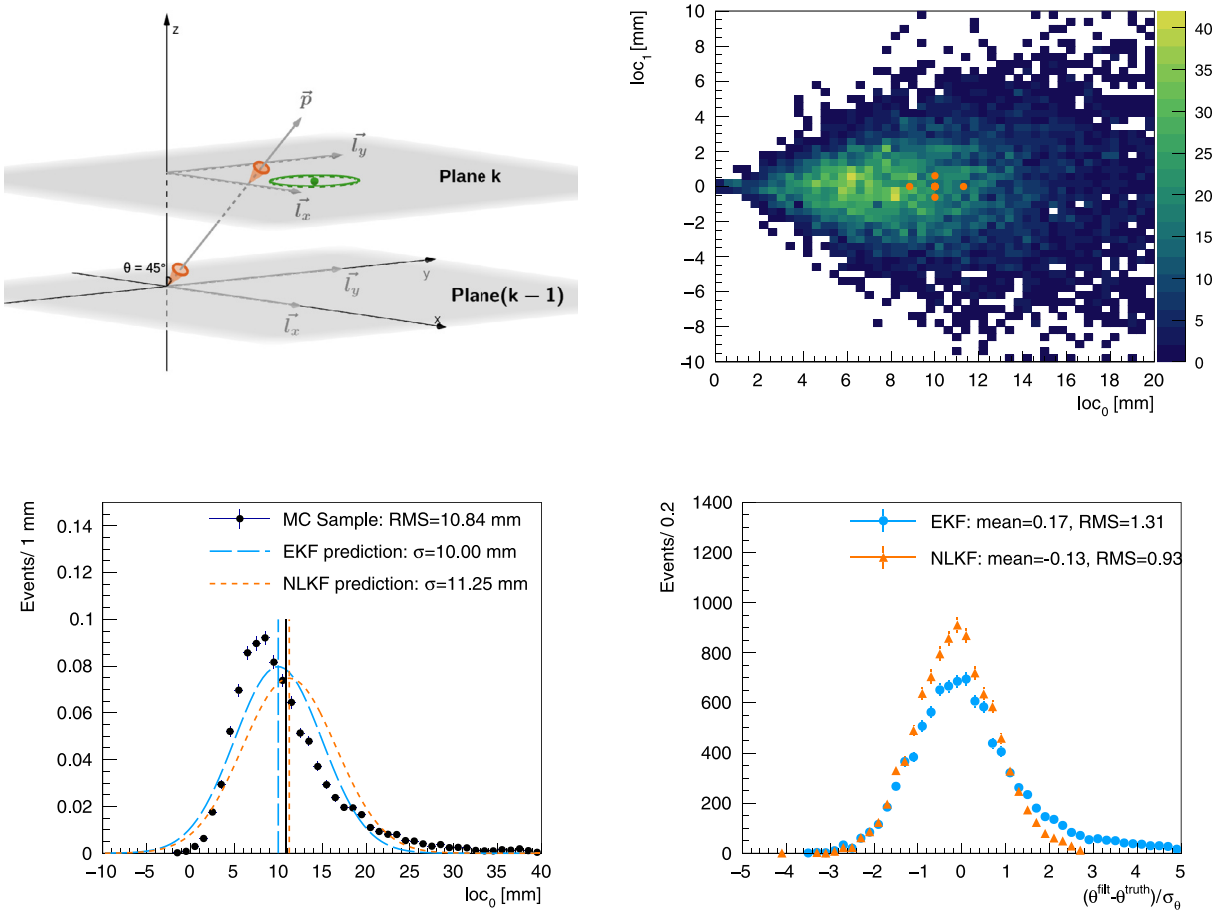

Fig. 2. Illustration of the impact of non-linear effects during track parameter transformation for a two layer detector without magnetic field. The detector planes are placed normal to global z axis with a distance of 10 mm. (Top left) A track with $\phi = 0°$ and $\theta = 45°$ intersects planes $k-1$ and k. Both ϕ and θ have uncertainty of 14.32° with their covariance denoted by the orange cones. The local coordinates of the track on plane $k-1$ has zero covariance. The green circle shows the covariance of a measurement located at the centre of the circle. (Top right) The scatter plot shows the local coordinates on plane k for 10,000 sampled tracks. The orange dots show the location of the sample points with NLKF (some of them overlap). (Bottom left) The normalized distribution of the coordinate loc_0 of the sampled tracks on plane k (black dots) and the Gaussian functions used to model the distribution by EKF (blue long-dashed line) and NLKF (orange short-dashed line). The black solid, blue long-dashed and orange short-dashed vertical lines denote the mean of the sample, EKF and NLKF, respectively. (Bottom right) Comparison of pull of the filtered θ using EKF (blue dots) and NLKF (orange triangles).

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, the method typically used to generate events corresponding to different physics processes, but using a set of representative sample points. For an N-dimensional track state vector x_{k-1} with covariance C_{k-1} at state $k-1$, $2N+1$ sample points are considered [14,15]. These comprise the nominal track state vector plus $2N$ vectors obtained by varying the nominal track state vector along

the direction of the eigenvectors of the covariance matrix. The magnitudes of the variations are chosen according to the eigenvalues of the covariance matrix. The eigenvectors and eigenvalues of the covariance matrix are obtained via Singular Value Decomposition (SVD) [23]. 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 ith column of U_{k-1} as u^i_{k-1} and the ith diagonal element of S_{k-1} as s^i_{k-1}, we define N sets of orthogonal shifting vectors δ^i_{k-1}:

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

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:

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

where γ is a scaling parameter defined as,

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

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

The sample points for x_{k-1} can be propagated to state k using the track model in Eq. (3),

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

and projected to a measurement point at state k using the measurement model in Eq. (4),

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

The mean, $\tilde{y}_k$, and covariance, P_k, of the projected track state are calculated as

$$\begin{aligned} \tilde{y}_k &= \sum_{i=0}^{2N} w^{(i)}_m \mathrm{Y}^{(i)}_k, \\ P_k &= \sum_{i=0}^{2N} w^{(i)}_c (\mathrm{Y}^{(i)}_k - \tilde{y}_k)(\mathrm{Y}^{(i)}_k - \tilde{y}_k)^T + V_k, \end{aligned} \tag{17}$$

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

$$T_k = \sum_{i=0}^{2N} w^{(i)}_c (\mathrm{X}^{(i)}_k - x^{k-1}_k)(\mathrm{Y}^{(i)}_k - \tilde{y}_k)^T. \tag{18}$$

In Eqs. (17) and (18), the weights $w^{(i)}_m$ and $w^{(i)}_c$ are defined as

$$\begin{aligned} w^{(0)}_m &= \frac{\lambda}{N+\lambda}, \quad i = 0, \\ w^{(0)}_c &= \frac{\lambda}{N+\lambda} + (1 - \alpha^2 + \beta), \quad i = 0, \\ w^{(i)}_m &= w^{(i)}_c = \frac{1}{2(N+\lambda)}, \quad i = 1, \dots, 2N, \end{aligned} \tag{19}$$

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. [24] 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 as follows

$$\begin{aligned} x_k &= x^{k-1}_k + K_k(m_k - \tilde{y}_k), \\ C_k &= C^{k-1}_k - K_k P_k K_k^T = C^{k-1}_k - T_k K_k^T. \end{aligned} \tag{21}$$

4.3. Implementation of NLKF in ACTS

In this paper, we apply only the non-linear corrections during the projection of the track state to the measurement point. In such cases, the propagation from state $k-1$ to state k is performed using the adaptive Runge–Kutta method in the presence of a magnetic field as in EKF and the sample points considered for x_k are passed to h_k in Eq. (16) to obtain $\mathrm{Y}^{(i)}_k$.

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, where the track state is constrained to the measurement plane, and projecting the local track parameters to the measurement with 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-1}_k and its covariance C^{k-1}_k represented by the local track parameters at state k using Kalman filtering formulae.

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 at state k to local track parameters at state k is replaced by the transformation of the 17 sets,[5] of global track parameters at state k to the local track parameters at state k. Eq. (16) is used and the corrected local track parameters ($x^{k-1,\mathrm{c}}_k$, the superscript c here and thereafter denotes 'corrected') and associated covariance ($C^{k-1,\mathrm{c}}_k$) are calculated according to Eq. (17). In ACTS, the covariance in Eq. (18) between the local track parameters and the measurement is part of the covariance matrix of the local track parameters, i.e.

$$T_k = C^{k-1,\mathrm{c}}_k H_k^T, \tag{22}$$

and

$$P_k = H_k C^{k-1,\mathrm{c}}_k H_k^T + V_k. \tag{23}$$

The Kalman gain formulae for the EKF and NLKF are identical:

$$K_k = T_k P_k^{-1} = C^{k-1,\mathrm{c}}_k H_k^T (V_k + H_k C^{k-1,\mathrm{c}}_k H_k^T)^{-1}, \tag{24}$$

and Eq. (21) for Kalman filtering becomes

$$\begin{aligned} x^{\mathrm{c}}_k &= x^{k-1,\mathrm{c}}_k + K_k(m_k - H_k x^{k-1,\mathrm{c}}_k), \\ C^{\mathrm{c}}_k &= C^{k-1,\mathrm{c}}_k - T_k K_k^T = (1 - K_k H_K) C^{k-1,\mathrm{c}}_k. \end{aligned} \tag{25}$$

The updated track state at state k with track position constrained to the measurement plane is then represented as global track parameters before being propagated to the next track state at $k+1$ using the adaptive Runge–Kutta method. The state vector of the global track parameters is transformed from the state vector of the local track parameters representing the track state analytically. The state covariance is transformed using the first-order derivative of the representing global track parameters at state k with respect to the representing local track parameters at state k.

The smoothing formulae for NLKF are:

$$\begin{aligned} x^{n,\mathrm{c}}_k &= x^{\mathrm{c}}_k + A_k(x^{n,\mathrm{c}}_{k+1} - x^{k,\mathrm{c}}_{k+1}), \\ C^{n,\mathrm{c}}_k &= C^{\mathrm{c}}_k + A_k(C^{n,\mathrm{c}}_{k+1} - C^{k,\mathrm{c}}_{k+1}) A_k^T, \end{aligned} \tag{26}$$

where

$$A_k = C^{\mathrm{c}}_k F_k^T (C^{k,\mathrm{c}}_{k+1})^{-1}. \tag{27}$$

As in the formulae for EKF, the index k runs from 1 to $K-1$ in the above formulae.

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

4.4. Comparison of the EKF and NLKF for track fitting

Fig. 2 illustrates the impact of the non-linear effects on track parameter propagation and Kalman filtering procedure using the configuration shown in the top left panel. With a straight track model and no material effects, the track direction at plane k has the same value and uncertainty as those at plane $k-1$. The uncertainty of the intersection of the track at plane k is correlated with the uncertainty of the track direction at plane k. Given the uncertainty of the track direction, the 2-dimensional distribution of the local coordinates of the intersection of the track on plane k obtained using 10,000 sampled tracks is shown in the top right panel. It can be seen that the distribution of loc_0, shown in the bottom left panel, is non-Gaussian with positive skew. Both EKF and NLKF use Gaussian functions to model the distribution. The mean and width of the Gaussian functions differ between EKF and NLKF. With NLKF, the mean and width are closer to the mean and *Root-Mean-Square* (RMS) of the MC sample.

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

$$\text{pull}_v = \frac{v^{\text{fit}} - v^{\text{truth}}}{\sigma_v}. \tag{28}$$

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. The widths of the distributions are estimated using robust RMS evaluations rather than Gaussian fits. The mean and the 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 one.

5. Performance studies

The performance of the NLKF as implemented in Section 4.3 is evaluated using the Open Data Detector (ODD) [25]. The layout of the ODD is shown in Fig. 3. It consists of a pixel detector and two strip detectors with different intrinsic resolutions and uses a realistic material model using the DD4hep [26] detector description tool. The ODD is immersed in a solenoidal magnetic field of 2 Tesla centred 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 and because the most probable use of the NLKF would be for high-precision track reconstruction applications. 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 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 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 emulate the impact of detector resolution. One- and two-dimensional measurements in the local coordinate frames of the detector 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.

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

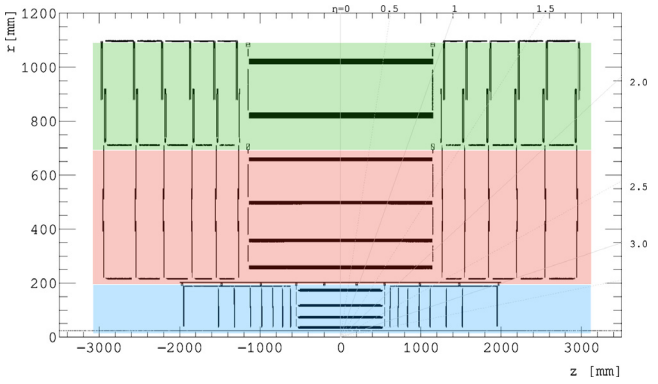

Fig. 3. Schematic layout of the ODD silicon tracking detector projected into the $z-r$ plane. The beam interaction would occur at $z = 0, r = 0$. The location of the pixel detector is shown in blue, the two strip detectors with different intrinsic resolution are shown in red and green, where the inner strip detector (red) has better resolution than the outer strip detector (green).

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. The x axes of the local coordinate frame of the strip detectors are parallel to the global x–y plane.

Subdetectors	σ_x [μm]	σ_y [μm]
Pixel	15	15
Inner strip	43	–
Outer strip	72	–

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

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

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 distributions 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 in current nuclear and particle physics experiments.

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 rad, a value chosen to balance between improved physics performance in the most relevant cases and increased computation time when the correction is applied. With such a threshold, the fraction of track states with the non-linear correction applied for the tracks in the region $|\eta| < 1$ is about 55% and increases to approximately 95% for the tracks in the region $2.5 < |\eta| < 3.0$. The NLKF performance is found to be insensitive to the tuning parameter α so a fixed value of $\alpha = 0.1$ is used.

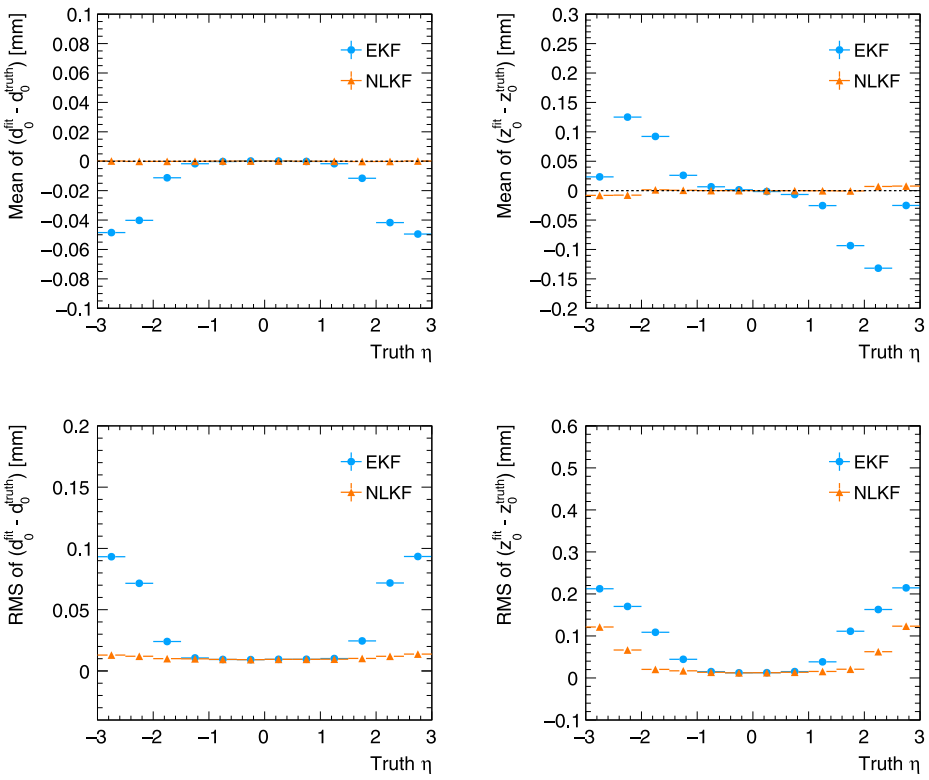

Fig. 4. The mean (top) and RMS (bottom) of the residual of fitted perigee track parameters d_0 (left) and z_0 (right) parameterized as a function of simulated particle η ($20 < p_T < 100$ GeV) for the ODD with the presence of a solenoidal magnetic field of 2 T and material effects. The blue dots and orange triangles show the results obtained using EKF and NLKF, respectively. The dashed horizontal lines in the upper panel denote the expected mean of the residuals. The mean and RMS are calculated using the residual in the range of [−0.5, 0.5] mm.

5.1. Track parameter estimation

The mean and RMS of the residuals, defined as $v^{\text{fit}} - v^{\text{truth}}$, and the pulls, defined in Eq. (28), of the perigee track parameters obtained by propagating the smoothed track parameters at the first measurement plane to the perigee plane 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, a pull would have a mean of zero and an RMS of one and the residuals would have means of zero and an RMS corresponding to 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 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 $p_T > 20$ GeV are shown in the upper panel of Fig. 4. The mean estimated using the EKF is biased from zero at higher $|\eta|$ bins due to non-linear effects in this region. No such biases are observed when the NLKF is used. 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 $p_T > 20$ GeV is shown in the lower panel of Fig. 4. The resolution gets worse in the higher $|\eta|$ bins and the effect is most significant for the EKF. The NLKF improves the resolution by up to 80% at higher $|\eta|$ bins compared to the EKF, given the resolution of the seed track parameters[7] in Table 2. All track parameters are studied, and similar improvements for ϕ and θ are observed, while there is no improvement in the resolution of q/p. The impact of non-linear effects depends on the resolution of the track parameters and hence, the improvement of NLKF with respect to EKF also depends on this, i.e. the worse the resolution of the initial track parameter estimate, the greater the improvement. For example, if the width of the Gaussian used to smear the ϕ and θ is changed from 1° to 0.5° and 2°, the improvement of the residual for d_0 in the region $1.5 < |\eta| < 2.5$ changes from 79% to 58% and 84%, respectively.

Fig. 5 shows the RMS of the pulls of the perigee track parameters as a function of η for simulated particles with $p_T > 20$ 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 2.2 and smallest for q/p. The deviation is significantly reduced using the NLKF, i.e. the RMS for all track parameters is below 1.2 in the entire η range. The NLKF improves the RMS of the pull for d_0 in the $1.5 < |\eta| < 2.5$ by 55%, which becomes 45% and 58% if the width of the Gaussian used to smear the ϕ and θ is changed to 0.5° and 2°.

Fig. 6 shows the RMS of the pulls of the impact parameters for simulated particles in the range of $1.0 < |\eta| < 2.5$ as a function of p_T. This η range was selected because non-linear effects are significant for these values. The RMS of the pulls for d_0 is smaller for lower p_T tracks using the EKF. 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 $p_T > 3$ GeV and 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. For NLKF, the dependence of the pulls on track p_T is mainly driven by material effects, while for EKF, non-linear effects are dominant. As the material effects, modelled using Gaussian distributions, are more significant at lower p_T, the pull distributions of

[7] As discussed earlier, results are shown for a scenario consistent with current experiments.

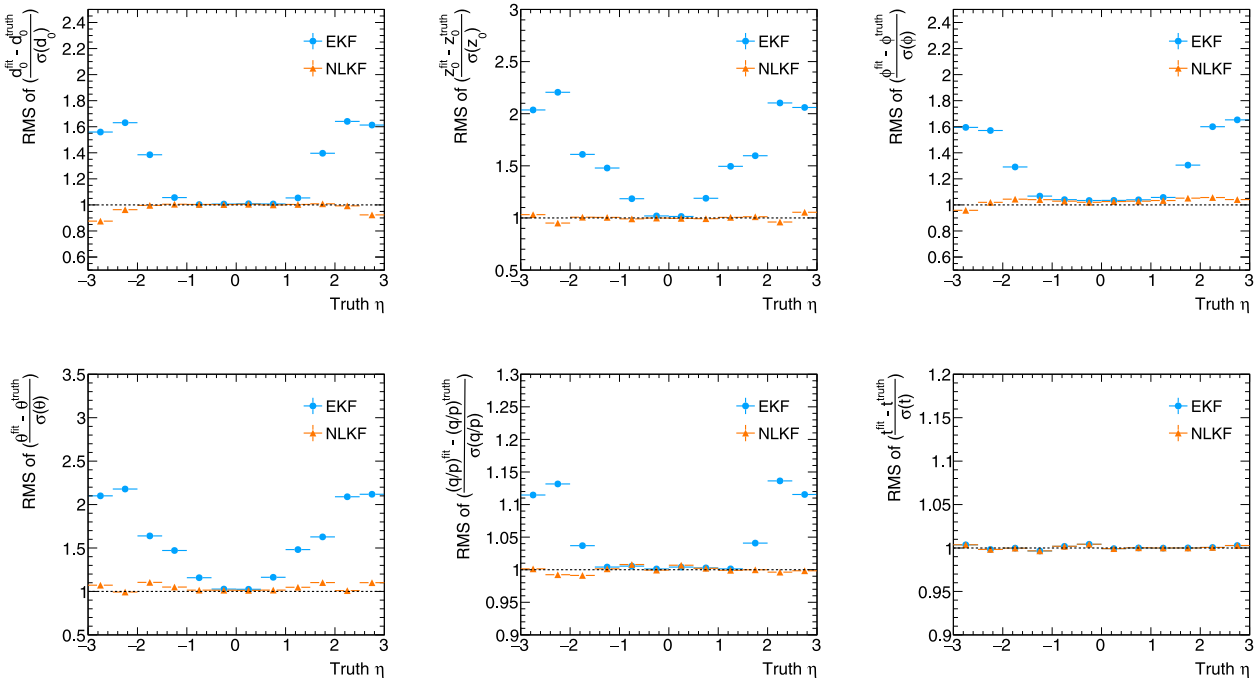

Fig. 5. The RMS of the pull of fitted perigee track parameters d_0, z_0, ϕ, θ, q/p and t parameterized as a function of the simulated particle η ($20 < p_T < 100$ GeV) for the ODD with the presence of a solenoidal magnetic field of 2 T and material effects. The blue dots and orange triangles show the results obtained using EKF and NLKF, respectively. The dashed horizontal lines denote the expected RMS of the pulls. The RMS is calculated using the pulls in the range of [−5, 5].

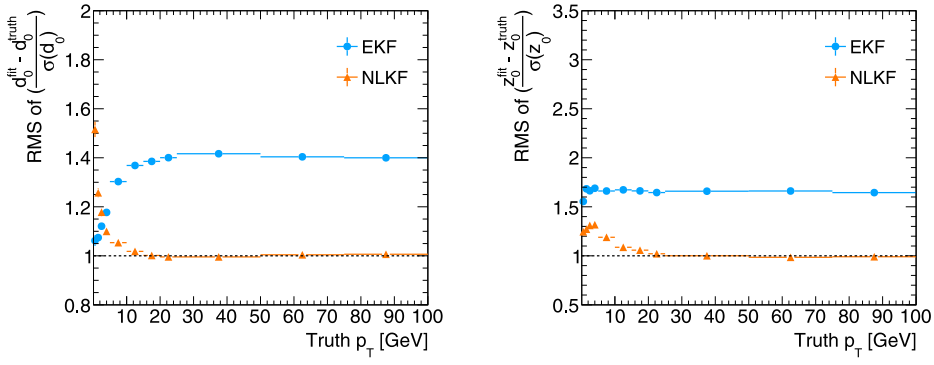

Fig. 6. The RMS of the pull of fitted perigee track parameters d_0 (left) and z_0 (right) parameterized as a function of simulated particle p_T ($1.0 < |\eta| < 2.5$) for the ODD with the presence of a solenoidal magnetic field of 2 T and material effects. The blue dots and orange triangles show the results obtained using EKF and NLKF, respectively. The dashed horizontal lines denote the expected RMS of the pulls. The RMS is calculated using the pulls in the range of [−5, 5].

EKF become closer to Gaussians. This can result in the pull RMS of EKF being closer to 1 at lower p_T.

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 1,000 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. 7 shows the track fitting time in HS06 [27] ×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 ×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. 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 applied the principle of an Unscented Kalman filter to implement a Non-linear Kalman filter for charged particle reconstruction, which uses a set of discretely sampled points to account for non-linear effects during the projection of the track parameters to the measurement.

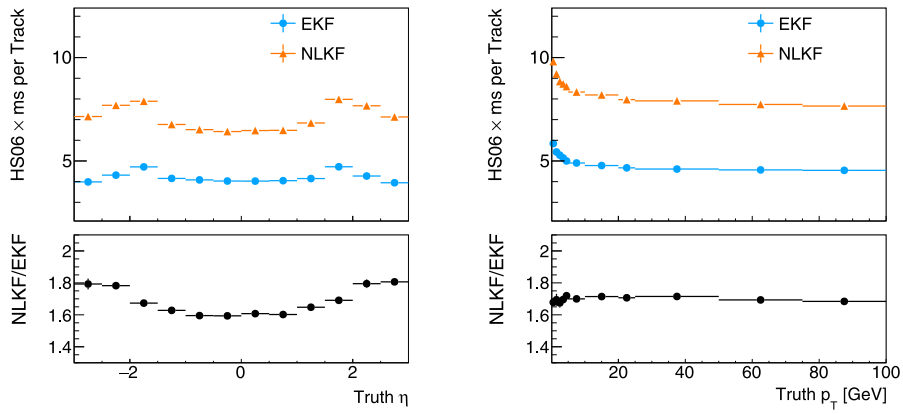

Fig. 7. (Left) A comparison of the fitting time per track as a function of the simulated particle η ($20 < p_T < 100$ GeV) between EKF and NLKF for the ODD at a solenoidal magnetic field of 2 T. (Right) A comparison of the track fitting time per track as a function of simulated particle p_T ($1.0 < |\eta| < 2.5$) between EKF and NLKF for the ODD at a solenoidal magnetic field of 2 T. (Top panels) The fitting time in HS06 ×ms per track. The blue dots and orange triangles show the results obtained using EKF and NLKF, respectively. (Bottom panels) The ratio of fitting time per track between NLKF and EKF.

We tested the performance of our NLKF algorithm using the ODD. The NLKF yields residuals for all track parameters with a mean of zero for all values of η. In addition, the RMS of the residuals are reduced for most track parameters. The level of improvement depends on the resolution of the starting track parameters, but the results obtained using NLKF are more stable, with a reduced dependence on the resolution of the starting parameters. The improvement 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 in high energy nuclear and particle physics experiments by accounting for non-linear effects. Its use can be warranted in applications where the precision of the track parameters is particularly important. Other approaches [23,24] may also be appropriate for addressing these issues, and their suitability with respect to both the EKF and NLKF could be investigated in future studies.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The code used for this research is available open source.

Acknowledgements

Xiaocong Ai, Nicholas Styles acknowledge support from DESY (Hamburg, Germany), a member of the Helmholtz Association HGF. Heather Gray acknowledges support by the National Science Foundation, USA under Cooperative Agreement OAC-1836650.

Funding

This work was funded by the National Science Foundation under Cooperative Agreement OAC-1836650.

Code availability

The code used for this research is available open source [28].

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.nima.2023.168041.

References

[1] R.E. Kalman, A New Approach to Linear Filtering and Prediction Problems, J. Basic Eng. 82 (1) (1960) 35–45, http://dx.doi.org/10.1115/1.3662552.
[2] P. Abreu, et al., DELPHI Collaboration Collaboration, Performance of the DELPHI detector, Nucl. Instrum. Methods A 378 (1996) 57–100, http://dx.doi.org/10.1016/0168-9002(96)00463-9.
[3] R. Frühwirth, Application of Kalman filtering to track and vertex fitting, Nucl. Instrum. Meth. A262 (1987) 444–450, http://dx.doi.org/10.1016/0168-9002(87)90887-4.
[4] A. Strandlie, R. Frühwirth, Track and vertex reconstruction: From classical to adaptive methods, Rev. Modern Phys. 82 (2010) 1419–1458, http://dx.doi.org/10.1103/RevModPhys.82.1419.
[5] H. Rauch, F. Tung, C. Striebel, Maximum likelihood estimates of linear dynamical systems, AIAA 3 (1965) 1445, http://dx.doi.org/10.2514/3.3166.
[6] P. Billoir, Progressive track recognition with a Kalman-like fitting procedure, Comput. Phys. Comm. (ISSN: 0010-4655) 57 (1) (1989) 390–394, http://dx.doi.org/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 (ISSN: 0168-9002) 294 (1) (1990) 219–228, http://dx.doi.org/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 (ISSN: 0168-9002) 395 (2) (1997) 169–184, http://dx.doi.org/10.1016/S0168-9002(97)00705-5.
[9] F.E. Daum, Extended Kalman filters, in: Encyclopedia of Systems and Control, Springer London, London, ISBN: 978-1-4471-5058-9, 2015, pp. 411–413, http://dx.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, Comput. Phys. Comm. (ISSN: 0010-4655) 110 (1) (1998) 80–86, http://dx.doi.org/10.1016/S0010-4655(97)00157-4.
[11] R. Frühwirth, A Gaussian-mixture approximation of the Bethe–Heitler model of electron energy loss by bremsstrahlung, Comput. Phys. Comm. (ISSN: 0010-4655) 154 (2) (2003) 131–142, http://dx.doi.org/10.1016/S0010-4655(03)00292-3.
[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 TeV, Eur. Phys. J. C 79 (2019) 639, http://dx.doi.org/10.1140/epjc/s10052-019-7140-6, arXiv:1902.04655.

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

[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, SPIE, 1997, pp. 182–193, http://dx.doi.org/10.1117/12.280797.

[15] S. Julier, J. Uhlmann, Unscented filtering and nonlinear estimation, Proc. IEEE 92 (3) (2004) 401–422, http://dx.doi.org/10.1109/JPROC.2003.823141.

[16] X. Ai, C. Allaire, N. Calace, A. Czirkos, M. Elsing, I. Ene, R. Farkas, L.-G. Gagnon, R. Garg, P. Gessinger, H. Grasland, H.M. Gray, C. Gumpert, J. Hrdinka, B. Huth, M. Kiehn, F. Klimpel, B. Kolbinger, A. Krasznahorkay, R. Langenberg, C. Leggett, G. Mania, E. Moyse, J. Niermann, J.D. Osborn, D. Rousseau, A. Salzburger, B. Schlag, L. Tompkins, T. Yamazaki, B. Yeo, J. Zhang, A common tracking software project, Comput. Softw. Big Sci. (ISSN: 2510-2044) 6 (1) (2022) 8, http://dx.doi.org/10.1007/s41781-021-00078-8.

[17] ATLAS Collaboration, Software Performance of the ATLAS Track Reconstruction for LHC Run 3, Tech. rep., CERN, Geneva, 2021, All figures including auxiliary figures are available at https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-PHYS-PUB-2021-012.

[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, Comput. Softw. Big Sci. (ISSN: 2510-2044) 5 (1) (2021) 23, http://dx.doi.org/10.1007/s41781-021-00068-w.

[19] K. Edmonds, S. Fleischmann, T. Lenz, C. Magass, J. Mechnich, A. Salzburger, The Fast ATLAS Track Simulation (FATRAS), Tech. Rep, CERN, Geneva, 2008.

[20] S. Agostinelli, et al., GEANT4 Collaboration Collaboration, GEANT4: A Simulation toolkit, Nucl. Instrum. Methods A 506 (2003) 250–303, http://dx.doi.org/10.1016/S0168-9002(03)01368-8.

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

[22] 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, J. Instrum. 4 (04) (2009) P04016, http://dx.doi.org/10.1088/1748-0221/4/04/p04016.

[23] 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.

[24] 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.

[25] C. Allaire, P. Gessinger, J. Hdrinka, M. Kiehn, F. Kimpel, J. Niermann, A. Salzburger, S. Sevova, OpenDataDetector, 2021, http://dx.doi.org/10.5281/zenodo.4674401.

[26] M. Petrič, M. Frank, F. Gaede, S. Lu, N. Nikiforou, A. Sailer, Detector Simulations with DD4hep, J. Phys. Conf. Ser. 898 (2017) 042015, http://dx.doi.org/10.1088/1742-6596/898/4/042015.

[27] HEP-SPEC06 benchmark, 2021, https://w3.hepix.org/benchmarking.html.

[28] ACTS on github, 2021, https://github.com/acts-project/acts/releases/tag/v19.0.0.