timtrack

A new formalism for particle reconstruction with timing detectors

Juan A. Garzón
LabCAF/ Univ. Santiago de Compostela

timTrack

Main features:

- TimTrack is a Least Squares fitting Method using a general Matrix Formalism

  The formalism can be extended to many purposes, not only tracking

- TimTrack allows to put all the significative parameters in the fitting model.

  For example: in a track reconstruction problem coordinates, slopes, time and velocity may be let free in the fit (many advantages)

  TimTrack provides the variance-covariance matrix between all the parameters

- TimTracks work with the primary data, without any reduction.

  Calibration parameters, or even alignment parameters, could be let free in the fit. Then, it provides a common formalism useful for tracking, calibration and alignment (at one's discretion)

Basic equations

When we perform a measurement, any data, d, may be written as a function, m, (model) of a set of parameters, **s**, as:

$$d = m(\mathbf{s}) + \varepsilon$$

where ε (residual) is a random magnitude that we assume always to have a gaussian distribution.

If the measurement equipment provides several independent data at the same time, we can write that expression in a vector form:

$$\mathbf{d} = \mathbf{m}(\mathbf{s}) + \boldsymbol{\varepsilon}$$

This equation can be extended to include other magnitudes as the set of alignment and calibration constants, **α**.

$$\mathbf{d} = \mathbf{m}(\mathbf{s};\boldsymbol{\alpha}) + \boldsymbol{\varepsilon}$$

timtrack

Main goal:

All the significative Data (**d**)

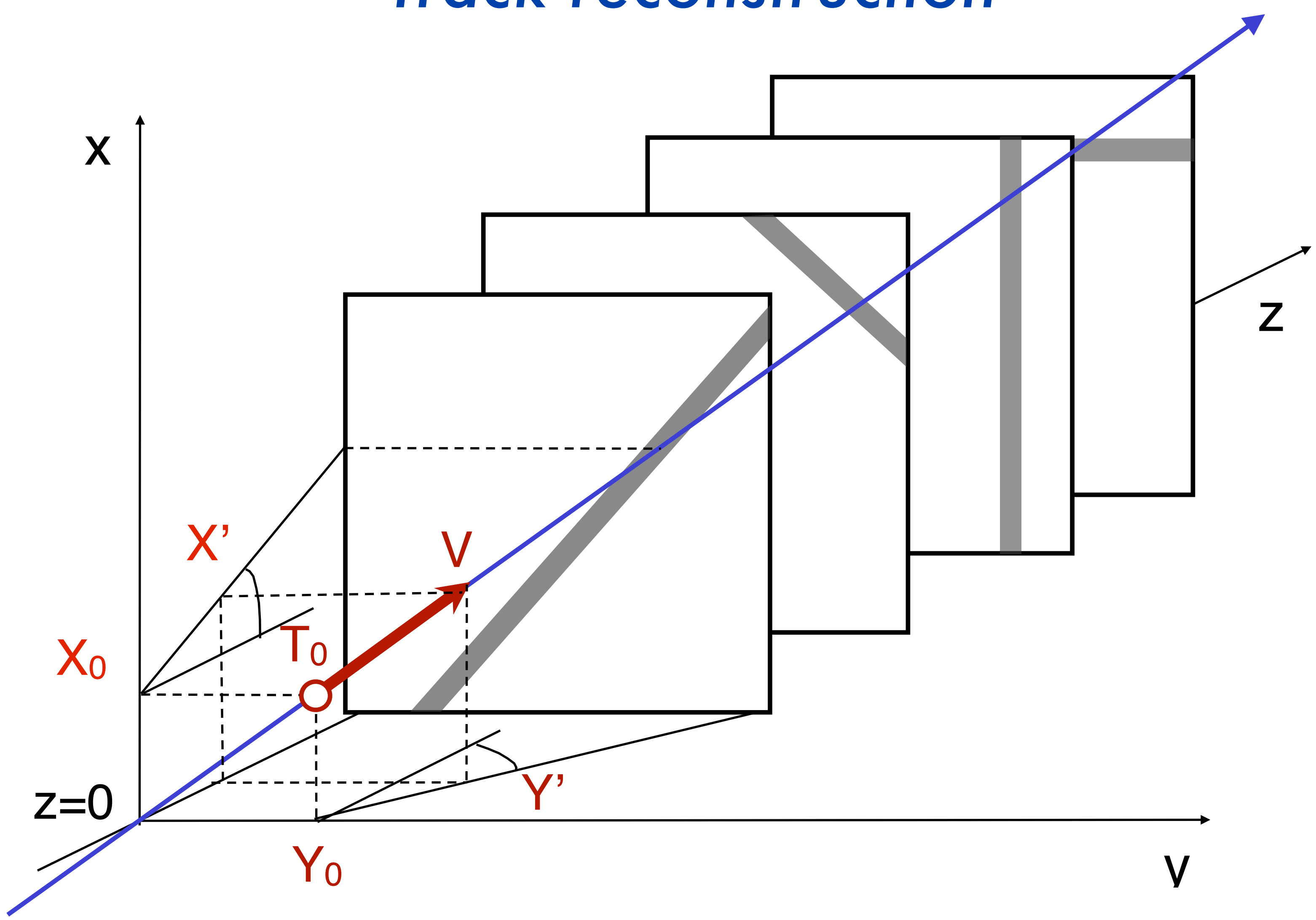
timtrack

Parameters (**s**)
+
Error matrix ($\mathcal{E}$)

Model: **m(d;s)**

s: All the unknown parameters

Track reconstruction



The particle is defined completely by a set of parameters: **s** = (X_0,X',Y_0,Y', T_0, V):

We call it **SAETA** (SmAllest sET of pArameters)

The Least Squares Method

Let suppose that we have:

- a set of n_m data: $\mathbf{d}$
- a set of n_s parameters: $\mathbf{s}$
- a model relating both sets: $\mathbf{d}=\mathbf{m}(\mathbf{s})$

The Least Squares Method (LSM) consists in finding the set of parameters $\mathbf{s}$ minimizing the function S:

$$S = \sum_i \left(\frac{d_i - m_i(\mathbf{s})}{\sigma_i} \right)^2$$

TimTrack: basic steps

Step 1: We write the function S in the form:

$$S = [\mathbf{d} - \mathbf{m}(\mathbf{s})]' \cdot W \cdot [\mathbf{d} - \mathbf{m}(\mathbf{s})]$$

where:

W is the inverse of the variance matrix of the data, or weigth matrix, that we assume to be diagonal.

(The apostrophe ' represents the transpose of a vector or matrix)

TimTrack: basic steps

Step 2. We decompose $\mathbf{m}$ in the form:

$$\mathbf{m}(\mathbf{s}) = G \cdot \mathbf{s} + \mathbf{g}(\mathbf{s})$$

where, G is the Jacobian matrix of $\mathbf{m}(\mathbf{s})$: $G = \frac{\partial \mathbf{m}(\mathbf{s})}{\partial \mathbf{s}}$.

When $\mathbf{m}(\mathbf{s})$ is linear:

$$\mathbf{g}(\mathbf{s}) = \mathbf{m}(\mathbf{s}) - G \cdot \mathbf{s} = \mathbf{g}_0$$

TimTrack: basic steps

Step 3. We introduce the new magnitudes:

$$K = G' \cdot W \cdot G$$
$$\mathbf{a} = G' \cdot W \cdot (\mathbf{d} - \mathbf{g}_0)$$
$$s_0 = (\mathbf{d} - \mathbf{g}_0)' \cdot W \cdot (\mathbf{d} - \mathbf{g}_0)$$

Using these magnitudes, S can be written in the form:

$$S = \mathbf{s}' \cdot K \cdot \mathbf{s} - 2\mathbf{s}' \cdot \mathbf{a} + s_0$$

TimTrack: basic steps

Step 4. The set of parameters $\mathbf{s}_m$, minimizing S, verifies:

$$\frac{\partial S}{\partial \mathbf{s}} = K \cdot \mathbf{s}_m - \mathbf{a} = \mathbf{0}$$

that is equivalent to:

$$K \cdot \mathbf{s}_m = \mathbf{a} \qquad \textit{(Normal equations)}$$

or:

$$\boxed{\mathbf{s}_m = K^{-1} \cdot \mathbf{a}}$$

Observe that:

- $\mathbf{a}$ has dimension $n_s \times 1$: *(Vector of reduced data)*
- K has dimension $n_s \times n_s$: *(Configuration matrix)*

TimTrack: basic steps

Step 5. The variance-covariance (error) matrix of the parameters in $\mathbf{s}_m$ is:

$$\boxed{\mathcal{E} = \left(\frac{1}{2}\frac{\partial^2 S}{\partial \mathbf{s}^2}\right)^{-1} = K^{-1}}$$

(Observe that the solution of the fit verifies: $\mathbf{s}_m = \mathcal{E} \cdot \mathbf{a}$)

TimTrack: summary of steps

1. $S = [\mathbf{d} - \mathbf{m}(\mathbf{s})]' \cdot W \cdot [\mathbf{d} - \mathbf{m}(\mathbf{s})]$

2 $\mathbf{m}(\mathbf{s}) = G \cdot \mathbf{s} + \mathbf{g}(\mathbf{s})$ $\quad \mathbf{g}(\mathbf{s}) = \mathbf{m}(\mathbf{s}) - G \cdot \mathbf{s} = \mathbf{g}_0$

3.
$$K = G' \cdot W \cdot G$$
$$\mathbf{a} = G' \cdot W \cdot (\mathbf{d} - \mathbf{g}_0)$$
$$s_0 = (\mathbf{d} - \mathbf{g}_0)' \cdot W \cdot (\mathbf{d} - \mathbf{g}_0)$$
$$\rightarrow \quad S = \mathbf{s}' \cdot K \cdot \mathbf{s} - 2\mathbf{s}' \cdot \mathbf{a} + s_0$$

4.
$$\frac{\partial S}{\partial \mathbf{s}} = K \cdot \mathbf{s}_m - \mathbf{a} = \mathbf{0} \quad \rightarrow \quad K \cdot \mathbf{s}_m = \mathbf{a}$$

$$\boxed{\mathbf{s}_m = K^{-1} \cdot \mathbf{a}}$$

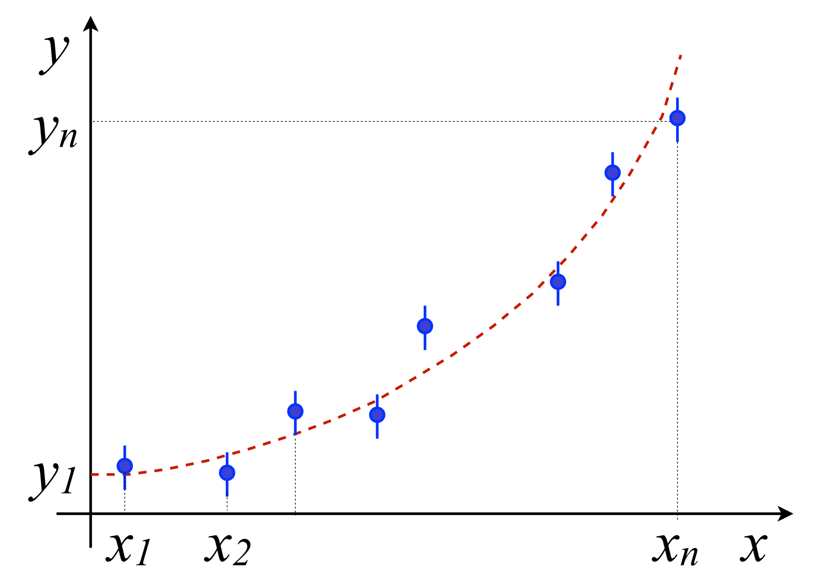
5.
$$\mathcal{E} = \left(\frac{1}{2} \frac{\partial^2 S}{\partial \mathbf{s}^2} \right)^{-1} = K^{-1}$$

Linear model

Non-tracking typical example: Polynomial fit

Let suppose that we have a set of data:

$$\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$$



and we want to fit them a function (model) of the form:

$$y_i = a + bx_i + cx_i^2$$

According Timtrack:

$$\mathbf{s} = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \qquad G = \begin{pmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ & \vdots & \\ 1 & x_n & x_n^2 \end{pmatrix}$$

and:

$$\mathbf{g}_0 = y - G \cdot \mathbf{s} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Linear model

Typical example: Polinomial fit

If, for simplicity, we assume $W = I_n$

we get:

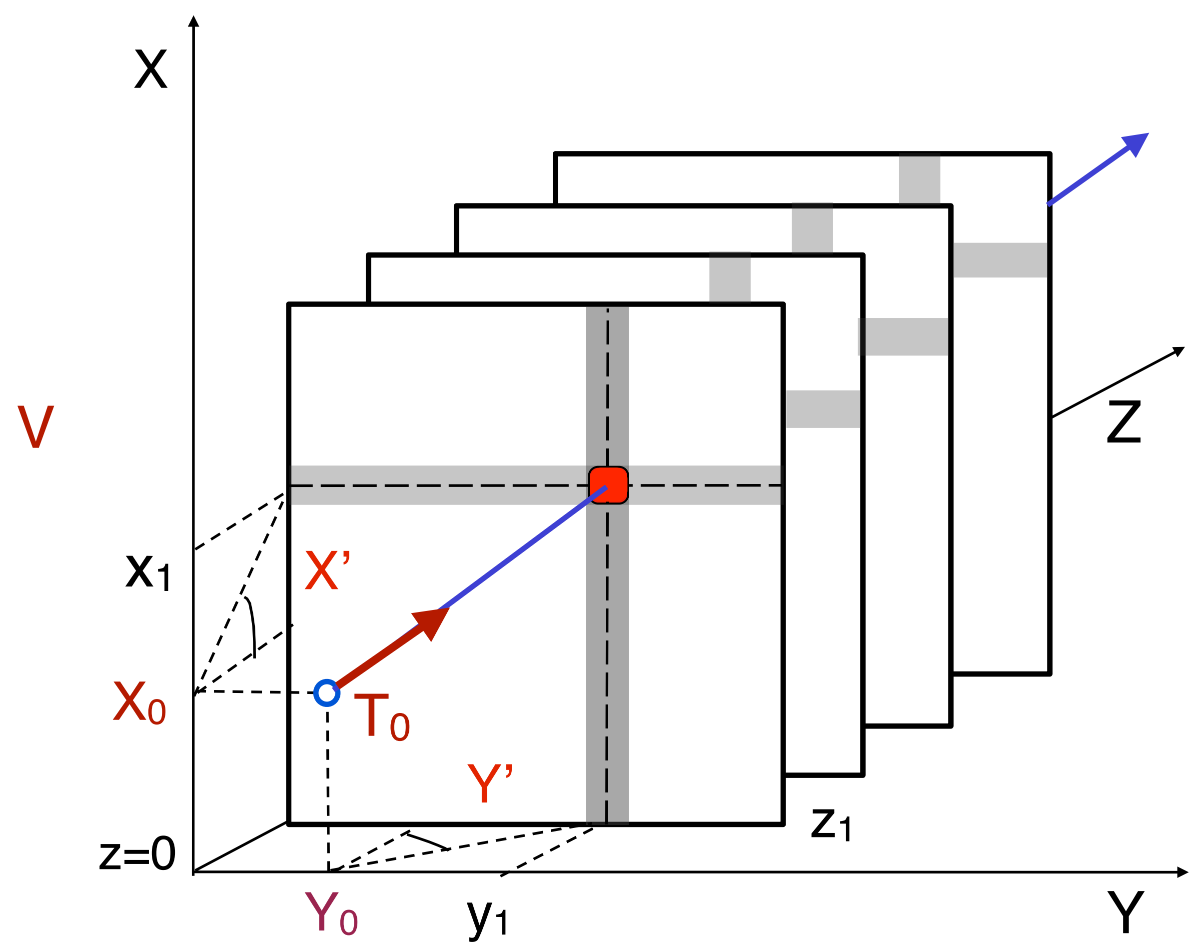
$$K = G' \cdot G = \begin{pmatrix} n & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{pmatrix} \qquad \mathbf{d} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \qquad \mathbf{a} = G' \cdot \mathbf{d} = \begin{pmatrix} \sum y_i \\ \sum y_i x_i \\ \sum y_i x_i^2 \end{pmatrix}$$

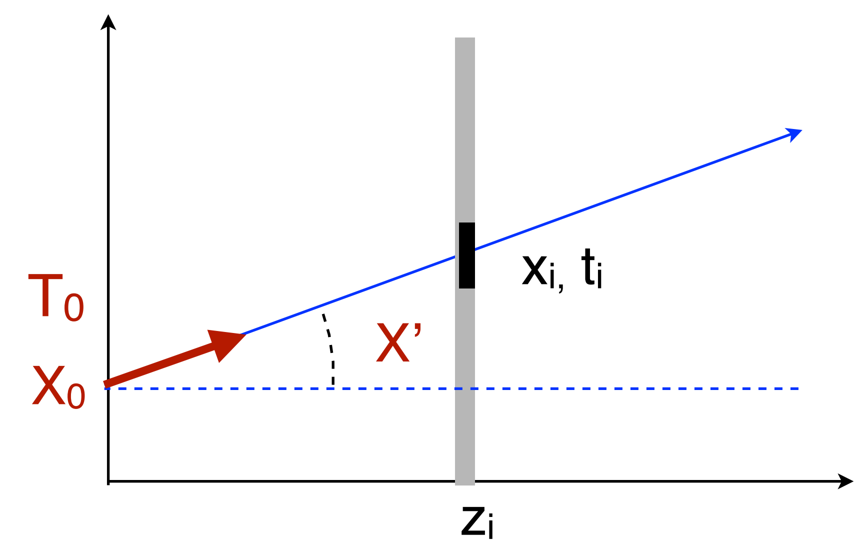
and the final solution is given by the formula:

$$\mathbf{s}_m = K^{-1} \cdot \mathbf{a} \longrightarrow \boxed{\begin{pmatrix} a \\ b \\ c \end{pmatrix}_{min} = \begin{pmatrix} n & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{pmatrix}^{-1} \cdot \begin{pmatrix} \sum y_i \\ \sum y_i x_i \\ \sum y_i x_i^2 \end{pmatrix}}$$

Linear model

EXAMPLE: Array of N pad detectors





$$x_i = X_0 + X' z_i$$
$$y_i = Y_0 + Y' z_i$$

Linear model

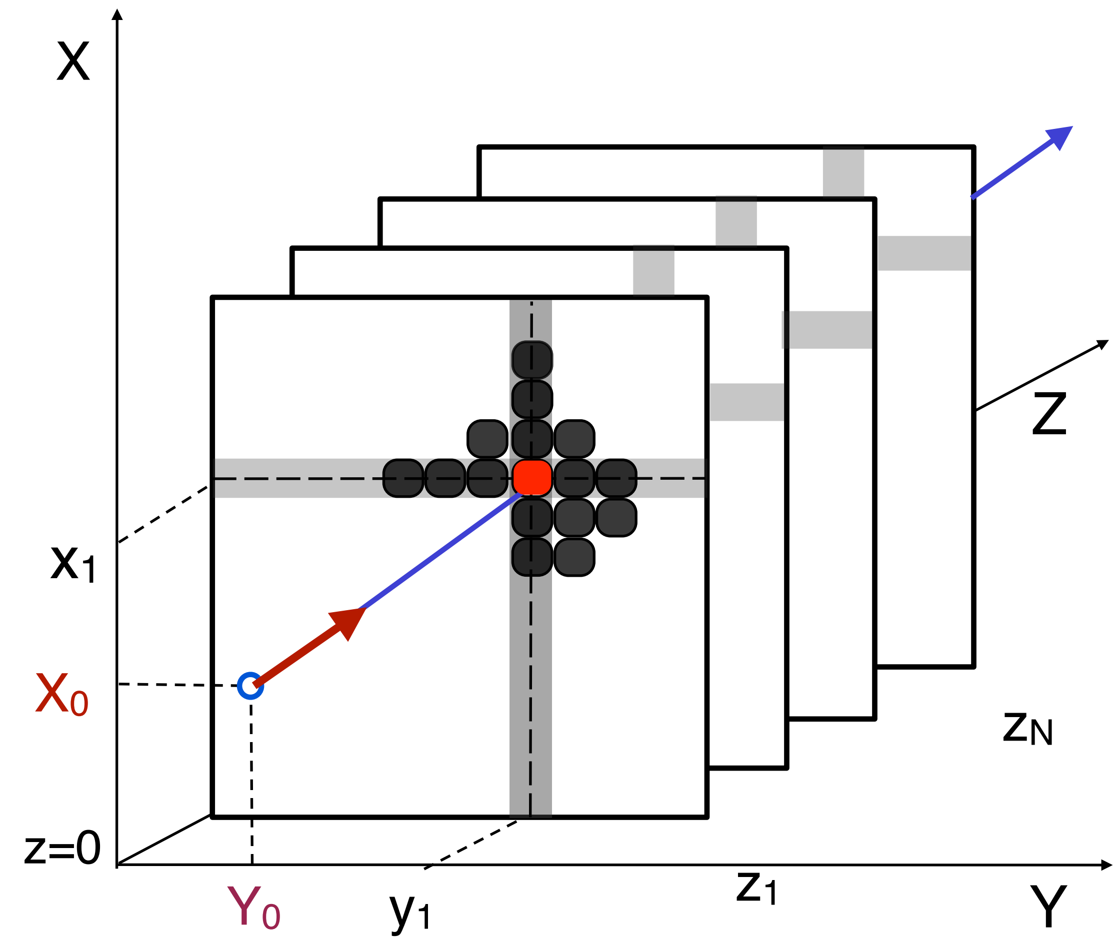
EXAMPLE: Array of N pad detectors

Saeta:

$$\mathbf{s} = (X_0, X', Y_0, Y')$$

Fitting model:

$$\mathbf{m}(\mathbf{s}) = \begin{pmatrix} X_0 + X' \cdot z_1 \\ Y_0 + Y' \cdot z_1 \\ \\ \vdots \\ X_0 + X' \cdot z_N \\ Y_0 + Y' \cdot z_N \\ \end{pmatrix}$$



Linear model

EXAMPLE: Array of N pad detectors

Saeta:

$$\mathbf{s} = (X_0, X', Y_0, Y')$$

Fitting model:

$$\mathbf{m}(\mathbf{s}) = \begin{pmatrix} X_0 + X' \cdot z_1 \\ Y_0 + Y' \cdot z_1 \\ \vdots \\ X_0 + X' \cdot z_N \\ Y_0 + Y' \cdot z_N \end{pmatrix}$$

$$\mathbf{m}(\mathbf{s}) = G \cdot \mathbf{s} + \mathbf{g}(\mathbf{s})$$

Main matrices and vectors:

$$G = \begin{pmatrix} 1 & z_1 & 0 & 0 \\ 0 & 0 & 1 & z_1 \\ & & \vdots & \\ 1 & z_N & 0 & 0 \\ 0 & 0 & 1 & z_N \end{pmatrix} \qquad \mathbf{g}_0 = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

timTrack

Linear model

EXAMPLE: Array of N pad detectors

Finding the solution from the data

$$\mathbf{d} = \begin{pmatrix} x_1^d \\ y_1^d \\ \vdots \\ x_N^d \\ y_N^d \end{pmatrix}, \quad \mathbf{d} - \mathbf{g}_0 = \mathbf{d} \rightarrow \begin{array}{l} \mathbf{a} = G' \cdot W \cdot (\mathbf{d} - \mathbf{g}_0) \\ K = G' \cdot W \cdot G \end{array} \rightarrow K = \begin{pmatrix} k_{xx} & k_{xx'} & 0 & 0 \\ k_{xx'} & k_{x'x'} & 0 & 0 \\ 0 & 0 & k_{yy} & k_{yy'} \\ 0 & 0 & k_{yy'} & k_{y'y'} \end{pmatrix}$$

and then:

$$\boxed{\mathbf{s} = K^{-1}\mathbf{a}}$$

Linear model

EXAMPLE: Array of N pad detectors

K matrix elements:

Element	Coordinate terms	Time terms
k_{xx}	$\sum \frac{1}{\sigma^2_{x_i}}$	
$k_{xx'}$	$\sum \frac{z_i}{\sigma^2_{x_i}}$	
$k_{x'x'}$	$\sum \frac{z_i^2}{\sigma^2_{x_i}}$	
k_{yy}	$\sum \frac{1}{\sigma^2_{y_i}}$	
$k_{yy'}$	$\sum \frac{z_i}{\sigma^2_{y_i}}$	
$k_{y'y'}$	$\sum \frac{z_i^2}{\sigma^2_{y_i}}$	

a vector elements:

a_x	$\sum \frac{X_i^m}{\sigma^2_{x_i}}$	
$a_{x'}$	$\sum \frac{X_i^m \cdot z_i}{\sigma^2_{x_i}}$	
a_y	$\sum \frac{Y_i^m}{\sigma^2_{y_i}}$	
$a_{y'}$	$\sum \frac{Y_i^m \cdot z_i}{\sigma^2_{y_i}}$	

$$\boxed{\mathbf{s} = K^{-1}\mathbf{a}}$$

Linear model

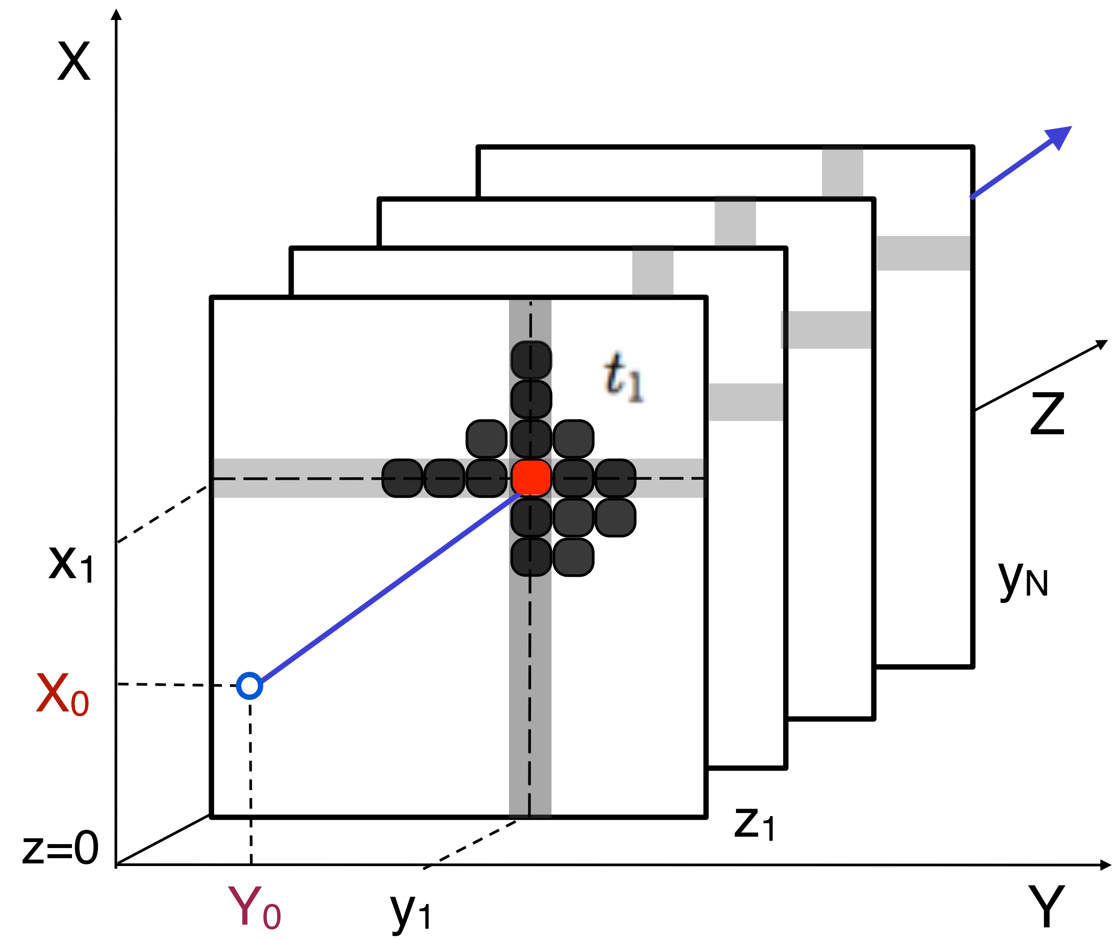
EXAMPLE: Array of N pad detectors

Comments:

- The error matrix of the saeta becomes:

$$\mathcal{E} = K^{-1} = \begin{pmatrix} e_{xx} & e_{xx'} & 0 & 0 \\ e_{xx'} & e_{x'x'} & 0 & 0 \\ 0 & 0 & e_{yy} & e_{yy'} \\ 0 & 0 & e_{yy'} & e_{y'y'} \end{pmatrix}$$

- The pairs (X_0,X'), (Y_0,Y') and (T_0,$1/V_z$) are correlated.



- Using the linear approach TimTrack provides the same result as classical methods fitting separately each pair, although the matrix formalism makes things easier and faster.

Linear model

EXAMPLE: Array of N strip detectors

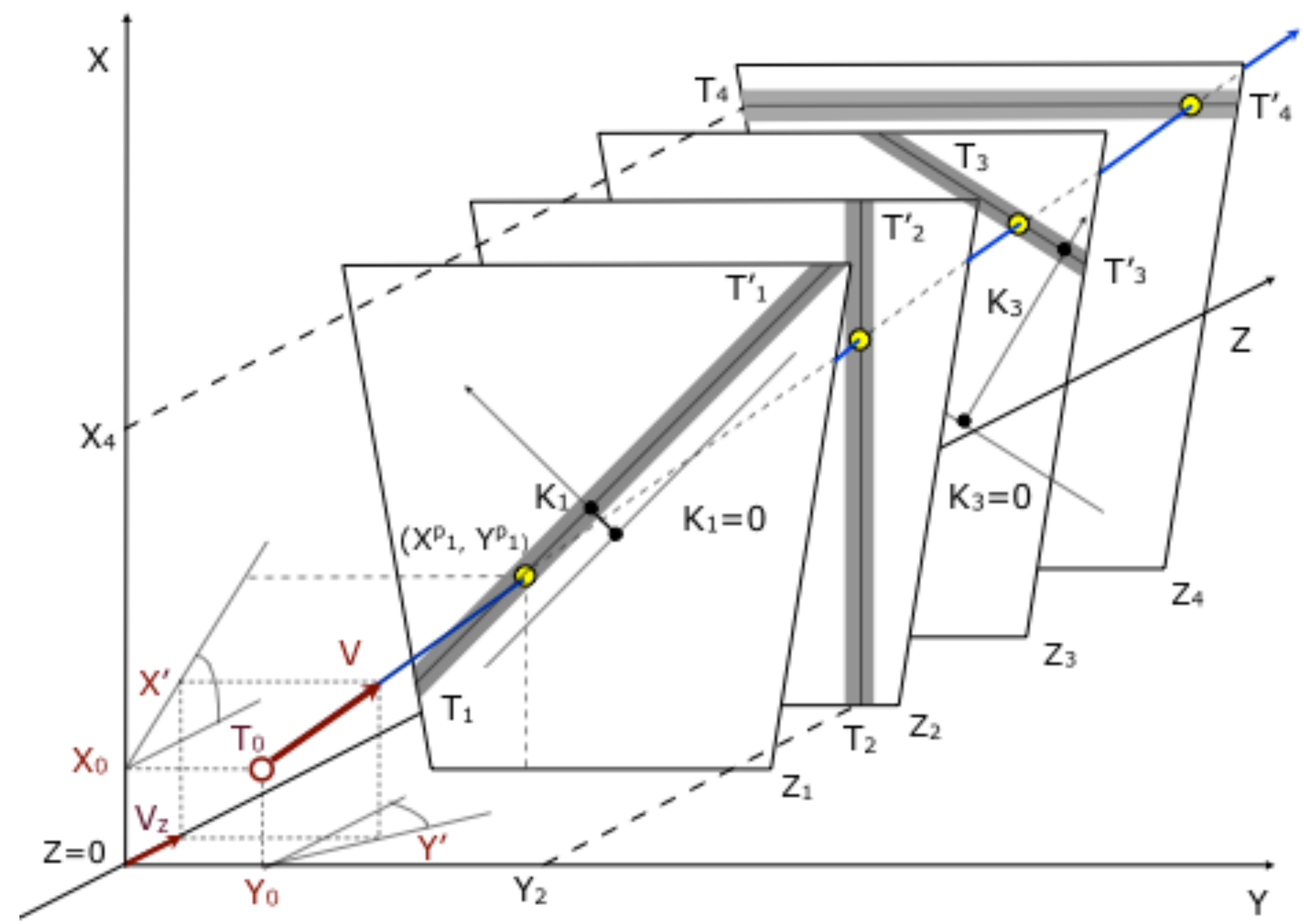
Saeta: $\mathbf{s} = (X_0, X', Y_0, Y')$

Cross-point at i-plane:

$$\begin{cases} X_i^p = X_0 + X' z_i \\ Y_i^p = Y_0 + Y' z_i \end{cases}$$

Model:

$$\begin{cases} K_i^e = X_i^p \cos\varphi_i + Y_i^p \sin\varphi_i \end{cases}$$



EXAMPLE: Array of N strip detectors

Non-linear model

When the model $\mathbf{m}(\mathbf{s})$ to be fitted is *non-linear*, the S function should be minimized using an iterative process:

- We look for an initial value of the set of parameters $\mathbf{s}_0$, that is near to the minimum of S.

- Then, we follow the same steps used with the linear model with the following differences:

Step 2.
We calculate $G(\mathbf{s})$ and then, we build: $\boxed{\mathbf{g}(\mathbf{s}) = \mathbf{m}(\mathbf{s}) - G(\mathbf{s}) \cdot \mathbf{s}}$

Step 3.
We calculate the matrix K and the vector $\mathbf{a}$ at $\mathbf{s} = \mathbf{s}_0$:

$$\boxed{\begin{aligned} K_0 &= G'(\mathbf{s}_0) \cdot W \cdot G(\mathbf{s}_0) \\ \mathbf{a}_0 &= G'(\mathbf{s}_0) \cdot W \cdot (\mathbf{d} - \mathbf{g}(\mathbf{s}_0)) \end{aligned}}$$

Non-linear model

Step 4.
We estimate a new set of parameters, $\mathbf{s}_1$, using the equation:

$$\mathbf{s}_1 = K_0^{-1} \cdot \mathbf{a}_0$$

We repeat this step iteratively until the convergence is reached:

$$\boxed{\mathbf{s}_i = K_{i-1}^{-1} \cdot \mathbf{a}_{i-1}}$$

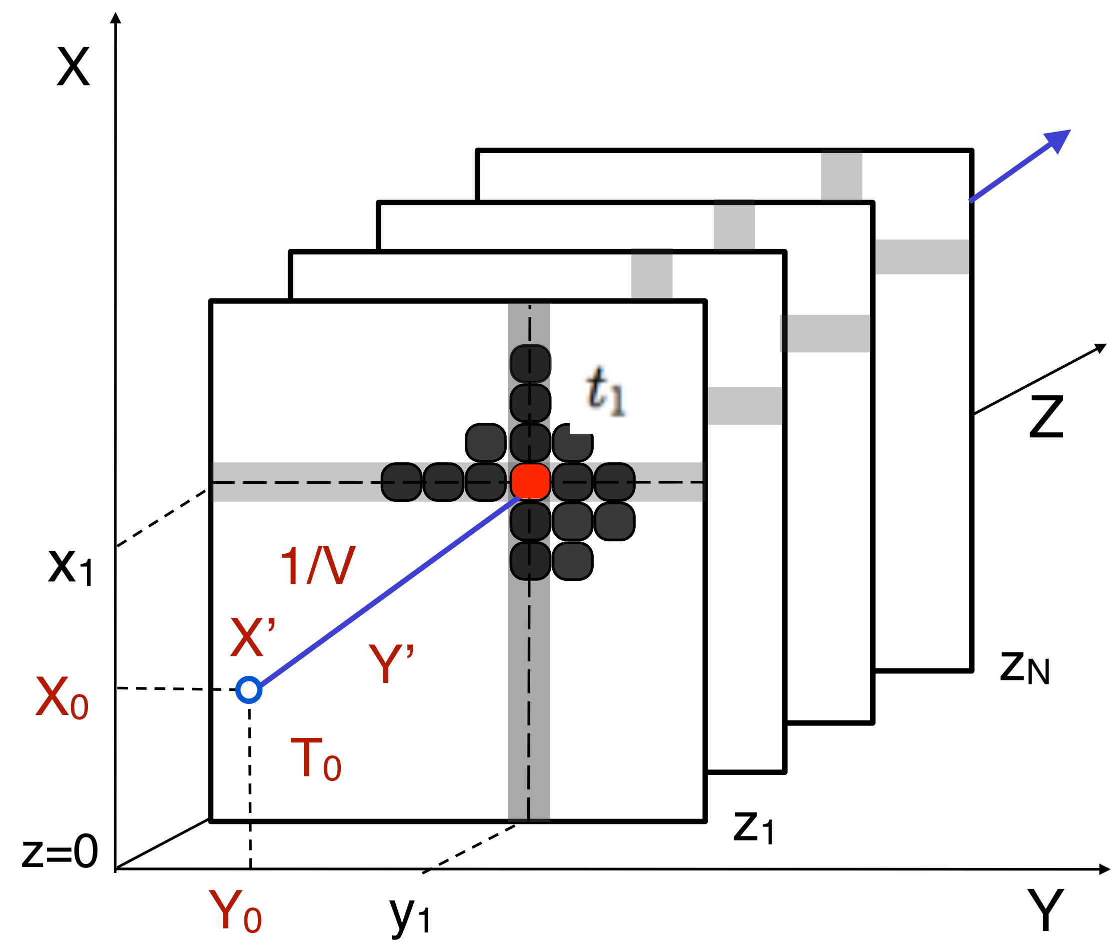
Non-linear model

EXAMPLE: Array of N pad detectors

Saeta: $\mathbf{s} = (X_0, X', Y_0, Y', T_0, 1/V)$

Model:

$$\mathbf{m(s)} = \begin{pmatrix} x_1 \\ y_1 \\ t_1 \\ \vdots \\ x_N \\ y_N \\ t_N \end{pmatrix} = \begin{pmatrix} X_0 + X' \cdot z_1 \\ Y_0 + Y' \cdot z_1 \\ T_0 + 1/V \cdot \sqrt{1 + X'^2 + Y'^2} \cdot z_1 \\ \vdots \\ X_0 + X' \cdot z_N \\ Y_0 + Y' \cdot z_N \\ T_0 + 1/V \cdot \sqrt{1 + X'^2 + Y'^2} \cdot z_N \end{pmatrix}$$



that is non-linear, both in X′ and Y′

Non-linear model

EXAMPLE: Array of N pad detectors

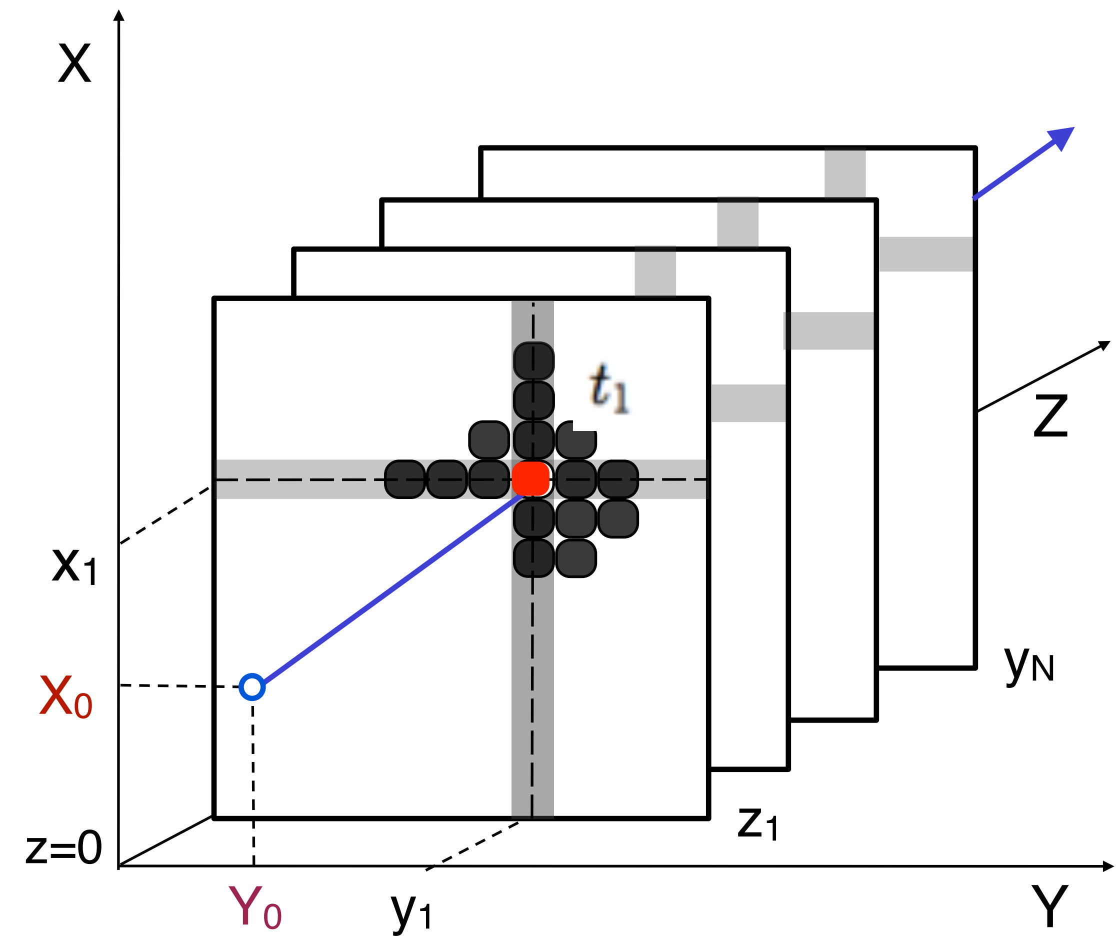
Saeta: $\mathbf{s} = (X_0, X', Y_0, Y', T_0, 1/V)$

When X' and Y' are small, we may use the approximation:

$$\sqrt{1+X'^2+Y'^2} = 1 + \frac{X'^2+Y'^2}{2}$$

and we arrive for each plane to:

$$G_i(\mathbf{s}) = \begin{pmatrix} 1 & z_i & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & z_i & 0 & 0 \\ 0 & \frac{X'}{V} z_i & 0 & \frac{Y'}{V} z_i & 1 & \left(1+\frac{X'^2+Y'^2}{2}\right) z_i \end{pmatrix}$$



$$\mathbf{g}_i(\mathbf{s}) = \begin{pmatrix} 0 \\ 0 \\ -\frac{X'^2+Y'^2}{V} z_1 \end{pmatrix} \longrightarrow \begin{array}{l} K_0 = G'(\mathbf{s}_0) \cdot W \cdot G(\mathbf{s}_0) \\ \mathbf{a}_0 = G'(\mathbf{s}_0) \cdot W \cdot (\mathbf{d} - \mathbf{g}(\mathbf{s}_0)) \end{array}$$

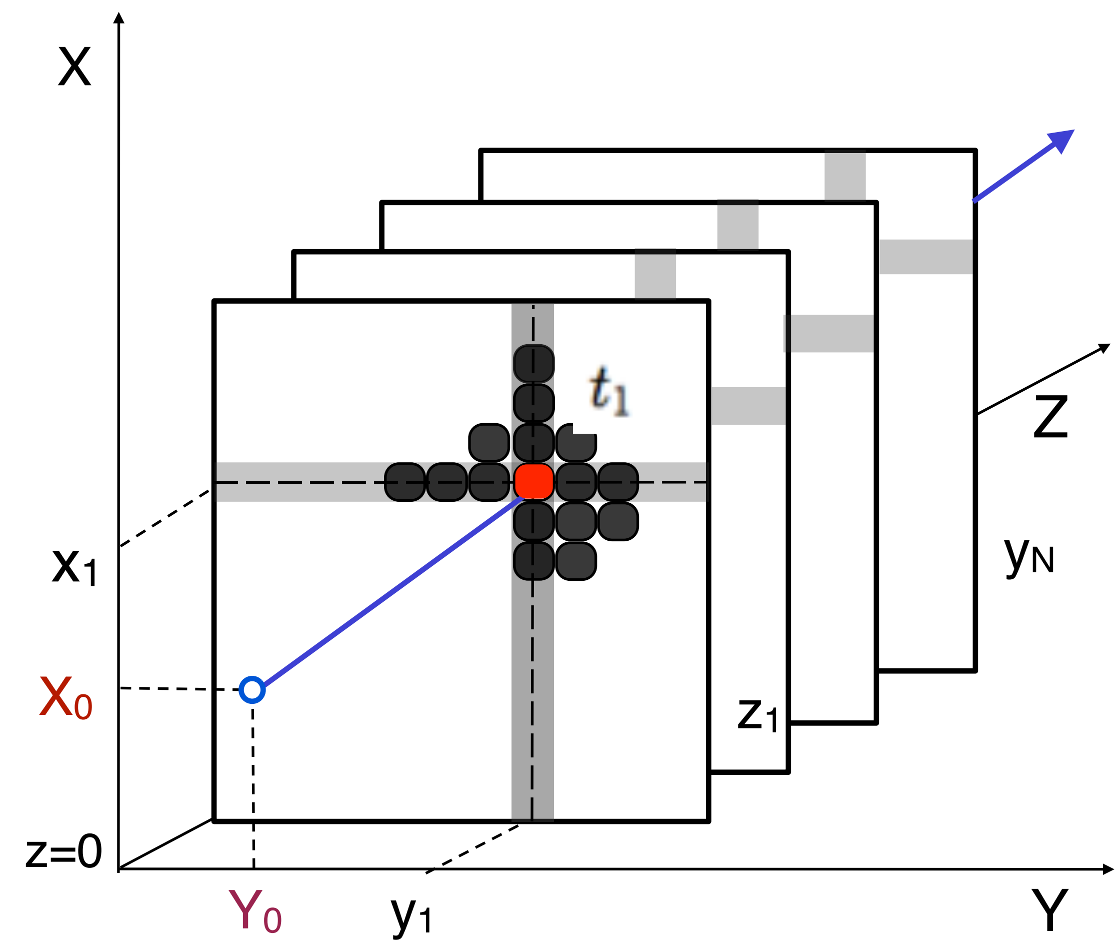
Non-linear model

EXAMPLE: Array of N pad detectors

Saeta: $\mathbf{s} = (X_0, X', Y_0, Y', T_0, 1/V)$

The matrix K takes the form:

$$K = \begin{pmatrix} k_{xx} & k_{xx'} & 0 & 0 & 0 & 0 \\ k_{xx'} & k_{xx'} & 0 & k_{xy'} & k_{xt} & k_{xv} \\ 0 & 0 & k_{yy} & k_{yy'} & 0 & 0 \\ 0 & k_{y'x'} & k_{y'y} & k_{y'y'} & k_{y't} & k_{y'v} \\ 0 & k_{tx'} & 0 & k_{ty'} & k_{tt} & k_{tv} \\ 0 & k_{vx'} & 0 & k_{vy'} & k_{vt} & k_{vv} \end{pmatrix}$$



Non-linear model with constraints in the parameters

Let consider the case where the set parameters $\mathbf{s}$ do fulfill:

$$\boxed{\mathbf{f}(\mathbf{s}) = 0 \qquad (n_c \text{ equations})}$$

- Now, we minimize iteratively the Lagrange function defined as:

$$\boxed{L(\mathbf{s}) = (\mathbf{d} - \mathbf{m}(\mathbf{s}))' \cdot W \cdot (\mathbf{d} - \mathbf{m}(\mathbf{s})) + 2 \cdot \lambda' \cdot \mathbf{f}(\mathbf{s})}$$

where λ is a vector of n_c elements: the Lagrange multipliers.

- We introduce the jacobian matrix of the constraint functions:

$$\boxed{R(\mathbf{s}) = \partial_{\mathbf{s}} \mathbf{f}(\mathbf{s})}$$

Non-linear model with constraints in the parameters

- Starting from a point, close enough to the minimum:

$$\mathbf{s} = \mathbf{s}_0, \quad R_0 = R(\mathbf{s}_0) \quad \mathbf{f}_0 = \mathbf{f}(\mathbf{s}_0)$$

The matrix equation:

$$\boxed{\begin{pmatrix} \delta\mathbf{s}_1 \\ \lambda_1 \end{pmatrix} = \begin{pmatrix} K_0 & R_0' \\ R_0 & 0 \end{pmatrix}^{-1} \cdot \begin{pmatrix} \mathbf{a}_0(\mathbf{s}_0) - K_0 \cdot \mathbf{s}_0 \\ -\mathbf{f}_0 \end{pmatrix}}$$

provides the next-step $\mathbf{s}$ and λ vectors:

$$\boxed{\begin{array}{l} \mathbf{s}_1 = \mathbf{s}_0 + \delta\mathbf{s}_1 \\ \lambda_1 \end{array}}$$

- The iteration is repeated until convergence and $\lambda_i = 0$.

Non-linear model with constraints in the parameters

Uncertainty analysis:

- We build:

$$\boxed{\begin{pmatrix} K & R' \\ R & 0 \end{pmatrix}^{-1} = \begin{pmatrix} H & Q' \\ Q & Z \end{pmatrix}}$$

-At the minimum, the *Error Matrix* is:

$$\boxed{\mathcal{E} = K^{-1}\,(I - R'Q)}$$

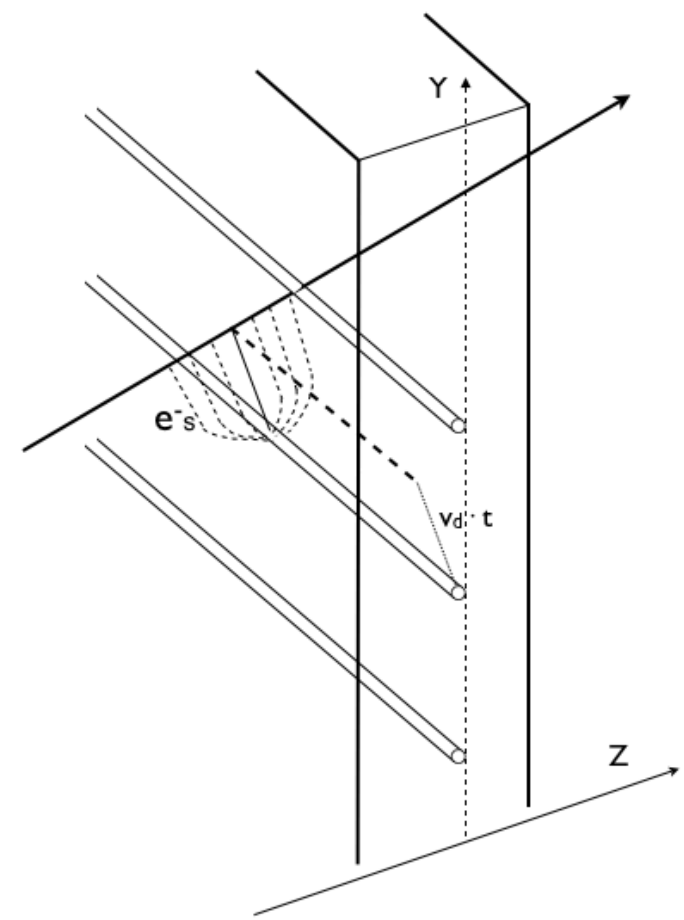
K has dimensions: $n_s \times n_s$
R' has dimensions: $n_s \times n_c$
Q' has dimensions: $n_c \times n_s$

-The number of degrees of freedom of the fit is: $\boxed{n_f = n_m - n_s + n_c}$

Examples

timTrack

Example: Drift chambers



In a drift chamber, each wire gives 3 data:

1. wire coordinate
2. drift time
3. dE/dx

TimTrack allows to treat them separately and to use the 3 pieces of information to estimate the particle movement parameters

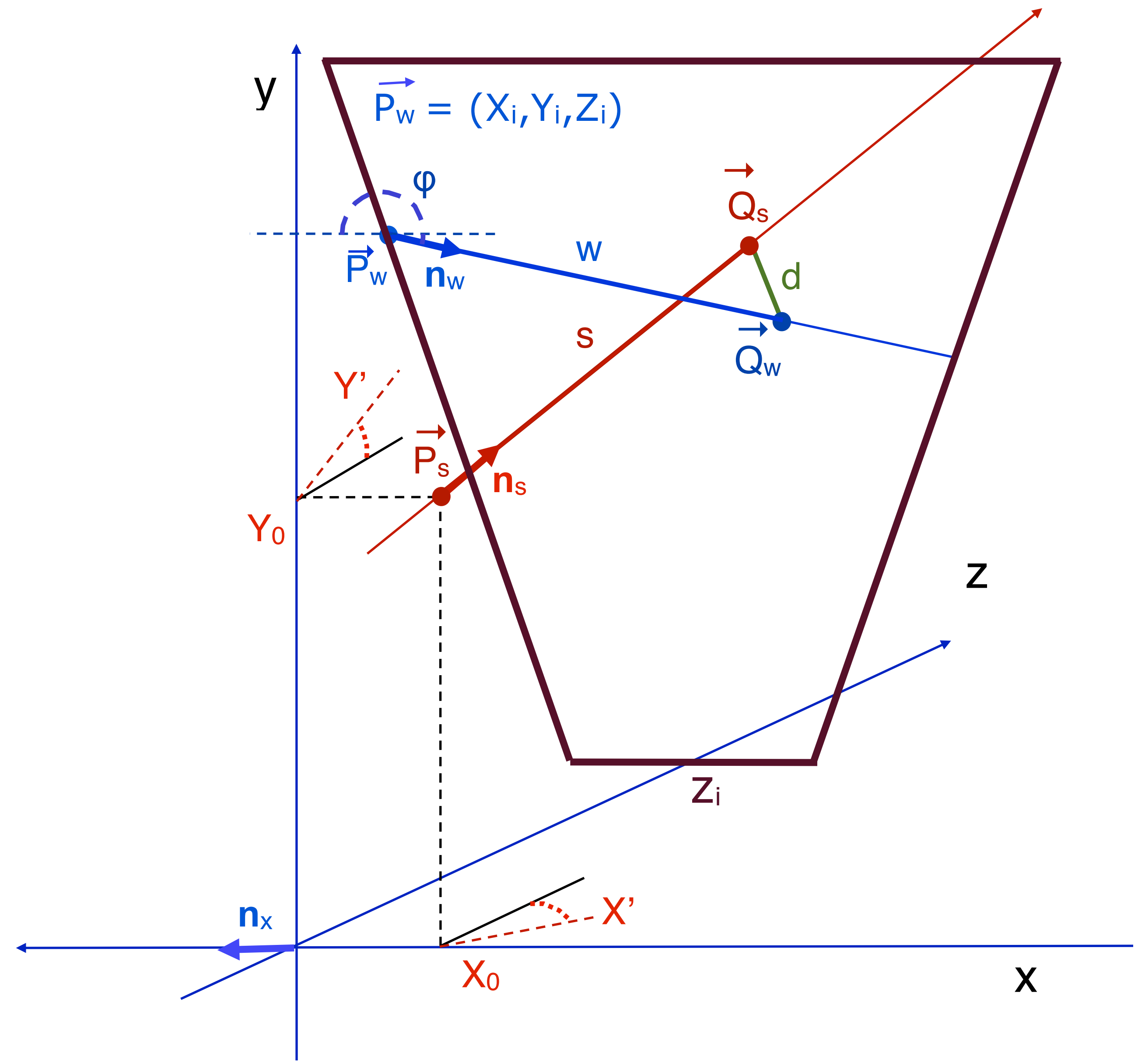
Example: Drift chambers

$$d = (P_s - P_w) \cdot \mathbf{n}_{sw}$$

$$\mathbf{n}_{sw} = \frac{\mathbf{n}_s \times \mathbf{n}_w}{|\mathbf{n}_s \times \mathbf{n}_w|}$$

$$s = \frac{(\mathbf{n}_s \times \mathbf{n}_w) \cdot (\mathbf{n}_w \times (\mathbf{P}_s - \mathbf{P}_w))}{|\mathbf{n}_s \times \mathbf{n}_w|^2}$$

$$w = \frac{(\mathbf{n}_s \times \mathbf{n}_w) \cdot (\mathbf{n}_s \times (\mathbf{P}_s - \mathbf{P}_w))}{|\mathbf{n}_s \times \mathbf{n}_w|^2}$$

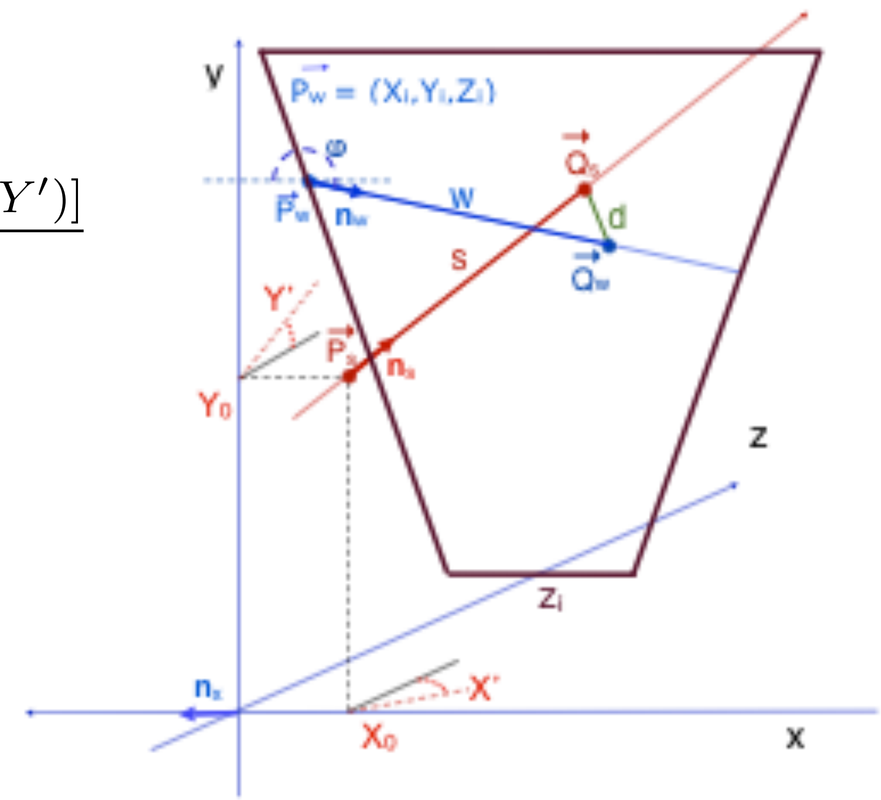


Example: Drift chambers

$$s = \frac{z_i \cdot \sqrt{X'^2 + Y'^2 + 1} \cdot [1 - (-X' \sin\varphi + Y' \cos\varphi)(-X'_i \sin\varphi + Y'_i \cos\varphi)]}{1 + (-X' \sin\varphi + Y' \cos\varphi)^2}$$

$$d = \frac{z_i \cdot [-(X' + X'_i) \sin\varphi + (Y' + Y'_i) \cos\varphi]}{\sqrt{1 + (-X' \sin\varphi + Y' \cos\varphi)^2}}$$

$$w = \frac{z_i \cdot [-(X' + X'_i) \cos\varphi + (Y' + Y'_i) \sin\varphi - (-X' \sin\varphi + Y' \cos\varphi)(X' Y'_i - X'_i Y')]}{1 + (-X' \sin\varphi + Y' \cos\varphi)^2}$$



where:

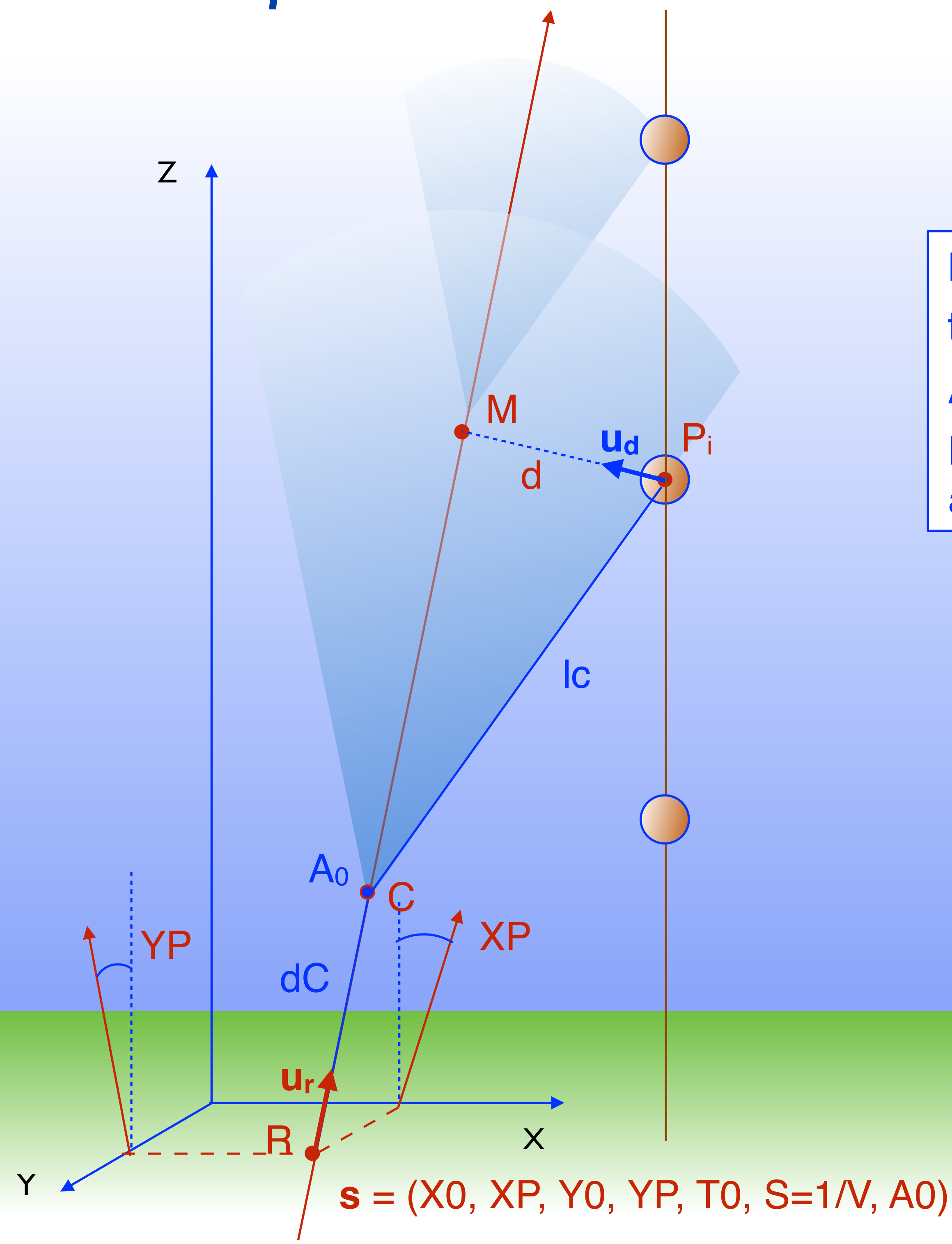
$$X'_i = \frac{X_0 - X_i}{z_i} \qquad Y'_i = \frac{Y_0 - Y_i}{z_i}$$

Finally, we fit:

$$t = T_0 + \frac{s}{V} + \frac{d}{v_d} + \frac{w}{v_w} = T_0 + s \cdot S + d \cdot s_d + w \cdot s_w$$

timTrack

Example: Antares observatory



Models for i-DOM:

t_i = T0 + dC S + lc n/c

$A_i = A_0\ e^{-lc/\lambda}$

K_i : config. matrix

a_i : reduced data vector

Solution:

$K = \Sigma K_i$

$\mathbf{a} = \Sigma\, \mathbf{a}_i$

$\mathcal{E} = K^{-1}$

$s = \mathcal{E}\, \mathbf{a}$

Example: Antares observatory. Code

```
(* mK_Antares
 matriz K para las señales de tiempo y amplitud
 de la luz Cherenkov producida por un muon en un DOM
 Sistema de coordenadas: ejez x vertical hacia arriba,
plano de referencia en el fondo del mar
  pR (X0, Y0, 0): punto de entrada del muon en el agua;
(XP, YP, 1): direccion del muon;
pP: posicion del DOM que ha hecho la medida;
pM: punto de maximo acercamiento del punto M a la direccion del muon;
pC: punto de emision de la luz Cherenkov medido en el DOM en el punto P
         *)

(* Constantes *)

c = 30 ; (* cm/ns *)
n = 1.3; (* indice de refraccion del agua *)
cn = c / n ; (* velocidad de la luz en el agua *)
sn = 1 / cn;
alen = 6000; (* m. Atenuation length *)


(* Saeta tipica, ademas con intensidad de emision A *)
vsa = {X0, XP, Y0, YP, T0, S, A};
(* Posicion del DOM i *)
pP = {Pxi, Pyi, Pzi};
(* Modelos: *)
(* Medida del tiempo en el DOM en el punto P:
  El tiempo de llegada de la luz Cherenkov al DOM referida al
   tiempo T0 es el tiempo de vuelo de la particula hasta el punto
   C y el tiempo de recorrido de la luz desde el punto C al punto P
*)
pR = {X0 , Y0, 0};
(* Vector director de la recta r *)
vr = {XP, YP, 1};
(* Vector director normalizado *)
nvr = Sqrt[XP^2 + YP^2 + 1];
ur = vr / nvr;
vPR = pP - pR;
dM = vPR.ur;
pM = pR + dM ur;
vd = Cross[ur, vPR];
nvd = Sqrt[vd[[1]] ^ 2 + vd[[2]] ^ 2 + vd[[3]] ^ 2];
ud = vd / nvd;
beta = 1 / (S c);
cost = 1 / (beta n);
sint = Sqrt[1 - cost^2];
tant = sint / cost;
(* Longitud de recorrido de la luz Cherenkov *)
lc = nvd / sint;
(* Punto de emision de la luz Cherenkov*)
dC = nvd / tant;
pC = pR + (dM - dC) ur;

(* Modelo de medida del tiempo *)
mt = T0 + dC S + lc sn;
(* Modelo de medida de atenacion de la luz *)
ma = A Exp[-lc / alen];
(* Vector de modelos*)
vm = {mt, ma};
```

```
(* Todas las derivadas para el jacobiano *)
dtdx0 = FullSimplify[D[mt, X0]];
dtdxp = FullSimplify[D[mt, XP]];
dtdy0 = FullSimplify[D[mt, Y0]];
dtdyp = FullSimplify[D[mt, YP]];
dtdt0 = FullSimplify[D[mt, T0]];
dtds = FullSimplify[D[mt, S]];
dtda = FullSimplify[D[mt, A]];

dadx0 = FullSimplify[D[ma, X0]];
dadxp = FullSimplify[D[ma, XP]];
dady0 = FullSimplify[D[ma, Y0]];
dadyp = FullSimplify[D[ma, YP]];
dadt0 = FullSimplify[D[ma, T0]];
dads = FullSimplify[D[ma, S]];
dada = FullSimplify[D[ma, A]];

(* Matriz jacobiana para el detector en i *)
mGi = {{dtdx0, dtdxp, dtdy0, dtdyp, dtdt0, dtds, dtda},
   {dadx0, dadxp, dady0, dadyp, dadt0, dads, dada}};
MatrixForm[%];
mGiT = Transpose[mGi];
vgi = vm - mGi.vsa;
MatrixForm[%];
vw = {wT, wA};
mW = DiagonalMatrix[vw];

(* Matriz de configuracion mK *)
mKi = mGiT.mW.mGi;
MatrixForm[%];
(* Vector reducido de datos *)
vdi = {dti, dai};
vai = mW.(vdi - vgi);
MatrixForm[%];
```

Example: Antares observatory. Reduced matrices

σ_t = 1ns
σ_A= 1%

Out[149]//MatrixForm=

$$\begin{pmatrix} 9.9963 & -0.92283 & -0.276968 & 0.372119 & 0.201969 & -0.209105 & -0.201514 \\ 0 & 0.000934888 & 0.372119 & -0.3192 & 0.181463 & -0.177195 & -0.184938 \\ 0 & 0 & 9.9963 & -0.92283 & 0.201969 & -0.209105 & -0.201514 \\ 0 & 0 & 0 & 0.000934888 & 0.181463 & -0.177195 & -0.184938 \\ 0 & 0 & 0 & 0 & 3.02582 & -0.993096 & -0.992733 \\ 0 & 0 & 0 & 0 & 0 & 0.0000857084 & 0.999526 \\ 0 & 0 & 0 & 0 & 0 & 0 & 47.33 \end{pmatrix}$$

Out[150]= 0.00256868

Out[151]= 0.004733

σ_t = 1000 ns
σ_A= 1%

Out[142]//MatrixForm=

$$\begin{pmatrix} 10.2391 & -0.865922 & -0.248224 & 0.392817 & 0.00297504 & -0.257847 & -0.252379 \\ 0 & 0.000962993 & 0.392817 & -0.27516 & 0.00303944 & -0.255216 & -0.260764 \\ 0 & 0 & 10.2391 & -0.865922 & 0.00297504 & -0.257847 & -0.252379 \\ 0 & 0 & 0 & 0.000962993 & 0.00303944 & -0.255216 & -0.260764 \\ 0 & 0 & 0 & 0 & 353.578 & -0.0117266 & -0.0117243 \\ 0 & 0 & 0 & 0 & 0 & 0.000118199 & 0.999749 \\ 0 & 0 & 0 & 0 & 0 & 0 & 65.2226 \end{pmatrix}$$

Out[143]= 0.00354242

Out[144]= 0.00652226

σ_t = 1 ns
σ_A= 1000%

Out[142]//MatrixForm=

$$\begin{pmatrix} 69.8128 & -0.884344 & -0.0509347 & 0.254027 & 0.380043 & -0.387414 & -0.0810417 \\ 0 & 0.00610635 & 0.254027 & -0.238725 & 0.0900304 & -0.0824782 & -0.0204867 \\ 0 & 0 & 69.8128 & -0.884344 & 0.380043 & -0.387414 & -0.0810417 \\ 0 & 0 & 0 & 0.00610635 & 0.0900304 & -0.0824782 & -0.0204867 \\ 0 & 0 & 0 & 0 & 16.0319 & -0.99967 & -0.215362 \\ 0 & 0 & 0 & 0 & 0 & 0.00046494 & 0.21539 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1243.78 \end{pmatrix}$$

Out[143]= 0.0139343

Out[144]= 0.124378

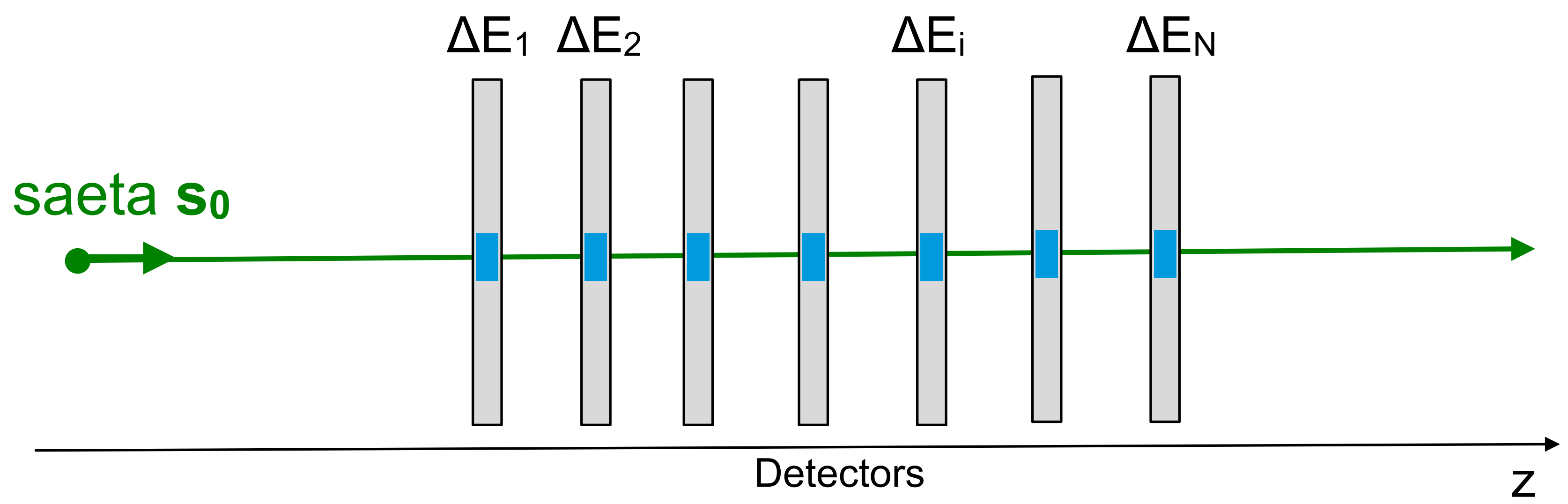
timTrack

TimTrack vs Kalman Filter

TimTrack		
Parameter space		Measurement space
$\mathbf{s}_p, \mathcal{E}_p$		
	F: Transport	
$\mathbf{s} = F \cdot \mathbf{s}_p$ $\mathcal{E}_s = F \cdot \mathcal{E}_p \cdot F'$		$\mathbf{d}$, $W_d = V_d^{-1}$
	G: Measurement $\mathbf{m}(\mathbf{s}) = G \cdot \mathbf{s} + \mathbf{g}(\mathbf{s})$	$V_s = G \cdot \mathcal{E}_s \cdot G'$
		$\mathbf{d}_c = \mathbf{d} - \mathbf{g}(\mathbf{s})$ $V_c = V_d + V_s$
$\mathbf{s}_d = C \cdot \mathbf{d}_c$ $\delta\mathcal{E}_s = \mathcal{E}_s \cdot W_s \cdot \mathcal{E}_s$	$C = \mathcal{E}_s \cdot G' \cdot V_c^{-1}$ $W_s = G' \cdot V_c^{-1} \cdot G$	
$\mathbf{s}_{p+1} = (I - \mathcal{E}_s \cdot W_s) \cdot (\mathbf{s} + \mathbf{s}_d)$ $\mathcal{E}_{p+1} = \mathcal{E}_s - \delta\mathcal{E}_s$		

Kalman Filter		
Parameter space		Measurement space
$\mathbf{r}_p, \mathcal{E}_p$		
	F: Transport	
$\mathbf{r} = F \cdot \mathbf{r}_p$ $\mathcal{E}_r = F \cdot \mathcal{E}_p \cdot F'$		$\mathbf{d}$, V_d
	H: Measurement $\mathbf{m}(\mathbf{r}) = H \cdot \mathbf{r} + \boldsymbol{\eta}$	$\mathbf{d}_r = H \cdot \mathbf{r}$ $V_r = H \cdot \mathcal{E}_r \cdot H'$
		$\delta\mathbf{d} = \mathbf{d} - \mathbf{d}_r$ $V_c = V_r + V_d$
$\delta\mathbf{r} = K \cdot \delta\mathbf{d}$ $\delta\mathcal{E}_r = \mathcal{E}_r \cdot W_r \cdot \mathcal{E}_r$	$K = \mathcal{E}_r \cdot H' \cdot V_c^{-1}$ $W_r = H' \cdot V_c^{-1} \cdot H$	
$\mathbf{r}_{p+1} = \mathbf{r} + \delta\mathbf{r}$ $\mathcal{E}_{p+1} = \mathcal{E}_r - \delta\mathcal{E}_r$		

Energy loss (Bethe Bloch)



Energy losses are related to the velocity (β) of a particle through the Bethe- Bloch formula:

$$-\frac{dE}{dx} \simeq k \cdot \frac{1}{\beta^2}\left(\frac{1}{2}\ln\frac{2m_e c^2\beta^2\gamma^2 T_{max}}{I^2} - \beta^2 - \frac{\delta(\beta\gamma)}{2}\right)$$

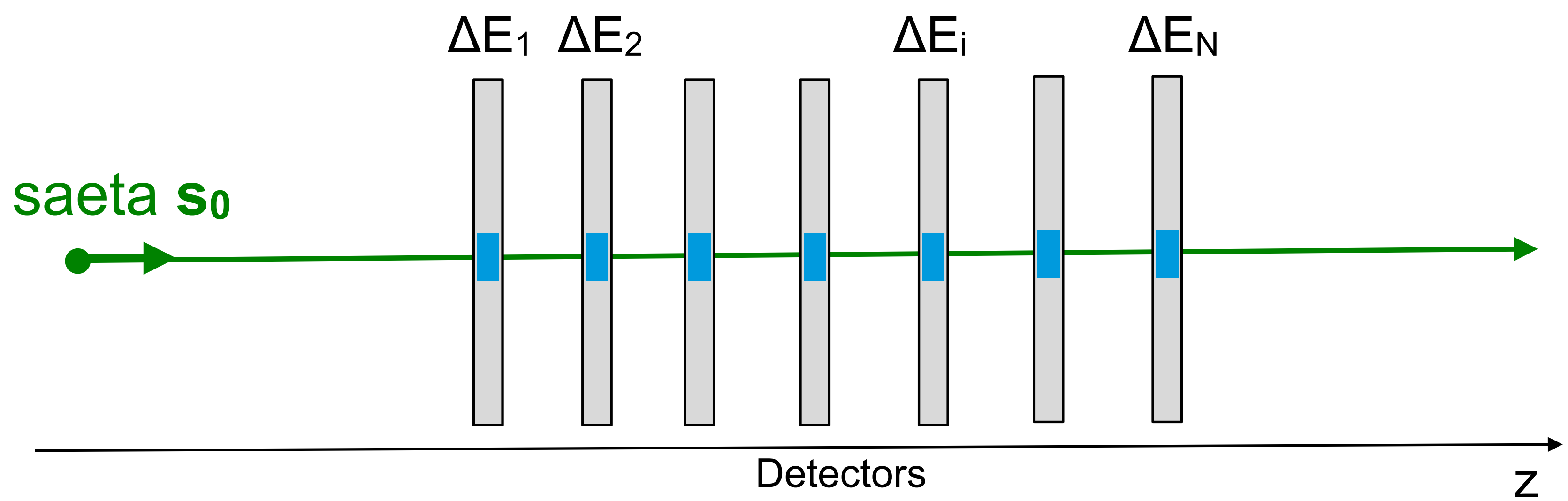
that can be written in a simplified form as:

$$\boxed{-\frac{dE}{dx} \simeq k \cdot \frac{1}{\beta^2}\left(\frac{1}{2}\ln\frac{1}{I_c^2}\frac{\beta^2}{(1-\beta^2)} - 1\right)}$$

with:

$$I_c = \frac{I}{2m_e c^2}$$

Energy loss (Bethe Bloch)



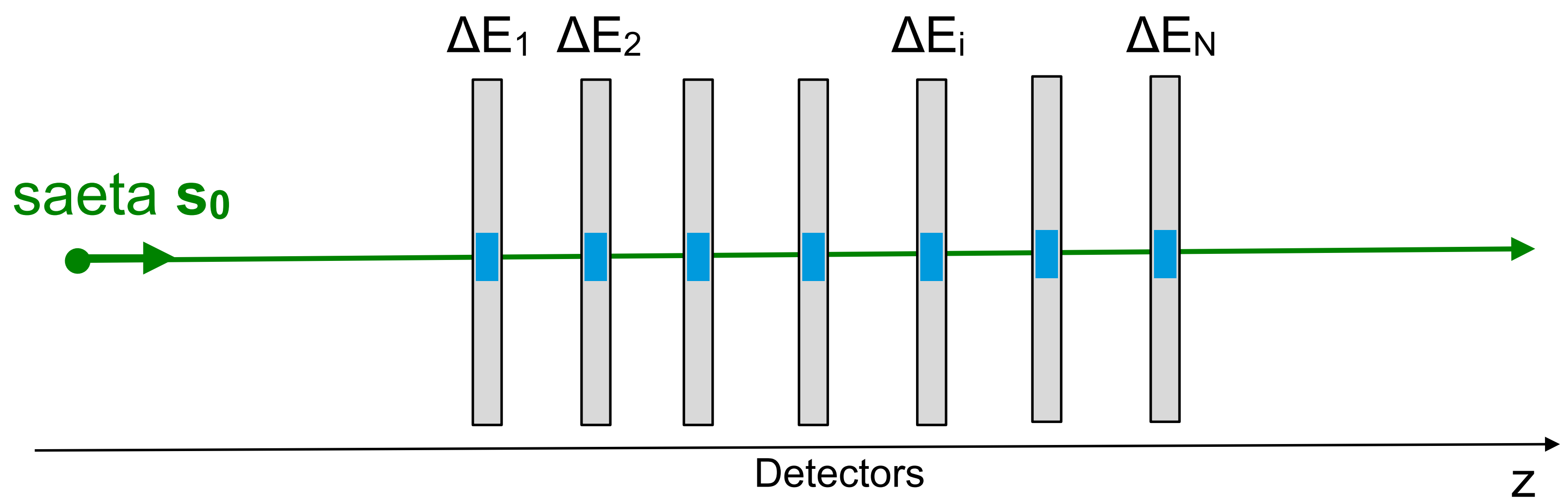
The Bethe- Bloch formula can be written as funcion of the slowness S (=1/V) as:

$$-\frac{dE}{dx} \simeq k \cdot S^2 \left(\frac{1}{2} \ln \frac{2 m_e c^2 T_{max}}{(S^2-1)^2 I^2} - \frac{1}{S^2} - \frac{\delta(S)}{2} \right)$$

that can be approximated and simplified as:

$$\boxed{-\frac{dE}{dx} \simeq k \left(S^2 \ln \frac{1}{(S^2-1) I_c} - 1 \right)}$$

Energy loss (Bethe Bloch)

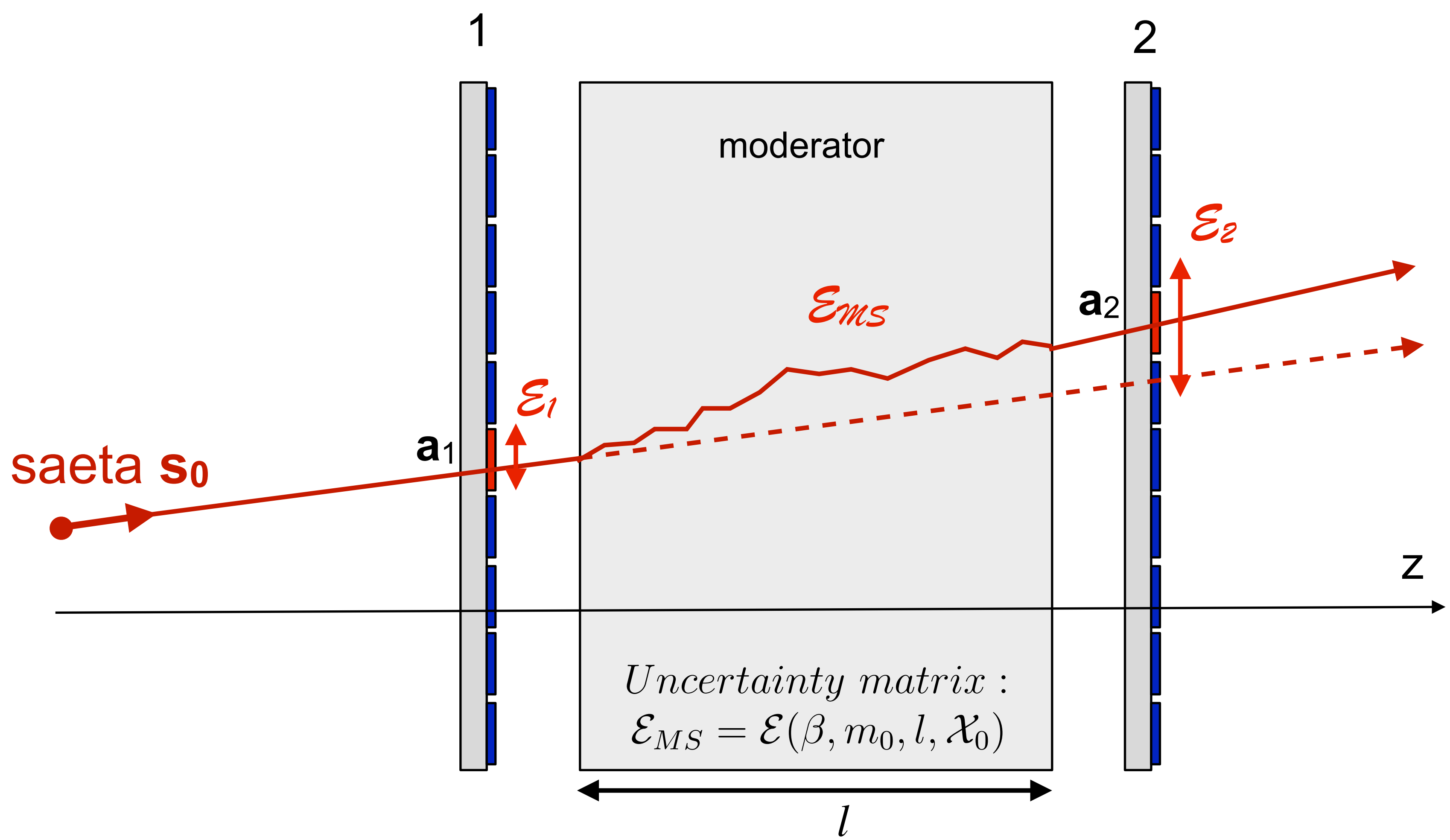


Observe that the equation:

$$\boxed{-\frac{dE}{dx} \simeq k\left(S^2 \ln\frac{1}{(S^2-1)I_c} - 1\right)}$$

relates the energy loss measurements to the unknown parameter S and, as consequence, those measurements (either separately for each detector or promediated to all the detectors) can be included as a new model in the formalism.

Multiple Scattering



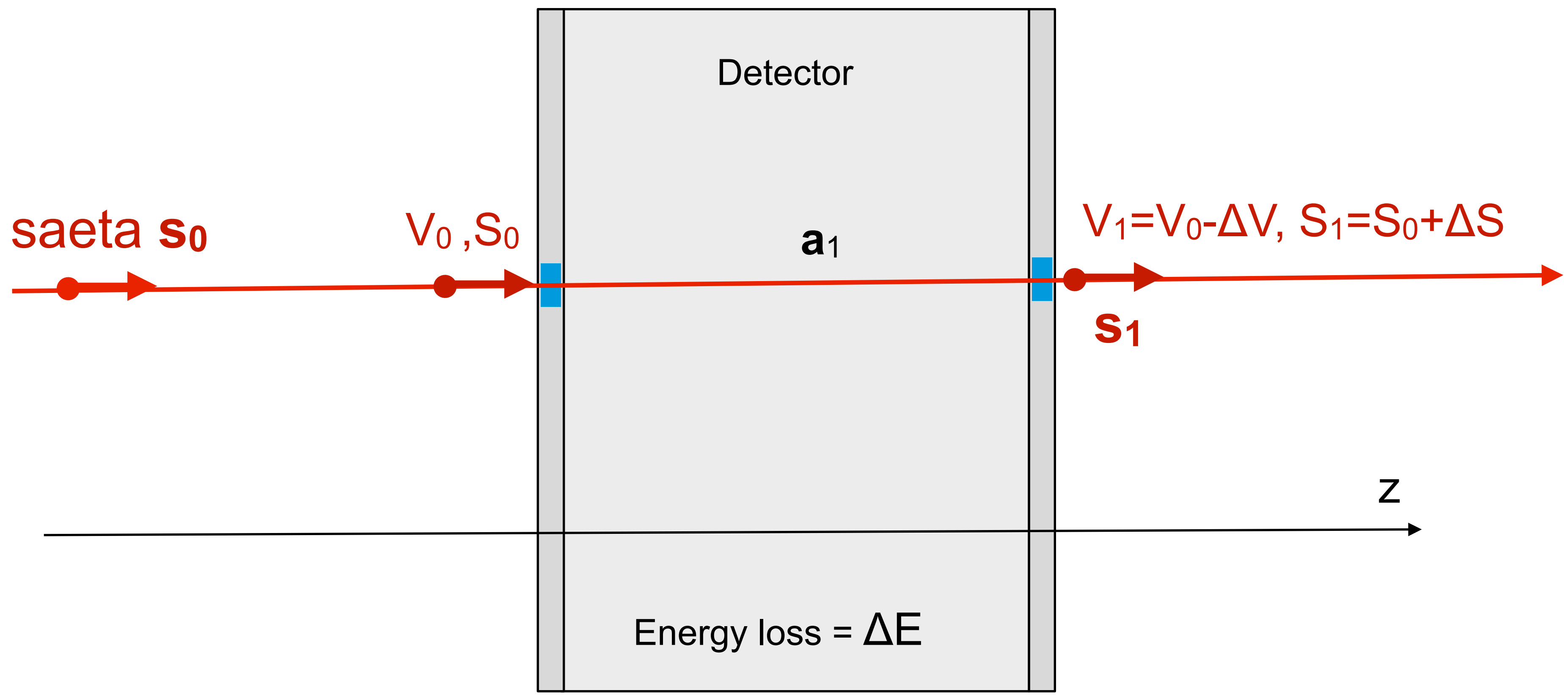
K matrices: K_1 $K_{MS} = \mathcal{E}_{MS}^{-1}$ $K_2 = \mathcal{E}_2^{-1}$

Following the TT formalism, we expect:

- $\mathbf{a}_1 = K_1\, \mathbf{s}_0$
- $\mathbf{a}_2 = K_{MS,2}\, \mathbf{s}_0$ with $K_{MS,2} = (\mathcal{E}_{MS} + \mathcal{E}_2)^{-1}$

$\Rightarrow$ $\boxed{\mathbf{s} = K_1^{-1}\, \mathbf{a}_1 + K_{MS,2}^{-1}\, \mathbf{a}_2}$

Energy loss



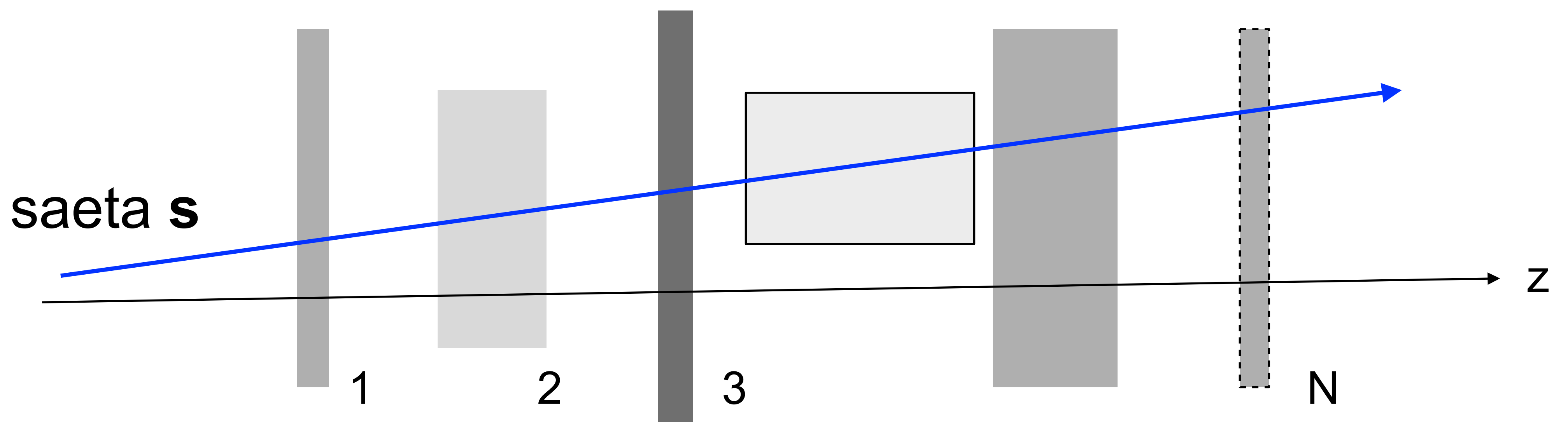
The slowness S (=1/V) increases with the loss of energy, ΔE, according to:

$$\boxed{\Delta S = -\frac{(S^2-1)^{\frac{3}{2}}}{m_0}\,\Delta E} \qquad \left(\Delta\beta = \frac{(1-\beta^2)^{\frac{3}{2}}}{\beta\, m_0}\,\Delta E\right)$$

If ΔE has been well measured, the increasing in the slowness S can be introduced as a constraint to improve the fit.

Set of several different detectors

If we have a set of different detectors (different models):



models:	$\mathbf{m}_1$	$\mathbf{m}_2$	$\mathbf{m}_3$	$\mathbf{m}_4$	$\mathbf{m}_5$	....
data (f.ex):	(x,y)	(x,t)	(x,y,t)	(β)	(γ)	....
K matrices:	K_1	K_2	K_3	K_4	K_5	....
a vectors:	$\mathbf{a}_1$	$\mathbf{a}_2$	$\mathbf{a}_3$	$\mathbf{a}_4$	$\mathbf{a}_5$	

and, finally:

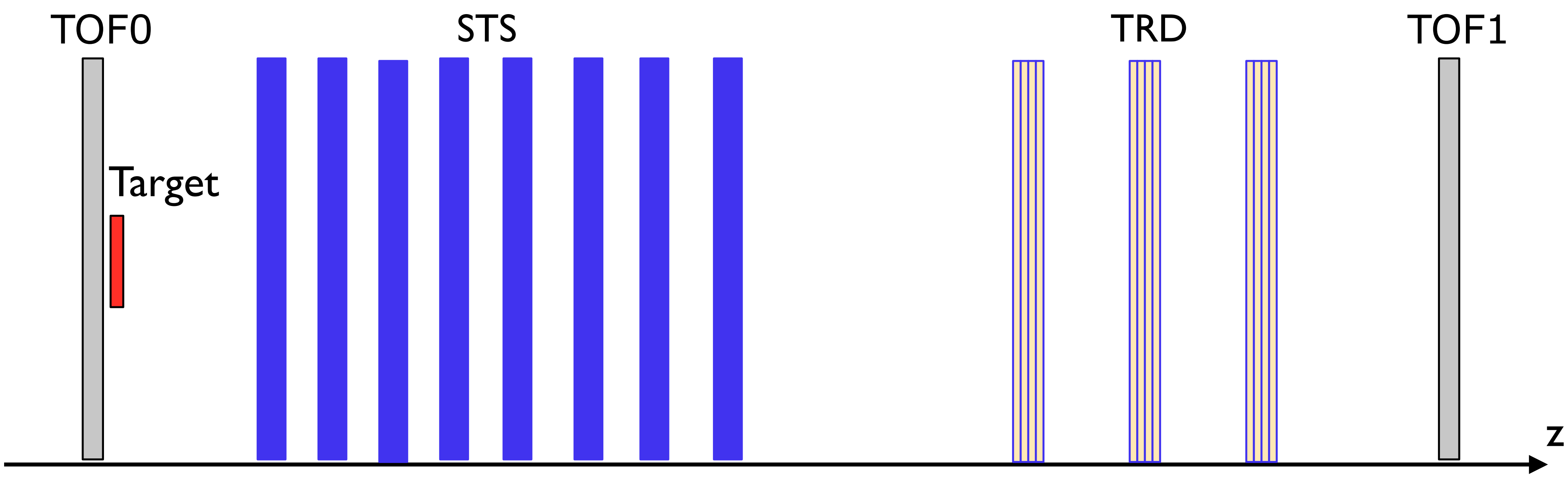
$$K = \sum K_i$$
$$\mathbf{a} = \sum \mathbf{a}_i$$

$$\rightarrow \boxed{\mathbf{s} = K^{-1}\mathbf{a}}$$

CBM

Example: D_0 decay analysis at CBM (GSI)

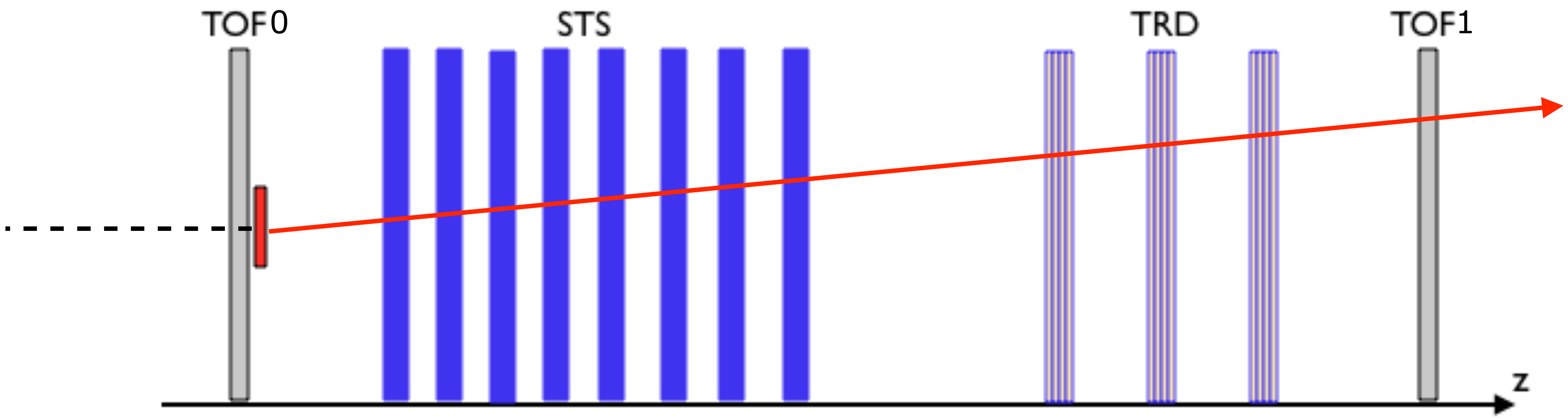
Layout



z(cm)	0	30 35 40 50 60 75 95 100	600 700 800	1000
σ_x	1cm	~17μm	0.3mm	1cm
σ_y	1cm	~17μm	2.5mm	1cm
σ_t	80ps	$3 \cdot 10^6$ps	2000ps	80ps

timTrack

Example: D_0 decay analysis at CBM (GSI)



Single Track

-Track fitting model:

$$x_i = X_0 + X' \cdot z_i$$
$$y_i = Y_0 + Y' \cdot z_i$$
$$t_i = T_0 + S \cdot \sqrt{1 + X'^2 + Y'^2} \cdot z_i$$

$$S = \frac{1}{V}$$

SAETA: **s**=(X_0, X', Y_0, Y', T_0, S=1/V) (6 parameters)

timTrack

Single Track: “reduced” Error Matrices

σ’s are in the diagonal

corr coefs.

X0	X’	Y0	Y’	T0	S=1/V
8.0311e-003	-5.8000e-001	0	0	0	0
	3.3457e-006	0	0	0	0
		8.0558e-003	-5.8000e-001	0	0
			3.4401e-006	0	0
				8.6771e+001	-4.1000e-001
					5.2507e-002

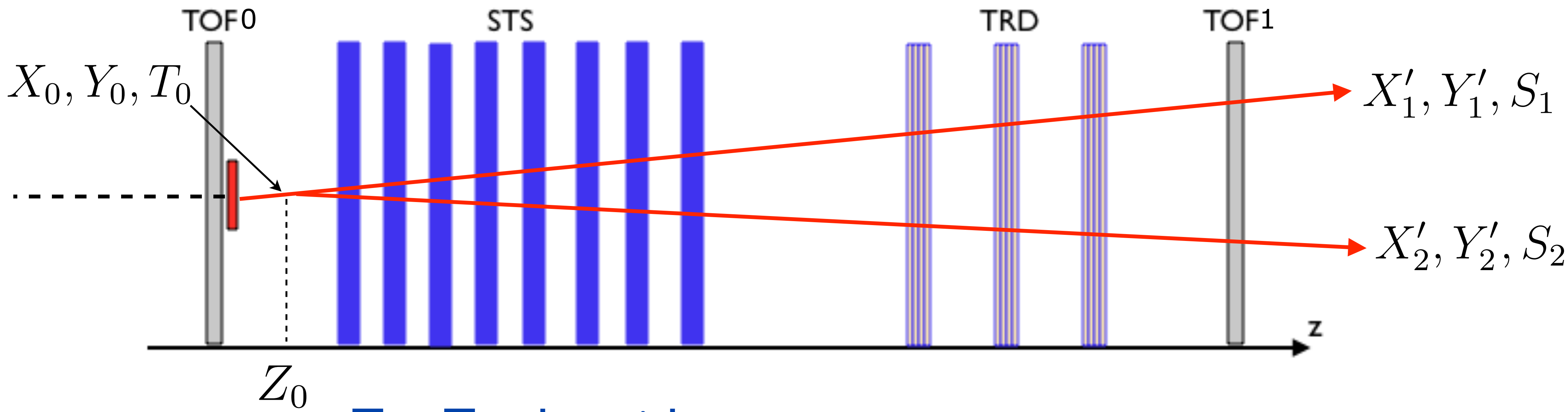
(Only TOF1)

X0	X’	Y0	Y’	T0	S=1/V
9.6829e-003	-7.4000e-001	0	0	0	0
0	1.2189e-005	0	0	0	0
		1.7297e-002	-9.3000e-001	0	0
			2.9081e-005	0	0
				7.9927e+001	-7.1000e-001
					7.2659e-003

(TOF0 and TOF1)

timTrack

Example: D_0 decay analysis at CBM (GSI)



Two Tracks with common vertex:

-Track fitting model:

$$x_i^1 = X_0 + X_1' \cdot z_i$$
$$y_i^1 = Y_0 + Y_1' \cdot z_i$$
$$t_i^1 = T_0 + S_1 \cdot \sqrt{1 + X_1'^2 + Y_1'^2} \cdot z_i$$
$$x_i^2 = X_0 + X_2' \cdot z_i$$
$$y_i^2 = Y_0 + Y_2' \cdot z_i$$
$$t_i^2 = T_0 + S_2 \cdot \sqrt{1 + X_2'^2 + Y_2'^2} \cdot z_i$$

multi-SAETA: **s** = (X_0, X_1', X_2', Y_0, Y_1', Y_2', S_1, S_2, T_0, Z_0) (10 parameters)

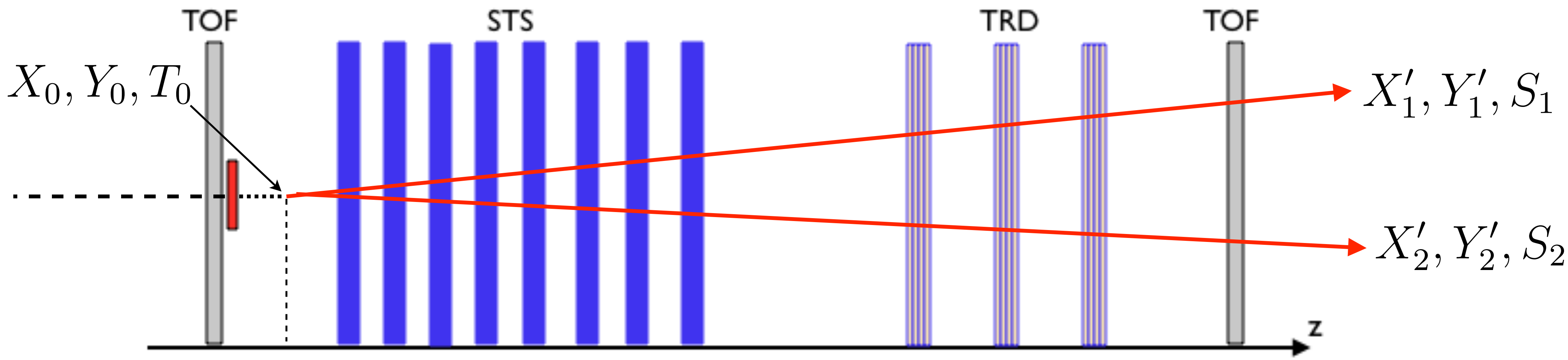
timTrack

TwoTracks with common vertex: "Reduced" error matrix

X_0	X'_1	X'_2	Y_0	Y'_1	Y'_2	S_1	S_2	T_0	Z_0
5.6793e-003	-4.5000e-001	-4.5000e-001	0	0	0	0	0	0	1.0000e-002
	3.0595e-006	2.0000e-001	1.0000e-002	-3.0000e-002	3.0000e-002	0	0	0	-8.0000e-002
		3.0595e-006	-1.0000e-002	3.0000e-002	-3.0000e-002	0	0	0	8.0000e-002
			5.7078e-003	-4.4000e-001	-3.9000e-001	0	0	0	-6.0000e-002
				3.4300e-006	1.0000e-002	0	0	0	4.1000e-001
					3.4300e-006	0	0	0	-4.1000e-001
						7.0578e-002	9.0000e-002	-3.0000e-001	0
							6.8720e-002	-3.0000e-001	0
								$\boxed{6.1356e+001}$	0
									1.1150e-002

Numerical estimation for D0's with K=2GeV at polar angle 0 and decay with azimuthal angle= 45°

Example: D_0 decay analysis at CBM (GSI)



TwoTracks with common vertex and mass constraint

-Track fitting model = Two Tracks with common vertex
- Mass constraint ($\mathbf{f}(\mathbf{s})=0$):

$$m(D_0) - \sqrt{(m_K^2 + m_\pi^2 + m_K \cdot m_{pi} \cdot \gamma_K \cdot \gamma_{pi}(1 + \beta_K \cdot \beta_\pi \cdot cos(\alpha))} = 0$$

multi-SAETA: $\mathbf{s}$ = (X_0, X_1', X_2', Y_0, Y_1', Y_2', S_1, S_2,T_0, Z_0; m1,m2) (10 parameters)

TwoTracks with D0 mass constraint: "Reduced" error matrix

X_0	X'_1	X'_2	Y_0	Y'_1	Y'_2	S_1	S_2	T_0	Z_0
5.6793e-003	-4.5000e-001	-4.5000e-001	0	0	0	0	0	0	1.0000e-002
	3.0595e-006	2.0000e-001	1.0000e-002	-3.0000e-002	3.0000e-002	0	0	0	-8.0000e-002
		3.0595e-006	-1.0000e-002	3.0000e-002	-3.0000e-002	0	0	0	8.0000e-002
			5.7078e-003	-4.4000e-001	-3.9000e-001	0	0	0	-6.0000e-002
				3.4300e-006	1.0000e-002	0	0	0	4.1000e-001
					3.4300e-006	0	0	0	-4.1000e-001
						6.9434e-002	-1.0000e+000	-2.6000e-001	0
							6.1507e-003	2.6000e-001	0
								$\boxed{5.8042e+001}$	0
									1.1150e-002

Let us compare both approaches:

TwoTracks with common vertex: "Reduced" error matrix

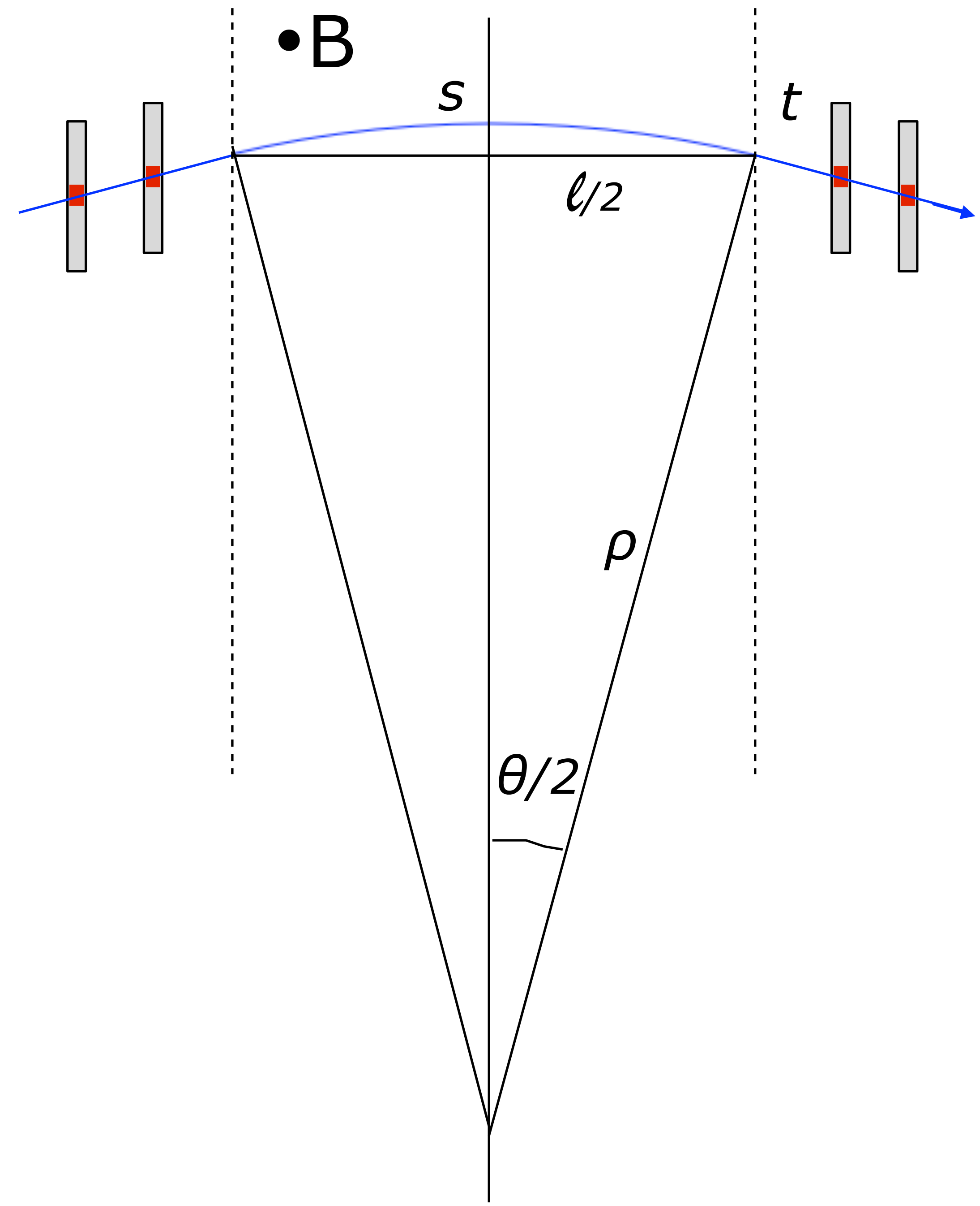
X_0	X'_1	X'_2	Y_0	Y'_1	Y'_2	S_1	S_2	T_0	Z_0
5.6793e-003	-4.5000e-001	-4.5000e-001	0	0	0	0	0	0	1.0000e-002
	3.0595e-006	2.0000e-001	1.0000e-002	-3.0000e-002	3.0000e-002	0	0	0	-8.0000e-002
		3.0595e-006	-1.0000e-002	3.0000e-002	-3.0000e-002	0	0	0	8.0000e-002
			5.7078e-003	-4.4000e-001	-3.9000e-001	0	0	0	-6.0000e-002
				3.4300e-006	1.0000e-002	0	0	0	4.1000e-001
					3.4300e-006	0	0	0	-4.1000e-001
						7.0578e-002	9.0000e-002	-3.0000e-001	0
							6.8720e-002	-3.0000e-001	0
								6.1356e+001	0
									1.1150e-002

TwoTracks with D0 mass constraint: "Reduced" error matrix

X_0	X'_1	X'_2	Y_0	Y'_1	Y'_2	S_1	S_2	T_0	Z_0
5.6793e-003	-4.5000e-001	-4.5000e-001	0	0	0	0	0	0	1.0000e-002
	3.0595e-006	2.0000e-001	1.0000e-002	-3.0000e-002	3.0000e-002	0	0	0	-8.0000e-002
		3.0595e-006	-1.0000e-002	3.0000e-002	-3.0000e-002	0	0	0	8.0000e-002
			5.7078e-003	-4.4000e-001	-3.9000e-001	0	0	0	-6.0000e-002
				3.4300e-006	1.0000e-002	0	0	0	4.1000e-001
					3.4300e-006	0	0	0	-4.1000e-001
						6.9434e-002	-1.0000e+000	-2.6000e-001	0
							6.1507e-003	2.6000e-001	0
								5.8042e+001	0
									1.1150e-002

Other developments

A particle in a magnetic field with external tracking

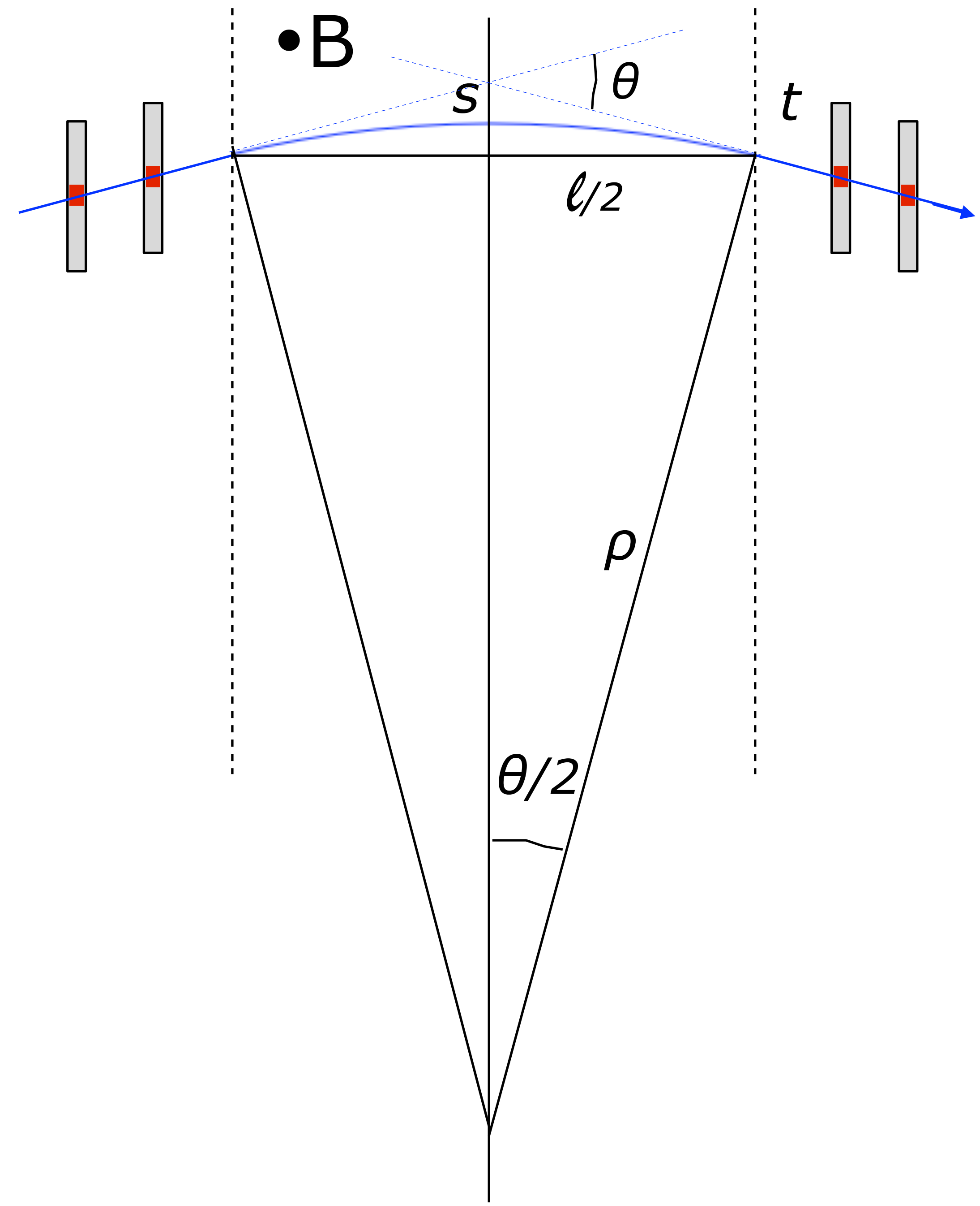


The properties of a charged particle in a uniform transverse magnetic field B may be summarized by the following important equations:

$$p = q \cdot B \cdot \rho$$

$$\omega = \frac{v}{\rho} = \frac{q \cdot B}{m}$$

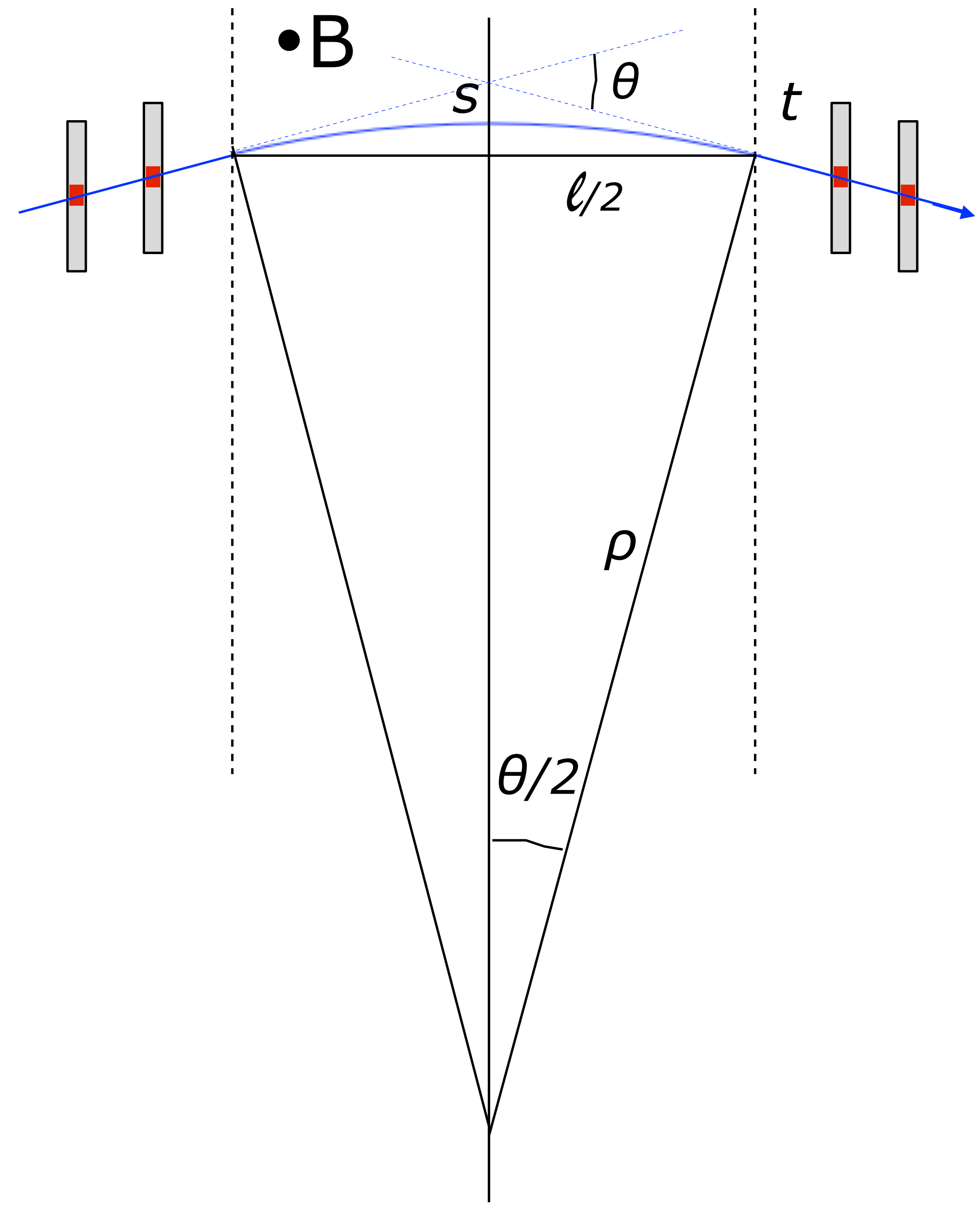
A particle in a magnetic field with external tracking



From the measurement of the deflected angle θ, we arrive to:

$$\left.\begin{aligned} p &= q \cdot B \cdot \rho \\ \sin\tfrac{\theta}{2} &= \tfrac{l/2}{\rho} \end{aligned}\right\} \Rightarrow \quad \sin\frac{\theta}{2} = \frac{q \cdot B \cdot l}{2p}$$

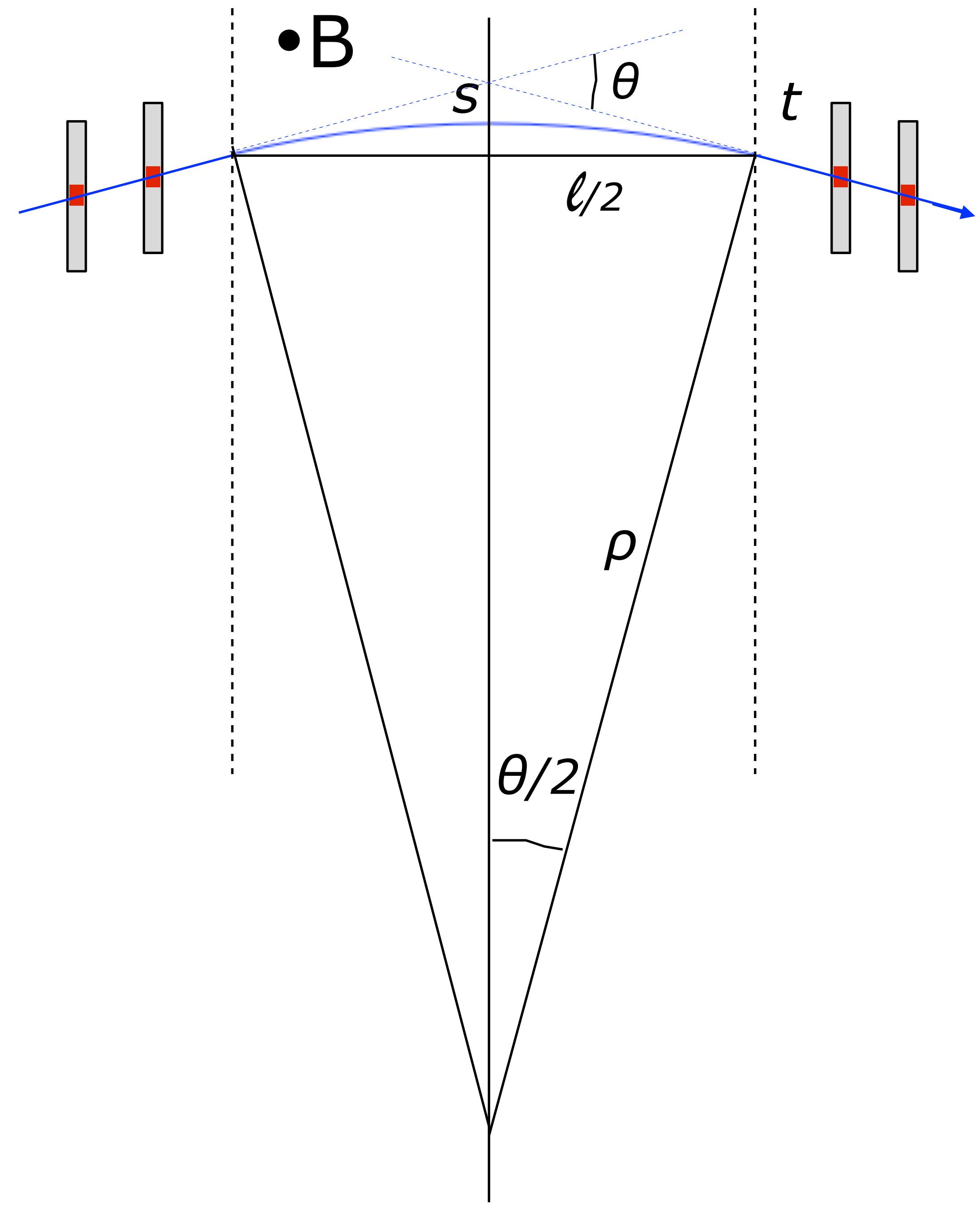
A particle in a magnetic field with external tracking



From the measurement of the flight time t, we arrive to:

$$\left.\begin{array}{l} \frac{v}{\rho} = \frac{q \cdot B}{m} \\[2em] s = \rho \cdot \theta = t \cdot v \end{array}\right\} \Rightarrow \quad t = \frac{\theta \cdot m}{q \cdot B}$$

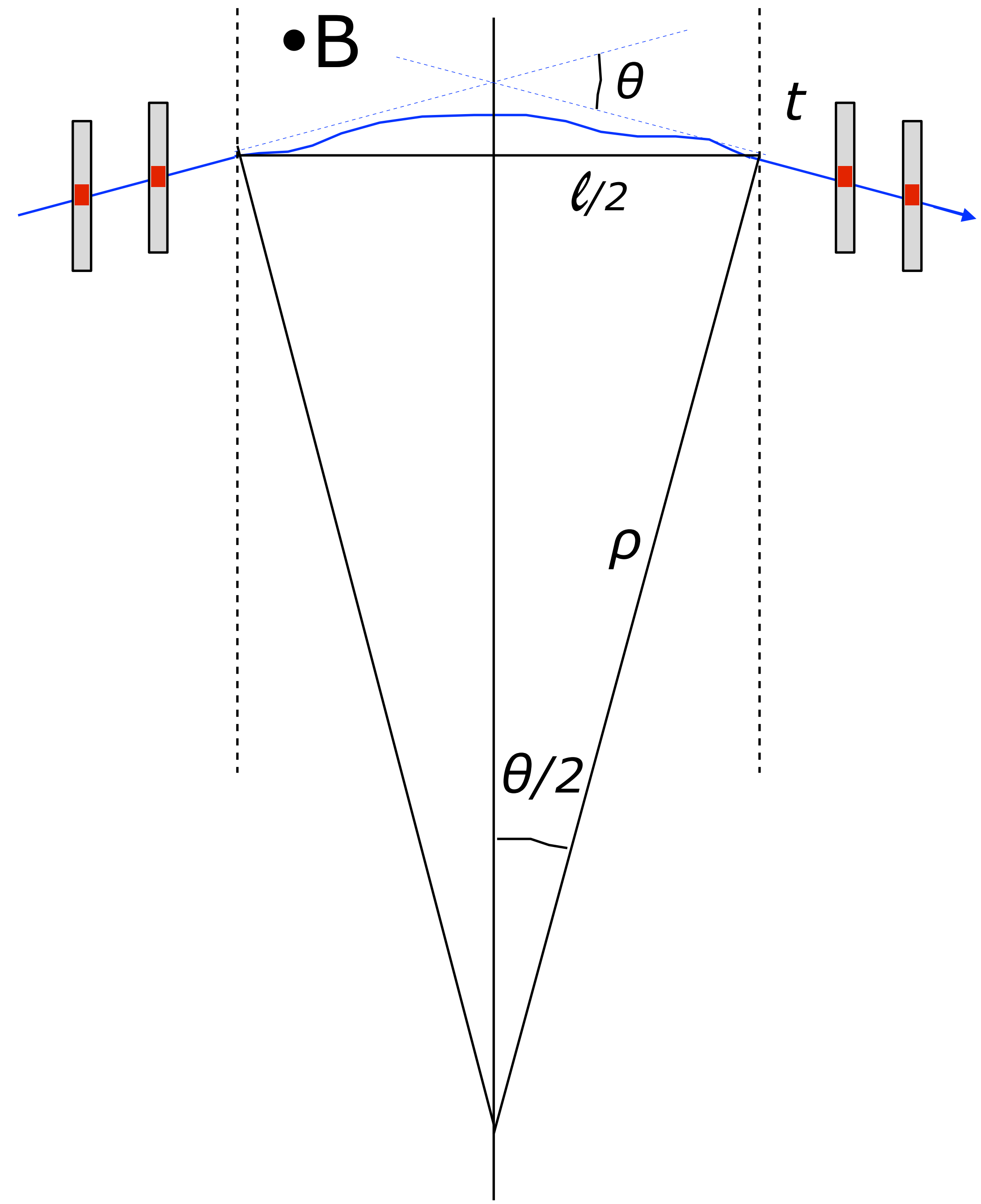
A particle in a magnetic field with external tracking



In a uniform magnetic field B, the measurements of both, the angle θ and the time t, should be linked by the couple of equations:

$$\left.\begin{aligned} \sin\frac{\theta}{2} &= \frac{B\cdot l}{2\cdot p} = \frac{q\cdot B\cdot l}{2\cdot\beta\cdot\gamma\cdot m_0} \\ t &= \frac{\theta\cdot\gamma\cdot m_0}{q\cdot B} = \frac{\theta\cdot\gamma\cdot m_0}{q\cdot B} \end{aligned}\right\}$$

A particle in a magnetic field with external tracking

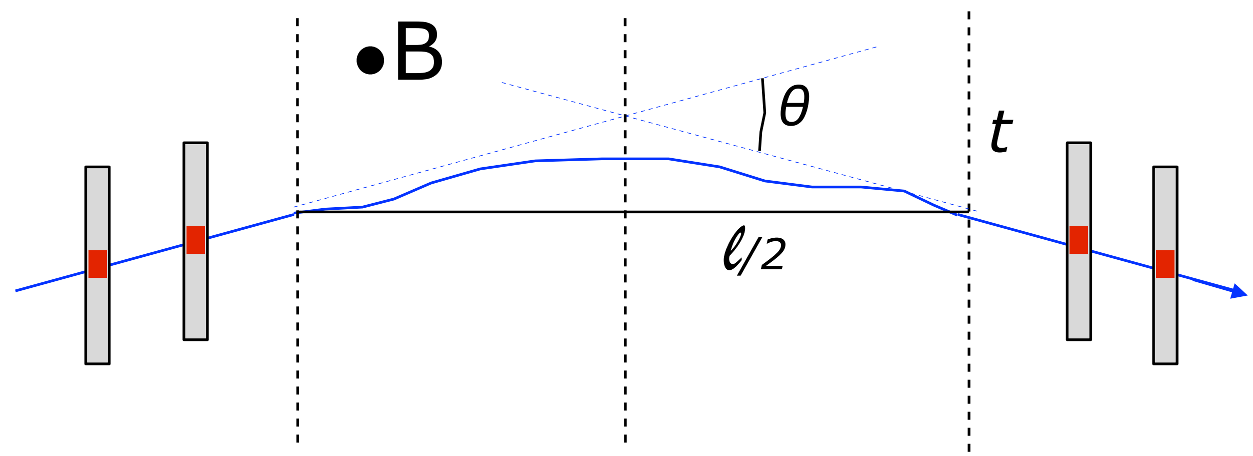


In a non-uniform magnetic field B, we can extract the velocity β from the previous equations:

$$\left.\begin{aligned} \sin\frac{\theta}{2} &= \frac{q\cdot B_\theta \cdot l}{2\cdot\beta\cdot\gamma\cdot m_0} \\ t &= \frac{\theta\cdot\gamma\cdot m_0}{q\cdot B_t} \end{aligned}\right\} \Rightarrow \beta = \left(\frac{B_\theta}{B_t}\right)\frac{l\cdot\theta}{2t\cdot\sin\frac{\theta}{2}}$$

where the ratio $n_B = (B_\theta/B_t)$ takes into account that both values of the magnetic field B_θ and B_t used to measure the angle θ and the time t, respectively, may not be equal.

Fitting saetas in a magnetic field



$\mathbf{s}_1$=(X0,X',Y0,Y',T0,S) $\mathbf{s}_2$=(X0,X'+θ,Y0,Y',T0+t,S)

$\mathbf{s}_1$ and $\mathbf{s}_2$ can be fitted a the same time to all the data at both sides of the magnetic field including a few new free parameters (θ,t) that are constrained by the equations:

$$\left.\begin{aligned} \sin\frac{\theta}{2} &= \frac{B_\theta \cdot l}{2\cdot\beta\cdot\gamma\cdot m_0} \\ t &= \frac{\theta\cdot\gamma\cdot m_0}{q\cdot B_t} \end{aligned}\right\} \Rightarrow \beta = \left(\frac{B_\theta}{B_t}\right)\frac{t\cdot\theta}{2\sin\frac{\theta}{2}}$$

TimTrack as calibration tool

The model of the data, **m**, can be expanded as a function of both, the set of parameters **s** and the set of some alignment and calibration constants **α**:

$$\mathbf{m} = \mathbf{m}(\mathbf{s};\boldsymbol{\alpha})$$

and

$$\mathbf{m}(\mathbf{s};\boldsymbol{\alpha}) = \left(\frac{\partial \mathbf{m}}{\partial \mathbf{s}}\right)\mathbf{s} + \left(\frac{\partial \mathbf{m}}{\partial \boldsymbol{\alpha}}\right)\boldsymbol{\alpha} + \mathbf{g}(\mathbf{s};\boldsymbol{\alpha})$$

$$\mathbf{m}(\mathbf{s},\boldsymbol{\alpha}) = \left(\frac{\partial \mathbf{m}}{\partial \mathbf{s}} \quad \frac{\partial \mathbf{m}}{\partial \boldsymbol{\alpha}}\right)\begin{pmatrix} \mathbf{s} \\ \boldsymbol{\alpha} \end{pmatrix} + \mathbf{g}(\mathbf{s};\boldsymbol{\alpha})$$

$$\mathbf{m}(\mathbf{s};\boldsymbol{\alpha}) = G_A \cdot \mathbf{s}_\alpha + \mathbf{g}(\mathbf{s};\boldsymbol{\alpha})$$

where: $G_A = \begin{pmatrix} \frac{\partial \mathbf{m}}{\partial \mathbf{s}} & \frac{\partial \mathbf{m}}{\partial \boldsymbol{\alpha}} \end{pmatrix}$ and $\mathbf{s}_\alpha = \begin{pmatrix} \mathbf{s} \\ \boldsymbol{\alpha} \end{pmatrix}$

TimTrack as calibration tool

The matrix G_A takes the form:

$$G_A = \begin{pmatrix} \frac{\partial m_1}{\partial s_1} & \dots & \frac{\partial m_1}{\partial s_i} & \dots & \frac{\partial m_1}{\partial s_n} & \frac{\partial m_1}{\partial \alpha_1} & \dots & \frac{\partial m_1}{\partial \alpha_j} & \dots & \frac{\partial m_1}{\partial \alpha_m} \\ \vdots & & \vdots & & \vdots & \vdots & & \vdots & & \vdots \\ \vdots & & \vdots & & \vdots & \vdots & & \vdots & & \vdots \\ \frac{\partial m_d}{\partial s_1} & \dots & \frac{\partial m_d}{\partial s_i} & \dots & \frac{\partial m_d}{\partial s_n} & \frac{\partial m_d}{\partial \alpha_1} & \dots & \frac{\partial m_d}{\partial \alpha_j} & \dots & \frac{\partial m_d}{\partial \alpha_m} \end{pmatrix}$$

TimTrack as a calibration tool

Particles with well known parameters may be used to determine the unknown calibration constants. The columns corresponding both to the known parameters and to the known calibration constants can be eliminated from G_A before calculating the K matrix and finding the solution $\mathbf{s}_\alpha$.

$$G_A = \begin{pmatrix} \frac{\partial m_1}{\partial s_1} & \cdots & \frac{\partial m_1}{\partial s_i} & \cdots & \frac{\partial m_1}{\partial s_n} & \frac{\partial m_1}{\partial \alpha_1} & \cdots & \frac{\partial m_1}{\partial \alpha_j} & \cdots & \frac{\partial m_1}{\partial \alpha_m} \\ \vdots & & \vdots & & \vdots & \vdots & & \vdots & & \vdots \\ \vdots & & \vdots & & \vdots & \vdots & & \vdots & & \vdots \\ \frac{\partial m_d}{\partial s_1} & \cdots & \frac{\partial m_d}{\partial s_i} & \cdots & \frac{\partial m_d}{\partial s_n} & \frac{\partial m_d}{\partial \alpha_1} & \cdots & \frac{\partial m_d}{\partial \alpha_j} & \cdots & \frac{\partial m_d}{\partial \alpha_m} \end{pmatrix}$$

The algorithm can be used to calculate the calibration constants related with positions, time and velocity, their associated correlations and the possible correlation within them and the fitted parameters.

TimTrack as calibration tool

We develope the formalism as always:

$$K_A = G'_A \cdot W \cdot G_A$$
$$\mathbf{a}_{\boldsymbol{\alpha}} = G'_A \cdot W \cdot (\mathbf{d} - \mathbf{g}(\mathbf{s};\boldsymbol{\alpha}))$$

and

$$\mathbf{s}_{\boldsymbol{\alpha}} = K_A^{-1} \cdot \mathbf{a}_{\boldsymbol{\alpha}}$$

timTrack

Main features:

- TimTrack introduces both time and velocity as fitting parameters in a natural way making from the track reconstruction a physics task and not merely a mathematical task.

- TimTrack formalism have been developed for many different detectors and many layouts

- TimTrack allows to introduce any physical effect (energy loss, multiple scattering, magnetic field...) as a constraint of the fit

- TimTrack allows in a natural way to reconstruct a particle with data coming from different kind of detectors

- TimTrack allows to introduce the mass of the partice as fit constraint just from the beginning, allowing the rejection of non physical candidates at the first steps of the reconstruction. It opens the door to “mass constrained” particle reconstruction

the end

Thanks to:

Georgui Kornakov

timTrack

Summary:

- The TimTrack algorithm offers a suggestive framework for the tracking of high energy charged particles with timing detectors and in complex systems

- TT provides, together with the direction, the velocity of the particle and the offset time at a reference plane.

- TT works directly with the primary data and has a simple matrix form making it easy and fast to implement.

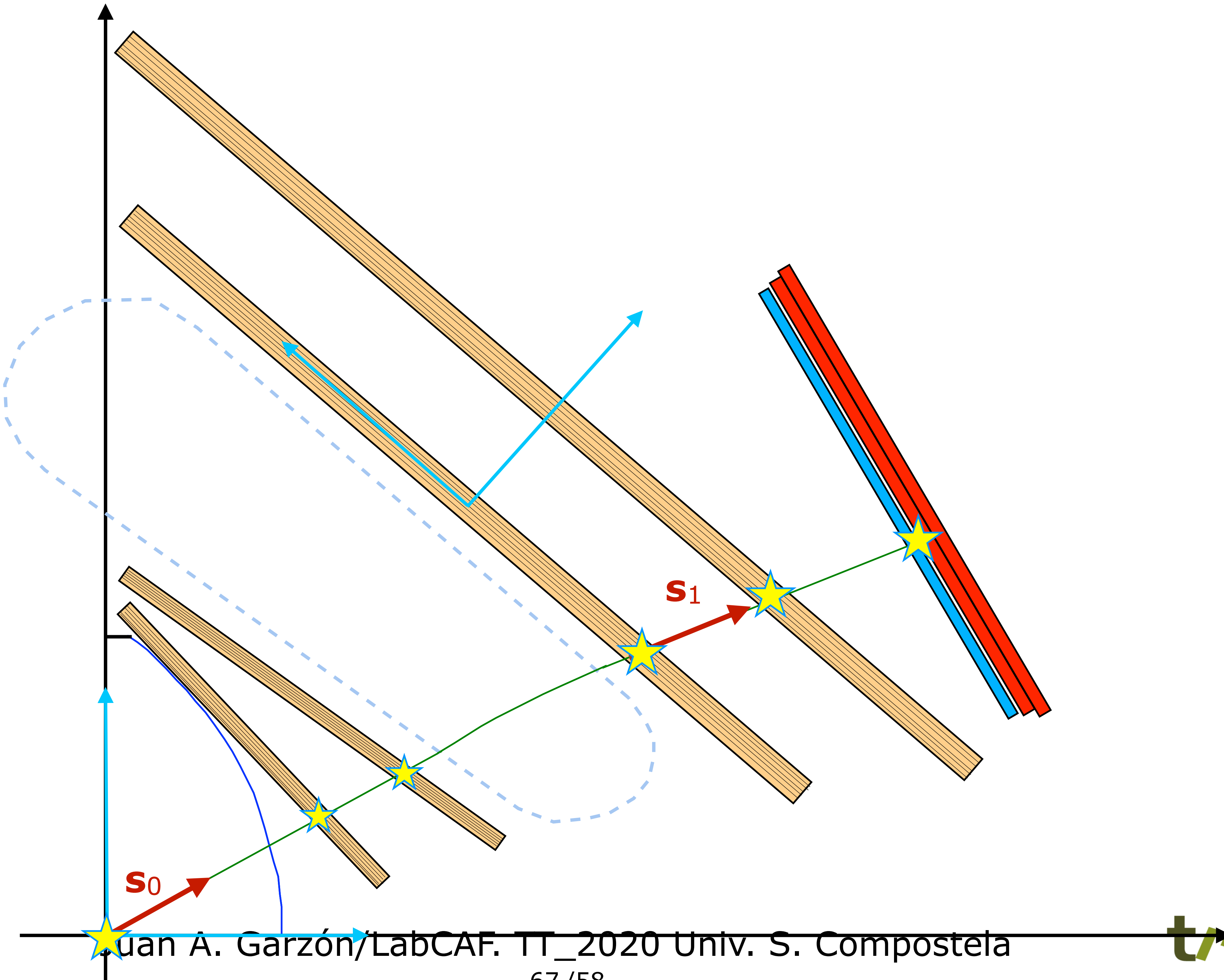
- TT deals with linear and non-linear models, including constraints among the parameters

- TT may be used for the simultaneus fit of several particles with common constraints

References (NIM):

J.A.Garzón et al: TimTrack: A new concept for the tracking of charged particles with timing detectors

J.A.Garzón et al TimTrack: A matrix formalism for a fast time and track reconstruction with timing detectors


s1
s0