顶点拟合和运动学拟合算法

Liang Yan
Fudan University

Workshop of Computing Software and Technologies in Particle Physics Experiment
June 10-12th, 2023
Qingdao, China

Outline

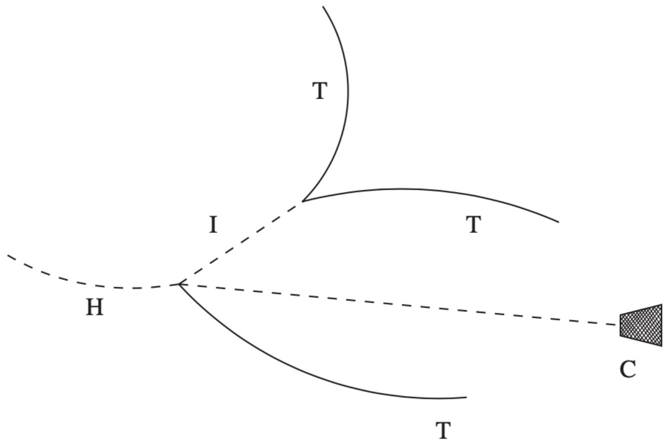
- **Introduction**
- **Algorithm**
- **Current Status and Limits**
- **Global fits**
- **Summary**

Introduction

What is Vertex Fitting and Kinematic Fitting

- **The mathematic process of improving the measurements according to the physics laws in the interaction and decay of the particles.**

- **For example: In the decays, the final particles should come from the common decay point (vertex fitting), the momentum of the final states equals to initial state (kinematic fitting).**



- **Physical requirements are provided through constraints in the form of an equation: each track for 2 constraints (vertex fitting), four-momentum conversation for 4 constraints (kinematic fitting).**

Algorithm

Least-squares fit

- **The constraints are expressed as χ^2 contribution, a function of the parameters, denoted by the vector x**

- **The solution to the fit is the value of x that minimizes the total χ^2**

  - ✓ **Lagrange multiplier method**

  - ✓ **Kalman filter method**

Method of least squares

input measurements

model: ‘prediction’

$$\chi^2 = \sum_i \left(\frac{m_i - h_i(x)}{\sigma_i} \right)^2$$

“estimated RMS of parent distribution of error of m_i”

Least-squares-estimator: value of x for which χ^2 is minimal $\left.\frac{\mathrm{d}\chi^2}{\mathrm{d}x}\right|_{\hat{x}} = 0$

Matrix notation

vector of residuals

$$\chi^2 = \sum_i \left(\frac{m_i - h_i(x)}{\sigma_i} \right)^2 = (m - h(x))^T \; V^{-1} \; (m - h(x))$$

measurement covariance matrix.
often diagonal

Linear least squares estimator

- consider a **linear model**

$$h(x) = h_0 + H\,x$$

- least squares condition:

$$\frac{d\chi^2}{dx} = -2\,H^T V^{-1}\,(m - h_0 - Hx) = 0$$

- solution:

$$\hat{x} = \left(H^T V^{-1} H\right)^{-1} H^T V^{-1}\,(m - h_0)$$

Linear least squares estimator (LSE)

Linear least squares estimator

- closer inspection:

$$\hat{\boldsymbol{x}} = \left(\boldsymbol{H}^T \boldsymbol{V}^{-1} \boldsymbol{H}\right)^{-1} \boldsymbol{H}^T \boldsymbol{V}^{-1} \left(\boldsymbol{m} - \boldsymbol{h}_0\right)$$

M-dim vector

$$\left(\boldsymbol{H}^T \boldsymbol{V}^{-1} \boldsymbol{H}\right) = \frac{1}{2} \frac{\mathrm{d}^2 \chi^2}{\mathrm{d}\boldsymbol{x}^2}\bigg|_{\boldsymbol{x}=0}$$

MxM symmetric matrix

$$\boldsymbol{H}^T \boldsymbol{V}^{-1} \left(\boldsymbol{m} - \boldsymbol{h}_0\right) \quad \leftrightarrow \quad \frac{1}{2} \frac{\mathrm{d}\chi^2}{\mathrm{d}\boldsymbol{x}}\bigg|_{\boldsymbol{x}=0}$$

M-dim vector

$$\mathrm{var}\left(\hat{\boldsymbol{x}}\right) \equiv \boldsymbol{C} = \left(\boldsymbol{H}^T \boldsymbol{V}^{-1} \boldsymbol{H}\right)^{-1}$$

Non-linear models: Newton-Raphson

1. expand around initial solution

$$h(x) \;=\; h(x_0) + H(x - x_0)$$

$$H \;=\; \left.\frac{\mathrm{d}h(x)}{\mathrm{d}x}\right|_{x_0}$$

2. compute a new value for x

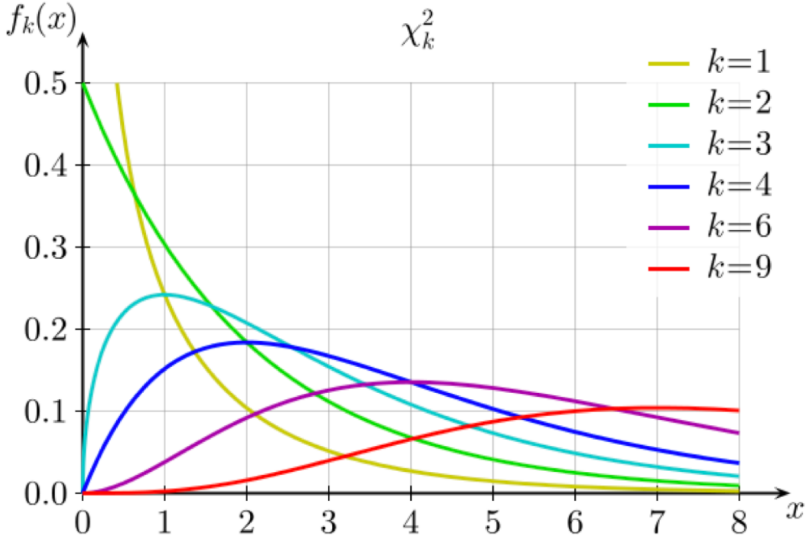
$$\hat{x} \;=\; x_0 + \underbrace{\left(H^T V^{-1} H\right)^{-1}}_{\left.\frac{1}{2}\frac{\mathrm{d}^2\chi^2}{\mathrm{d}x^2}\right|_{x_0}} \;\underbrace{H^T V^{-1}(m - h(x_0))}_{\left.\frac{1}{2}\frac{\mathrm{d}\chi^2}{\mathrm{d}x}\right|_{x_0}}$$

3. use x-hat as new expansion point and iterate until $\Delta\chi^2$ is small

χ^2 and degree of freedom

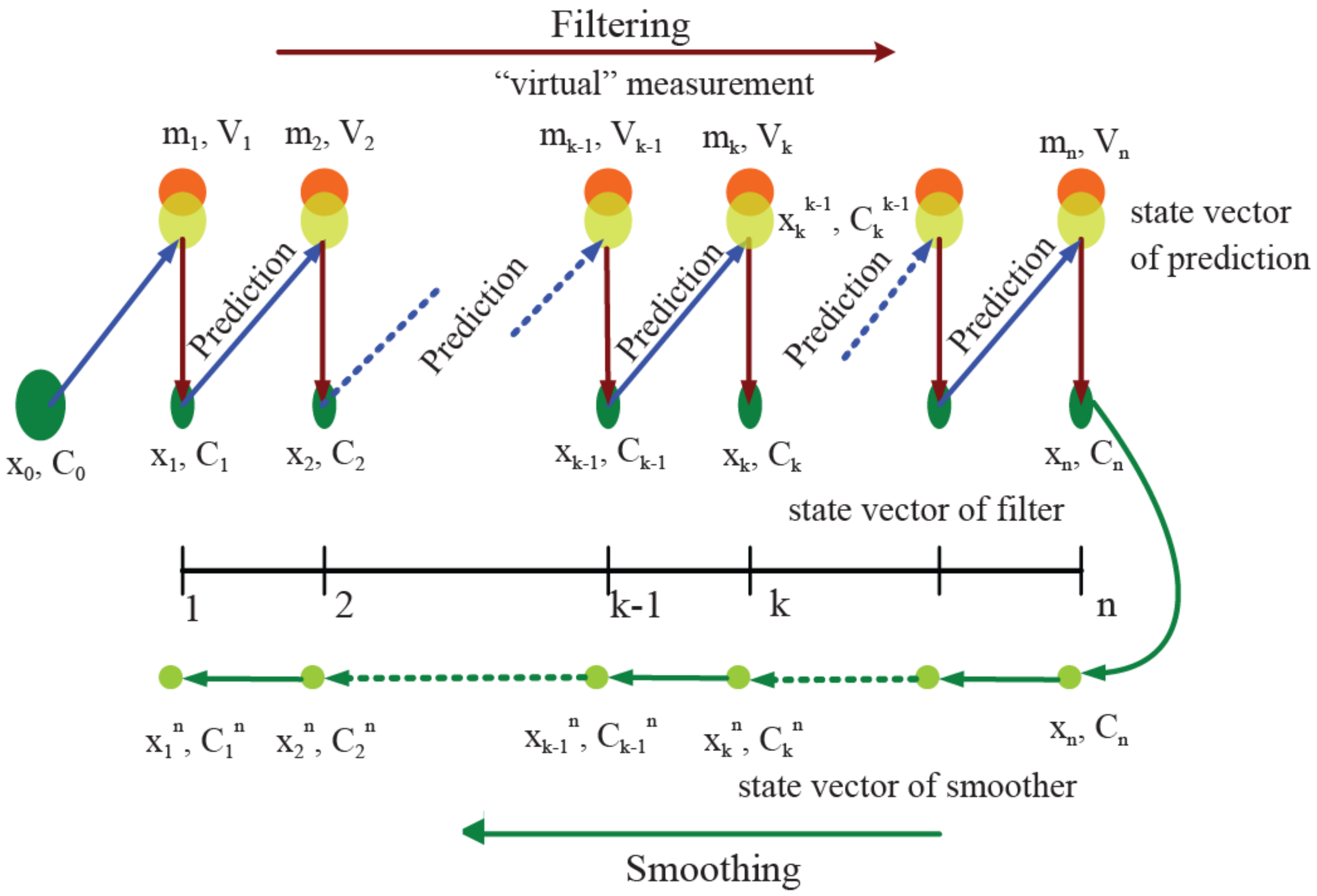
- consider chi-square for **x = x^true^**

$$\chi^2 = \sum_{i=0}^{N} \left(\frac{m_i - h_i(x_{\text{true}})}{\sigma_i} \right)^2$$



- if model is correct, then each entry in sum has zero mean and unit variance
  - sum is distributed according to “chi2 with N degrees-of-freedom”
  - expectation value is **N**
- chi-square for least squares estimate x = x-hat is *smaller*
  - for M fit parameters, expectation value is ndof = **N – M**
  - chi2/ndof often used to select good tracks, good vertices etc

Kalman filter method (progressive fit)



Kalman filter

- consider splitting the chi-square in two parts

$$\chi^2 = (m_1 - h_1(x))^T V_1^{-1} (m_1 - h_1(x)) + (m_2 - h_2(x))^T V_2^{-1} (m_2 - h_2(x))$$

- suppose that we have already found a solution by minimizing the first part:

$$C_1 = (H_1^T V_1^{-1} H_1)^{-1}$$

$$x_1 = x_0 + C_1 H_1^T V_1^{-1} (m_1 - h_1(x_0))$$

- can we reuse this when computing the minimum for the full chi-square?

Kalman filter

- solution can be written in many ways, but if the dimension of m_2 is small, a particularly efficient form is ("Kalman gain-matrix formalism")

$$x = x_1 + K(m_2 - h_2(x_1))$$

with

$$K = C_1 H_2^T (V_2 + H_2 C_1 H_2^T)^{-1}$$

"Kalman gain matrix"

- if m has small dimension, then we only invert small matrices!

Covariance matrix in the Kalman filter

- there exist different expressions for the covariance matrix

$$C = (1 - KH_2)\, C_1$$

fast, but unstable

$$C = (1 - KH_2)\, C_1\, (1 - KH_2)^T + KV_2K^T$$

stable but slow

$$C = (1 - 2KH_2)\, C_1 + K\,(V_2 + H_2C_1H_2^T)\, K^T$$

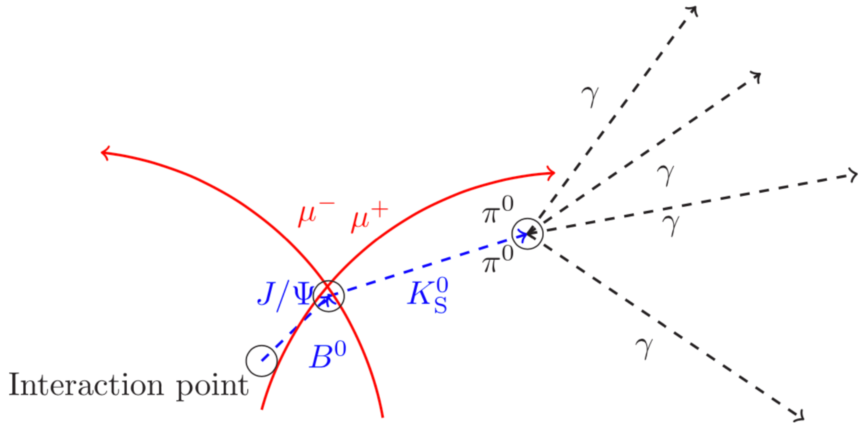
stable and fast

- expressions differ in computation speed and computation accuracy
- be careful which one you pick!

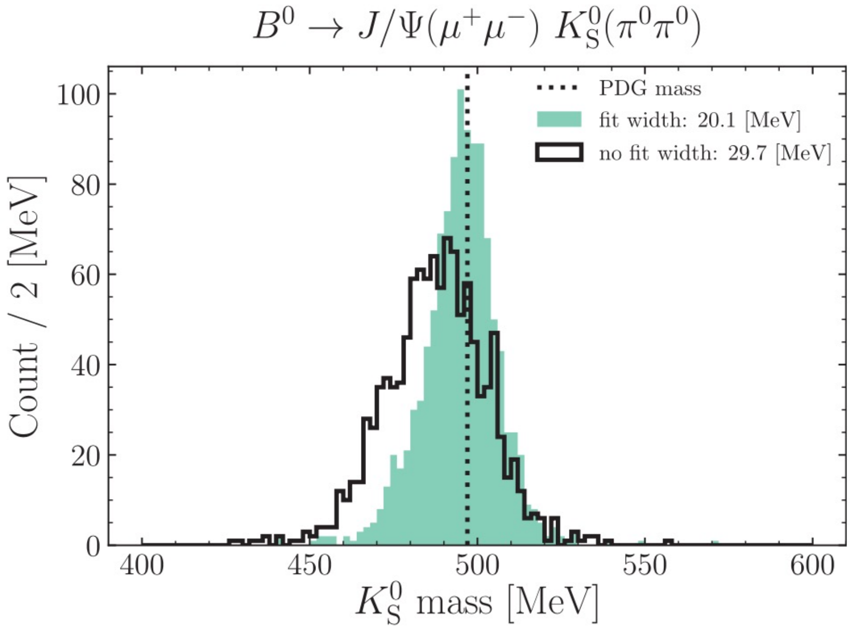
Current Status and Limits

Current: Leaf by Leaf

- ✓ In high-energy physics experiments decay reactions that proceed via intermediate metastable states are usually reconstructed by following a bottom–up approach.

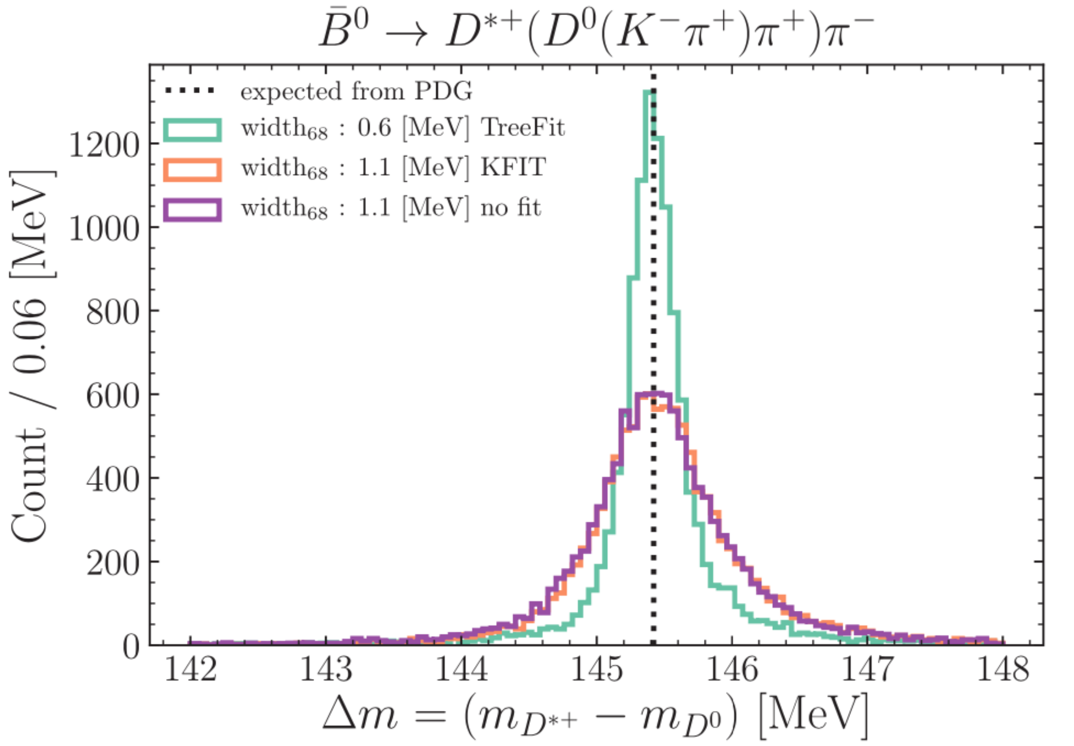


- ✓ One starts by extracting the parameters of those decay vertices from which the reconstructed final state particles emerge and uses the intermediate 'composite' particles for the reconstruction of upstream decays.

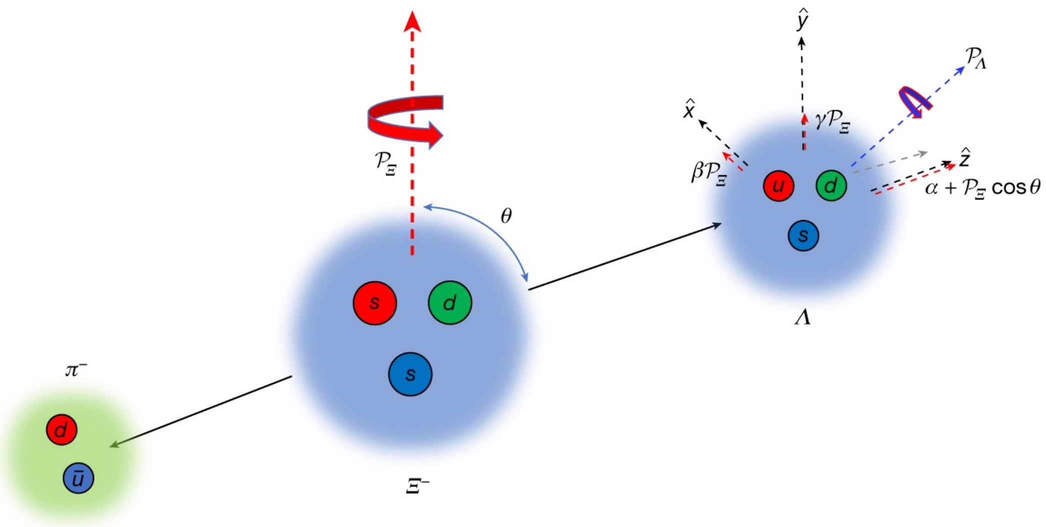


- ✓ Disadvantage: constraints that are upstream of a decay vertex do not contribute to the knowledge of the parameters of the vertex.

Other examples



Not only Kinematics fitting and vertex fitting are separated, but also the process with continuous decay vertices are separated.



The update of Lambda could not impact the proton and pion.

Improve the precision of hyperon parameter measurements.

Global fit

Decay Tree Fitter

Fits the whole decay tree at once. Vertices, known masses, measured tracks & neutrals and beam/target measurement (4C) are included as constraints. The common approach is the χ^2 fit with Lagrange multipliers.

->Very large parameter space and large matrices have to be inverted!

Solution: Kalman Filter approach

- ✓ Calculation of χ^2 is linearized
- ✓ Each constraint to the fit enters as one separate, scalar term
- ✓ Measurements are constraints and are treated similar to, e.g. four-momentum conservations

Constraints

$$h \equiv \begin{pmatrix} d_0 \\ \phi_0 \\ \omega \\ z_0 \\ \tan\lambda \end{pmatrix} = \begin{pmatrix} (p_{t0} - p_t)/aq \\ \mathrm{atan2}(p_{y0}, p_{x0}) \\ aq/p_t \\ z - lp_z/p_t \\ p_z/p_t \end{pmatrix}$$

Tracks

$$h = \begin{pmatrix} x + \theta p_x \\ y + \theta p_y \\ z + \theta p_z \\ \sqrt{p_x^2 + p_y^2 + p_z^2} \end{pmatrix}$$

Photons

In principle, θ can be added to the photon parameter list and extracted from the fit. Since this parameter is not very interesting, it is preferable to eliminate it.

$$r'(x) \equiv \begin{pmatrix} (x_{\mathrm{clus}} - x)p_y - (y_{\mathrm{clus}} - y)p_x \\ (x_{\mathrm{clus}} - x)p_z - (z_{\mathrm{clus}} - z)p_x \\ E_{\mathrm{clus}} - \sqrt{p_x^2 + p_y^2 + p_z^2} \end{pmatrix}.$$

Fit Parameters

(3) Primary vertex
(3) Secondary vertices
(3) Final state momenta
(4) Composite's four-momenta

Constraints

(5) Tracks (helix parameters)
(3) Clusters
(4) Initial four-momentum
(4) Internal four-momentum conservations
(1) Mass constraints

From Ralf Kliemt @ PANDA

Fit Parameters

- (3) Primary vertex
- (3) Secondary vertices
- (3) Final state momenta
- (4) Composite's four-momenta

Constraints

- (5) Tracks (helix parameters)
- (3) Clusters
- (4) Initial four-momentum
- (4) Internal four-momentum conservations
- (1) Mass constraints

Example

$$\psi(2S) \to J/\psi\ \pi^+\pi^-$$
$$\hookrightarrow \mu^+\mu^-$$

23 Parameters:

- 12 4 Final State Particles
- 8 2 Composites
- 3 Primary vertex

32 Constraints:

- 20 4 Helices
- 8 2 P4-Conservations
- 4 Beam-Target

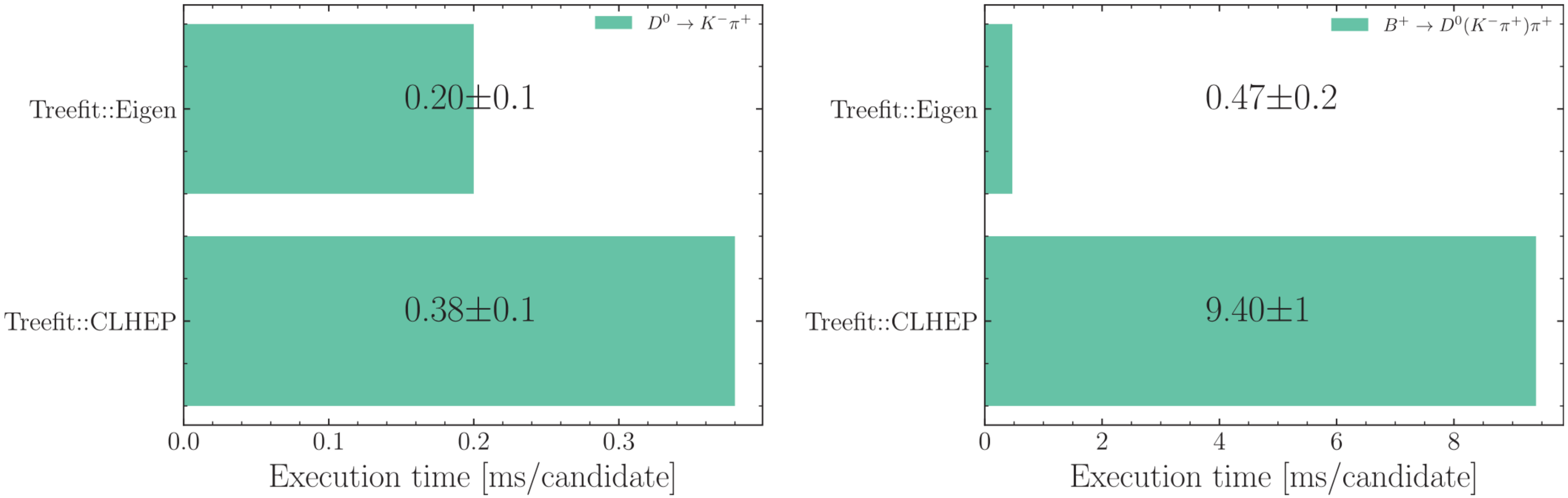
$\to$ 9 Degrees of Freedom

Beam: 4 plus Vertex: $2n - 3 = 5$

From Ralf Kliemt @ PANDA

Eigen

Eigen is a C++ template library for linear algebra: matrices, vectors, numerical solvers, and related algorithms.

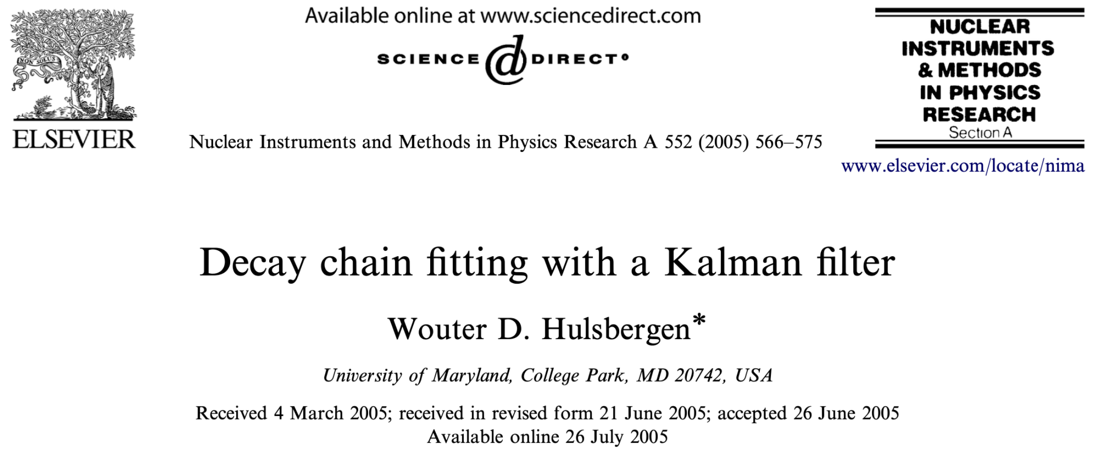


(a) (b)

Eigen doesn't have any dependencies other than the C++ standard library.

Existing Decay Tree Fitter

- **BaBar have a Tree Fitter, written by W. Hulsbergen.**
- **LHCb, PANDA and Belle II have similar Tree Fitter.**
- **Belle II code is shared (Github)**


Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

ELSEVIER

Nuclear Instruments and Methods in Physics Research A 552 (2005) 566–575

NUCLEAR INSTRUMENTS & METHODS IN PHYSICS RESEARCH Section A

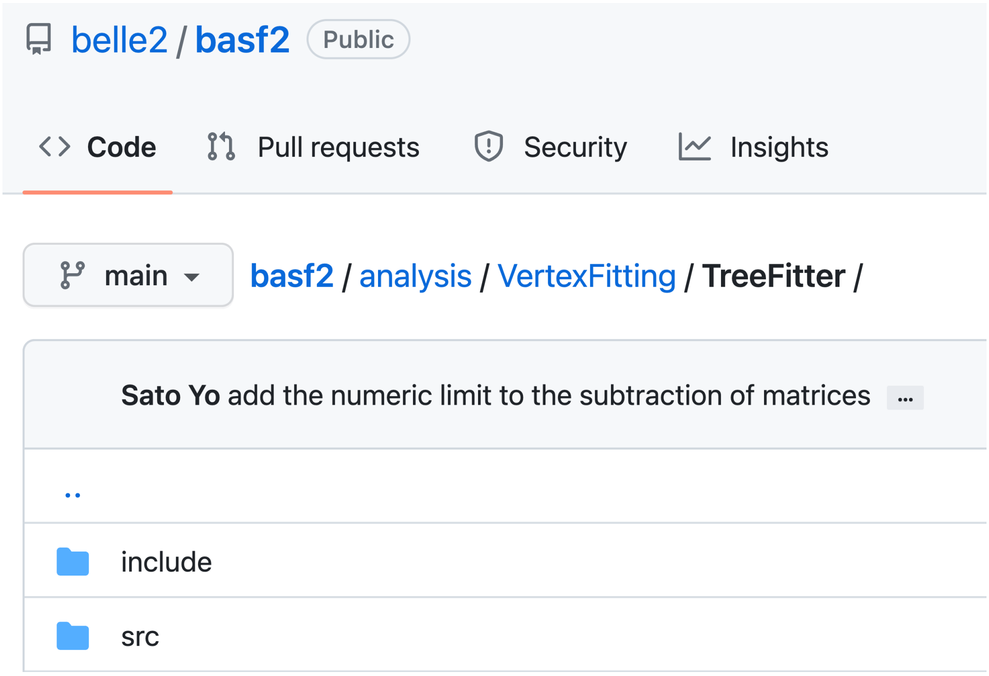
www.elsevier.com/locate/nima

Decay chain fitting with a Kalman filter

Wouter D. Hulsbergen*

University of Maryland, College Park, MD 20742, USA

Received 4 March 2005; received in revised form 21 June 2005; accepted 26 June 2005
Available online 26 July 2005



Summary

- **Kalman filter method has been implemented in vertex fitting and kinematic fitting.**
- **Leaf by Leaf fit method could deal with most of the vertex and kinematic problems.**
- **Global fit is useful in cascade decay, especially long-life particles.**
- **Eigen and Kalman filter could save the Global fit costs.**