# UNDERSTANDING THE KALMAN FILTER:
# TESTING ON A TOY MONTECARLO MODEL

Presents:
Federico Battisti

# KALMAN FILTER IN GENERAL

- A Kalman filter is an iterative algorithm which uses a system's physical laws of motion, known control inputs and multiple sequential measurements to form an estimate of the system's varying quantities
- At each step of the iteration an estimate of the state of the system is produced as a weighted average of the system's predicted state and of the new measurement. The weights are calculated from the covariance.
- The extended Kalman filter expands the Kalman filter technique to non-linear systems
- The models for state transition and measurement can be written as:

$$\boxed{\begin{aligned} x_k &= f(s_{k-1}, x_{k-1}) + w_{k-1} \\ z_k &= h(s_k) + v_k \end{aligned}}$$

- Where $f$ is the function of the previous state, $s_{k-1}$ , and the free parameter, $u_{k-1}$, that provides the current state $s_k$. $h$ is the measurement function that relates the current state, $x_k$, to the measurement $z_k$. $w_{k-1}$and $v_k$ are Gaussian noises for the process model and the measurement model with covariance $Q$ and $R$, respectively

# KALMAN FILTER IN GENERAL

1. Make a priori predictions for the current step's state and covariance matrix using the a posteriori best estimate of the previous step (i.e. updated using measurement)

STATE VECTOR $\hat{s}_k^- = f(\hat{s}_{k-1}^+, x_{k-1})$

COVARIANCE MATRIX $P_k^- = F_{k-1} P_{k-1}^+ F_{k-1}^T + Q$

$$F_{k-1} = \left.\frac{\partial f}{\partial s}\right|_{\hat{s}_{k-1}^+, u_{k-1}}$$

JACOBIAN

$$H_k = \left.\frac{\partial h}{\partial s}\right|_{\hat{s}_k^-}$$

CONVERSION MATRIX

$$Q$$

PROCESS NOISE COVARIANCE

Note: In the first iteration step we use step 0 estimates for the state vector and the covariance matrix $(x_0, P_0)$, which can be made very roughly

# KALMAN FILTER IN GENERAL

2. Calculate the measurement residual and the Kalman Gain

| | |
|---|---|
| RESIDUAL | $\tilde{y}_k = z_k - h(\hat{s}_k^-)$ |
| KALMAN GAIN | $K_k = P_k^- H_k^T \left(R + H_k P_k^- H_k^T\right)^{-1}$ |

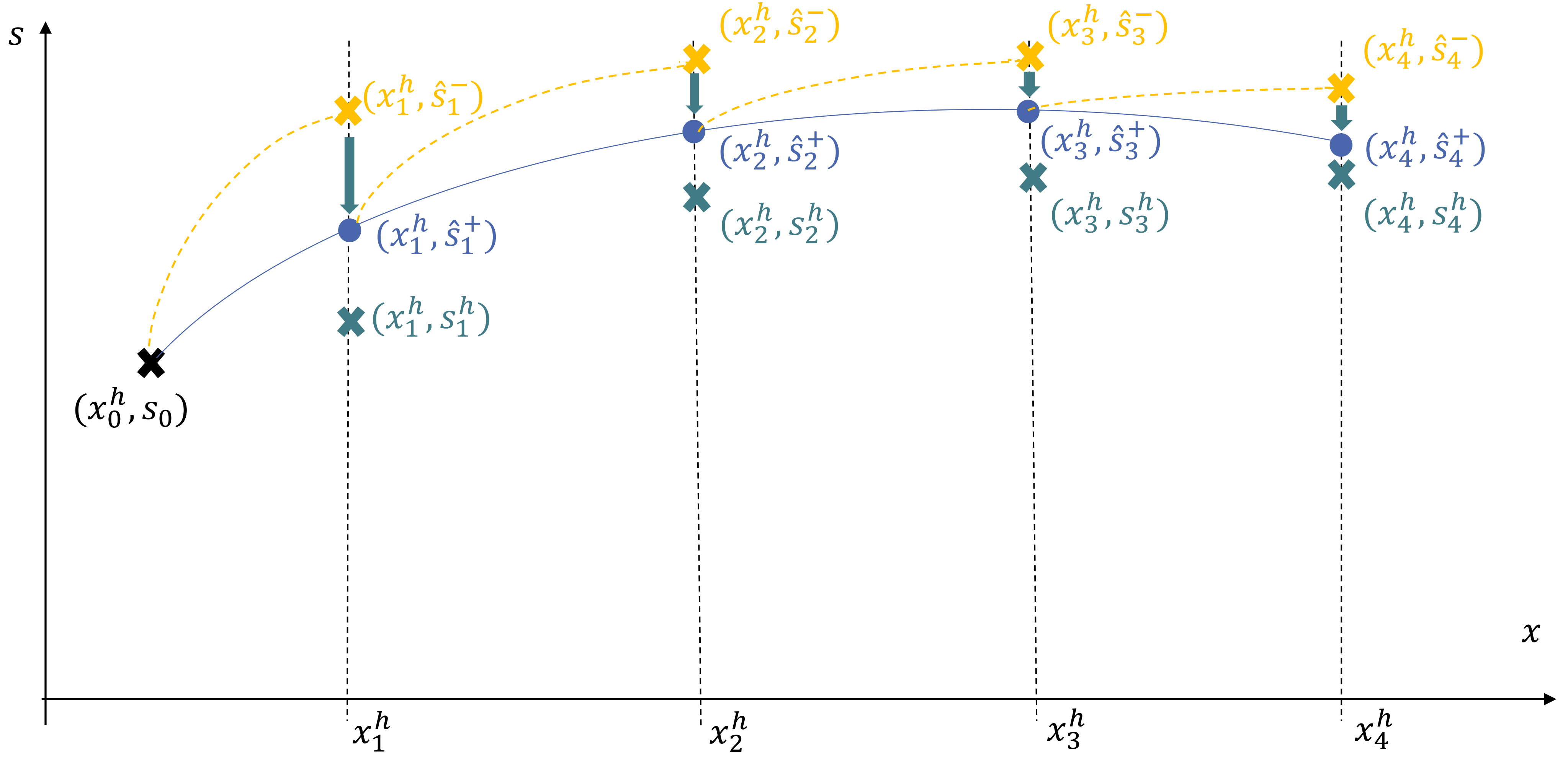
$\boxed{R}$

MEASUREMENT NOISE COVARIANCE

3. Update the estimate

| | |
|---|---|
| STATE VECTOR | $\hat{x}_k^+ = \hat{x}_k^- + K_k \tilde{y}$ |
| COVARIANCE MATRIX | $P_k^+ = (1 - K_k H_k) P_k^-$ |

# KALMAN FILTER IN GENERAL



# KALMAN FILTER APPLICATION

- We want to apply the extended Kalman filter to the motion of a charged particle in the magnetic field

$$\begin{cases} x = x_0 + r \tan\lambda\,(\phi - \phi_0) \\ y = y_C - r\cos\phi \\ z = z_C + r\sin\phi \end{cases}$$

TRACK PARAMETERS

We apply the Kalman filter to track candidates, consisting of groups of TPC clusters, which are identified and put together during the reconstruction process. Each step of the Kalman filter algorithm is identified by one of these TPC clusters

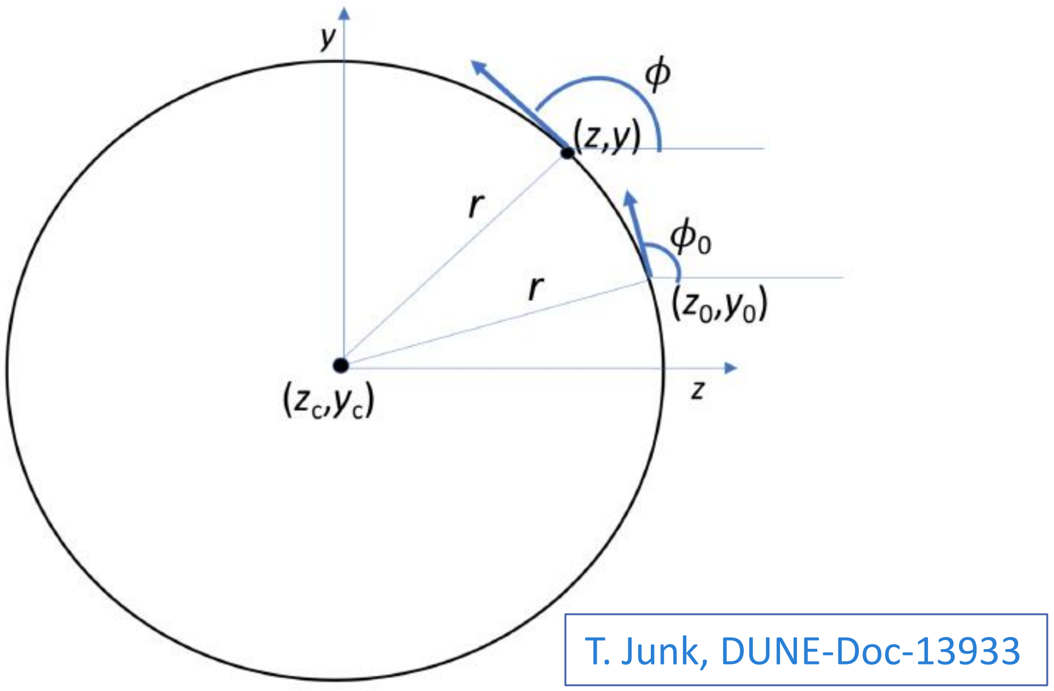



FIG. 1: Track parameter definitions in the $(z, y)$ plane.

https://docs.dunescience.org/cgi-bin/private/ShowDocument?docid=13933

# KALMAN FILTER APPLICATION: INITIAL ESTIMATES

- Before the Kalman filter algorithm can be applied, we need an initial estimate for the state vector, which in our case includes $y, z, 1/r\,, \phi$ and $\lambda$ and the covariance matrix

STATE VECTOR

$$x_0^T = (y_0 \quad z_0 \quad 1/r_0 \quad \phi_0 \quad \lambda_0) = (0 \quad 0 \quad 0.1 \quad 0 \quad 0)$$

COVARIANCE MATRIX

$$P_0 = \begin{pmatrix} 1^2 & 0 & 0 & 0 & 0 \\ 0 & 1^2 & 0 & 0 & 0 \\ 0 & 0 & 0.5^2 & 0 & 0 \\ 0 & 0 & 0 & 0.5^2 & 0 \\ 0 & 0 & 0 & 0 & 0.5^2 \end{pmatrix}$$

- The estimated quantities from the state vector can be used to estimate the particle's momentum

$$\begin{cases} p_x = p_T \tan\lambda \\ p_y = p_T \sin\phi \\ p_z = p_T \cos\phi \end{cases}$$

$$p_T(\mathrm{GeV}/c) = 0.3 \times B(T) \times r(m)$$

# KALMAN FILTER APPLICATION: PREDICTION AND MEASUREMENT

- From the equation of motion we obtain the prediction function for our state vector.

$$\hat{x}_k^- = f\begin{pmatrix} \hat{y}_{k-1}^+ \\ \hat{z}_{k-1}^+ \\ 1/\hat{r}_{k-1}^+ \\ \hat{\phi}_{k-1}^+ \\ \hat{\lambda}_{k-1}^+ \end{pmatrix} = \begin{pmatrix} \hat{y}_{k-1}^+ + \mathrm{dx_k} \times \cot\hat{\lambda}_{k-1} \times \sin\hat{\phi}_{k-1}^+ \\ \hat{z}_{k-1}^+ + \mathrm{dx_k} \times \cot\hat{\lambda}_{k-1} \times \cos\hat{\phi}_{k-1}^+ \\ 1/\hat{r}_{k-1}^+ \\ \hat{\phi}_{k-1}^+ + \mathrm{dx_k} \times \cot\hat{\lambda}_{k-1} \times \sin\hat{\phi}_{k-1}^+ \\ \hat{\lambda}_{k-1}^+ \end{pmatrix}$$

Note: the prediction model does not account for dE/dx energy loss

- The only measured quantities in our case are $y$ and $z$, and are set at the center of the TPC cluster correspondent to the present step.

$$z_k = \begin{pmatrix} y_k^h \\ z_k^h \end{pmatrix}$$

# KALMAN FILTER APPLICATION: KINEMATIC FIT

- Each algorithm step corresponds to a TPC cluster. The x coordinate is treated as independent and used to identify the step width $dx$. The step width is determined for each algorithm step, so that it minimizes:

$$\Delta = \left[\frac{\left(x_k^h - \hat{x}_k^-\right)}{\sigma_x}\right]^2 + \left[\frac{\left(y_k^h - \hat{y}_k^-\right)}{\sigma_{yz}}\right]^2 + \left[\frac{\left(z_k^h - \hat{z}_k^-\right)}{\sigma_{yz}}\right]^2$$

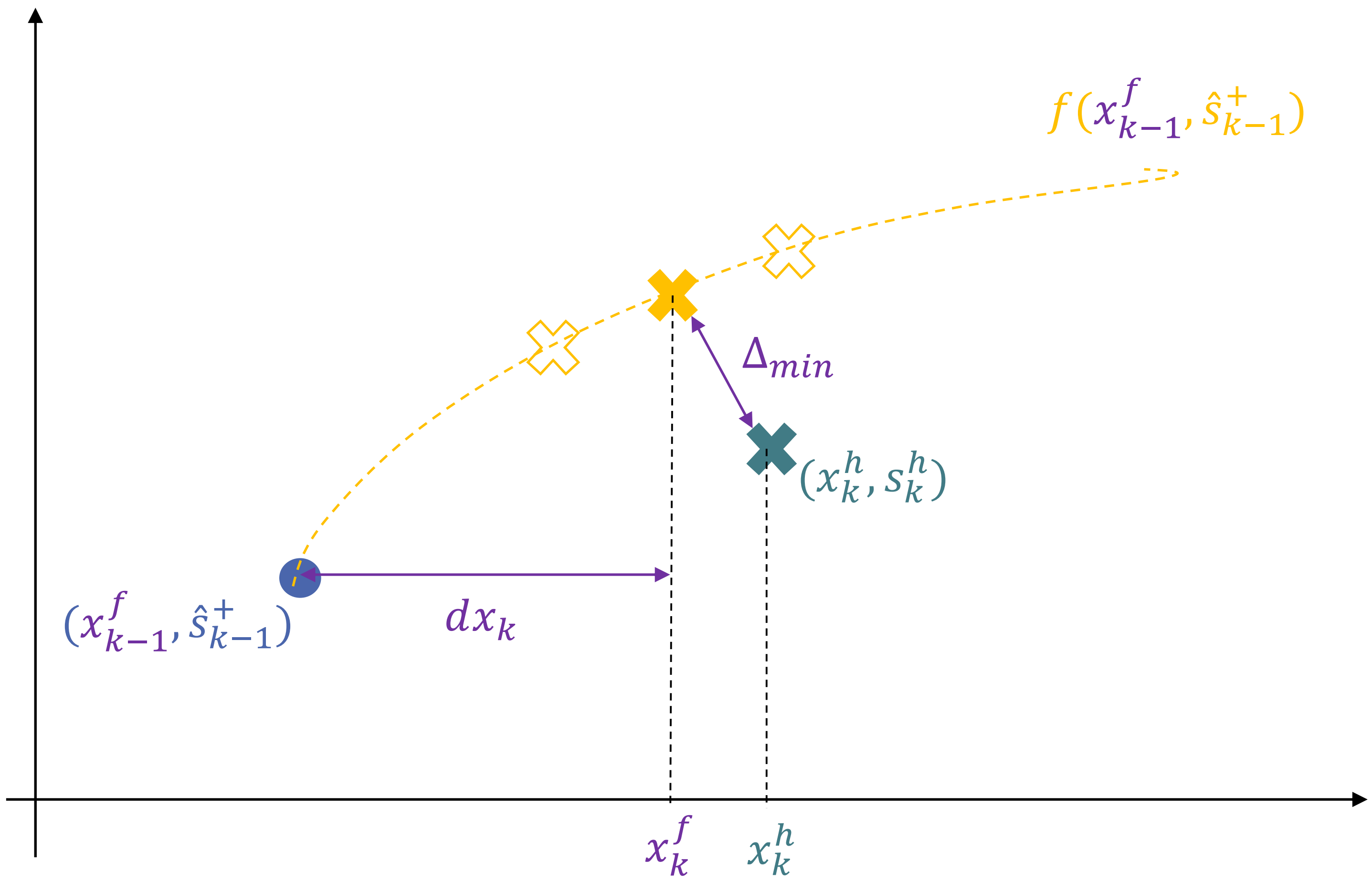
$$= \left[\frac{(dx_k)}{\sigma_x}\right]^2 + \left[\frac{\left(y_k^h - \hat{y}_{k-1}^+ + dx_k \times \cot\hat{\lambda}_{k-1} \times \sin\hat{\phi}_{k-1}^+\right)}{\sigma_{yz}}\right]^2 + \left[\frac{\left(z_k^h - \hat{y}_{k-1}^+ + dx_k \times \cot\hat{\lambda}_{k-1} \times \sin\hat{\phi}_{k-1}^+\right)}{\sigma_{yz}}\right]^2$$

$\sigma_x = 0.5\text{cm}$ and $\sigma_{yz} = 1\text{cm}$ are arbitrary values, which do not coincide with the TPC values in the R matrix

- We call this procedure kinematic fit (note that this is not part of the Kalman Filter but runs alongside it):

$$\frac{d\Delta}{d(dx_k)} = 0 \quad \Rightarrow \quad \boxed{dx_k = \frac{\left(\frac{\cot\hat{\lambda}_{k-1}^+}{\sigma_{yz}^2}\left(\left(y_k^h - \hat{y}_{k-1}^+\right)\sin\hat{\phi}_{k-1}^+ + \left(z_k^h - \hat{z}_{k-1}^+\right)\cos\hat{\phi}_{k-1}^+\right)\right) + \frac{\left(x_k^h - \hat{x}_{k-1}^+\right)}{\sigma_x^2}}{\cot^2\hat{\lambda}_{k-1}^+/\sigma_{yz}^2 + 1/\sigma_x^2}}$$

# KALMAN FILTER: KINEMATIC FIT




$$x^f_k = x^f_{k-1} + dx_k$$

$dx_k$ is taken such that the distance between the a priori prediction $f(x^f_{k-1}, \hat{s}^+_{k-1})$ and the measurement $(x^h_k, s^h_k)$ is minimized

# KALMAN FILTER APPLICATION: COVARIANCE MATRIX PREDICTION

- First step to make the prediction for the covariance matrix is to calculate the Jacobian

$$F_{k-1} = \frac{\partial f(\hat{x}_{k-1}^{+})}{\partial \hat{x}_{k-1}^{+}} = \begin{bmatrix} 1 & 0 & 0 & dx_k \cot \hat{\lambda}_{k-1}^{+} \cos \hat{\phi}_{k-1}^{+} & dx_k \sin \hat{\phi}_{k-1}^{+}(-1 - \cot^2 \hat{\lambda}_{k-1}^{+}) \\ 0 & 1 & 0 & -dx_k \cot \hat{\lambda}_{k-1}^{+} \sin \hat{\phi}_{k-1}^{+} & dx_k \cos \hat{\phi}_{k-1}^{+}(-1 - \cot^2 \hat{\lambda}_{k-1}^{+}) \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & dx_k \cot \hat{\lambda}_{k-1}^{+} & 1 & dx_k / \hat{r}_{k-1}^{+} (-1 - \cot^2 \hat{\lambda}_{k-1}^{+}) \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

- The step uncertainty matrix is also needed

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{\Delta 1/r} & 0 & 0 \\ 0 & 0 & 0 & \sigma_{\Delta\phi} & 0 \\ 0 & 0 & 0 & 0 & \sigma_{\Delta\lambda} \end{bmatrix}$$

- The prediction is then:

$$P_k^{-} = F_{k-1} P_{k-1} F_{k-1}^{T} + Q$$

# KALMAN FILTER APPLICATION: EVALUATE THE RESIDUAL

- We now evaluate the residual and Kalman Gain

RESIDUAL

$$\tilde{y}_k = z_k - H(\hat{x}_k^-) = \begin{pmatrix} y_k^h - \hat{y}_k^- \\ z_k^h - \hat{z}_k^- \end{pmatrix}$$

KALMAN GAIN

$$K_k = P_k^- H (R + H P_k^- H^T)^{-1}$$

With:

$$R = \begin{pmatrix} \sigma_{yz}^2 & 0 \\ 0 & \sigma_{yz}^2 \end{pmatrix}$$

MEASUREMENT NOISE COVARIANCE

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

CONVERSION MATRIX

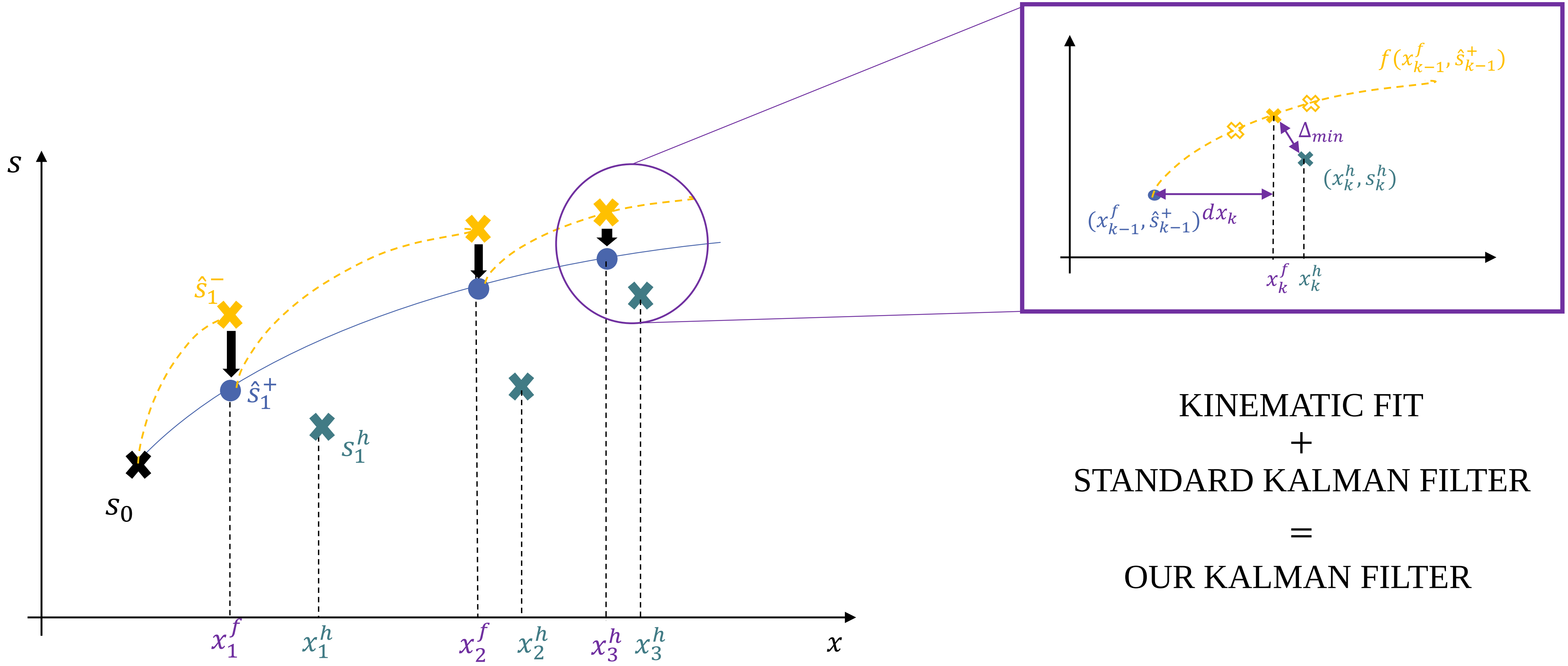
Note: The uncertainties in R are fixed, before the Kalman filter is applied, as external parameters: R is not updated.

# KALMAN FILTER APPLICATION: PREDICTION UPDATE

- We are now finally able to update our estimates using both the a priori prediction and the measurement

| | |
|---|---|
| STATE VECTOR | $\hat{x}_k^+ = \hat{x}_k^- + K_k \tilde{y}_k$ |
| COVARIANCE MATRIX | $P_k^+ = (1 - H_k K_k) P_k^-$ |

# OUR KALMAN FILTER: VISUALIZATION




KINEMATIC FIT

+

STANDARD KALMAN FILTER

=

OUR KALMAN FILTER

# KALMAN FILTER APPLICATION: $\chi^2$

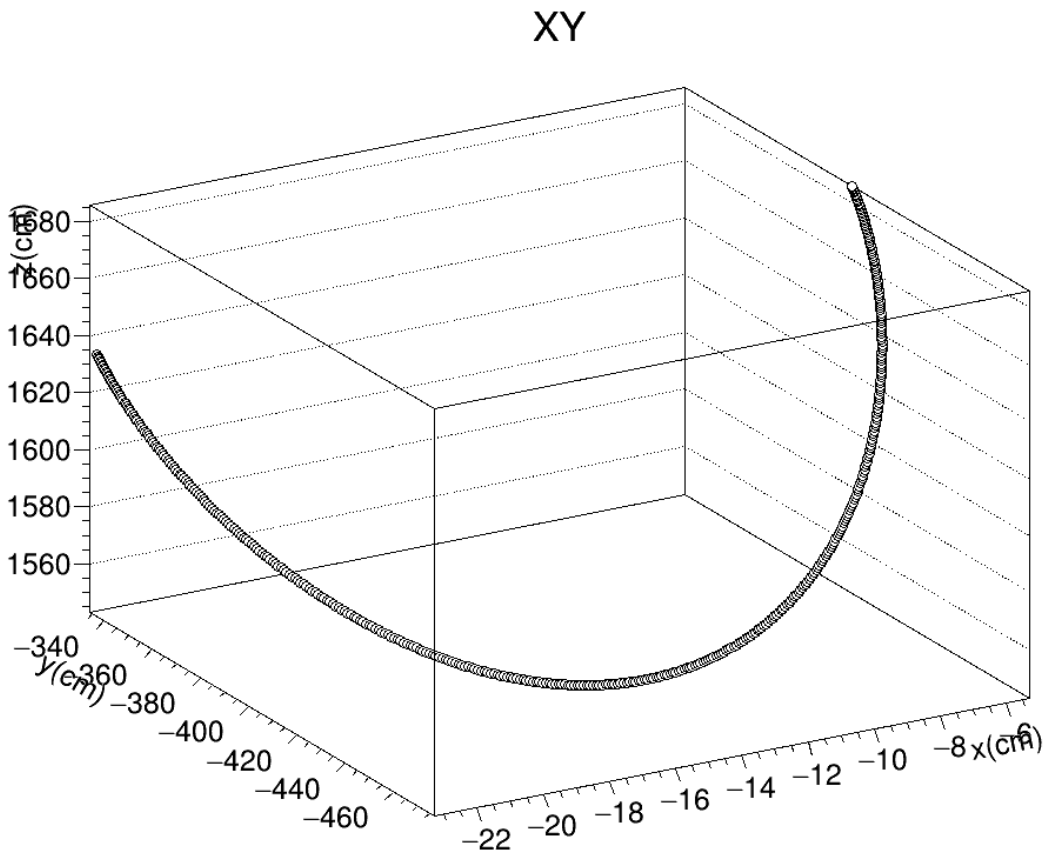
- The Kalman filter algorithm is applied to the track candidate both ways (i.e. From first TPC cluster to last and vice versa) and each time a $\chi^2$ value is calculated

$$\boxed{\chi^2 = \frac{\sum |\tilde{y}_k|^2}{\sum \left|y_k^{typ}\right|^2}}$$

- Where $y_k^{typ}$ is the residual typical value which depends on the particle trajectory position in the detector and the tracking planes (fixed values evaluated a priori)

# TOY MONTECARLO

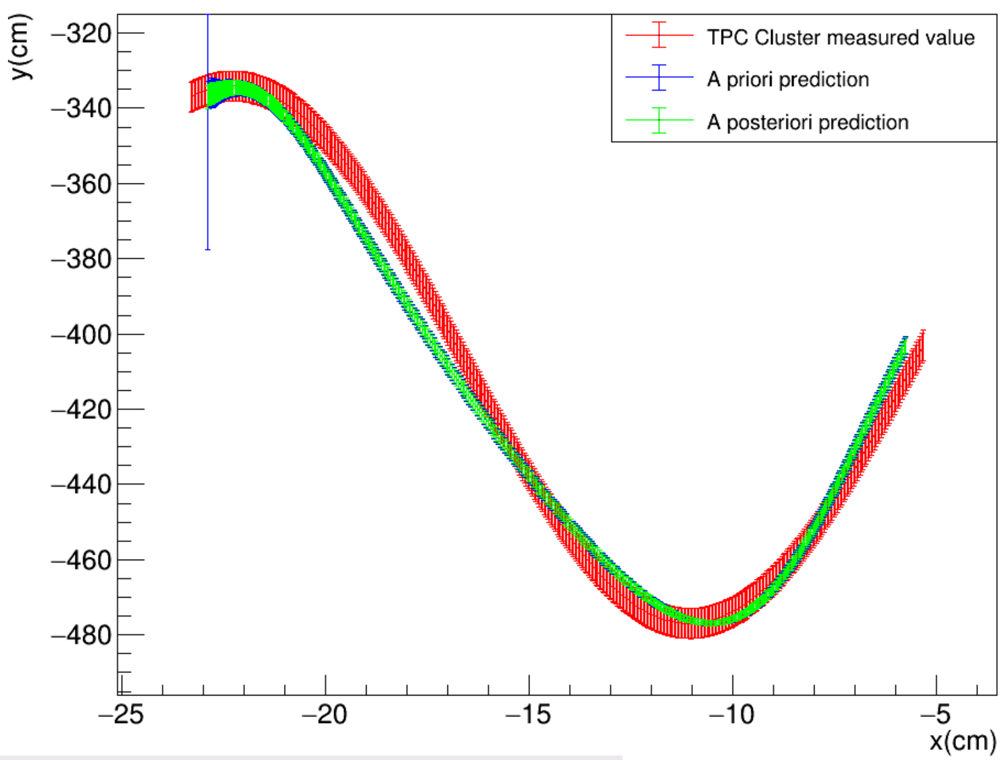
- Applied Kalman Filter to ideal measurements following perfectly an helix with initial coordinates: $x_i^T = (y_i \quad z_i \quad 1/r_i \quad \phi_i \quad \lambda_i) = (y_1^h \quad z_1^h \quad 0.014\, cm^{-1} \quad 6 \text{ rad} \quad -0.05 \text{ rad})$ and the free parameter $x_i = x_1^h$ and the step $dx = 0.04$ cm



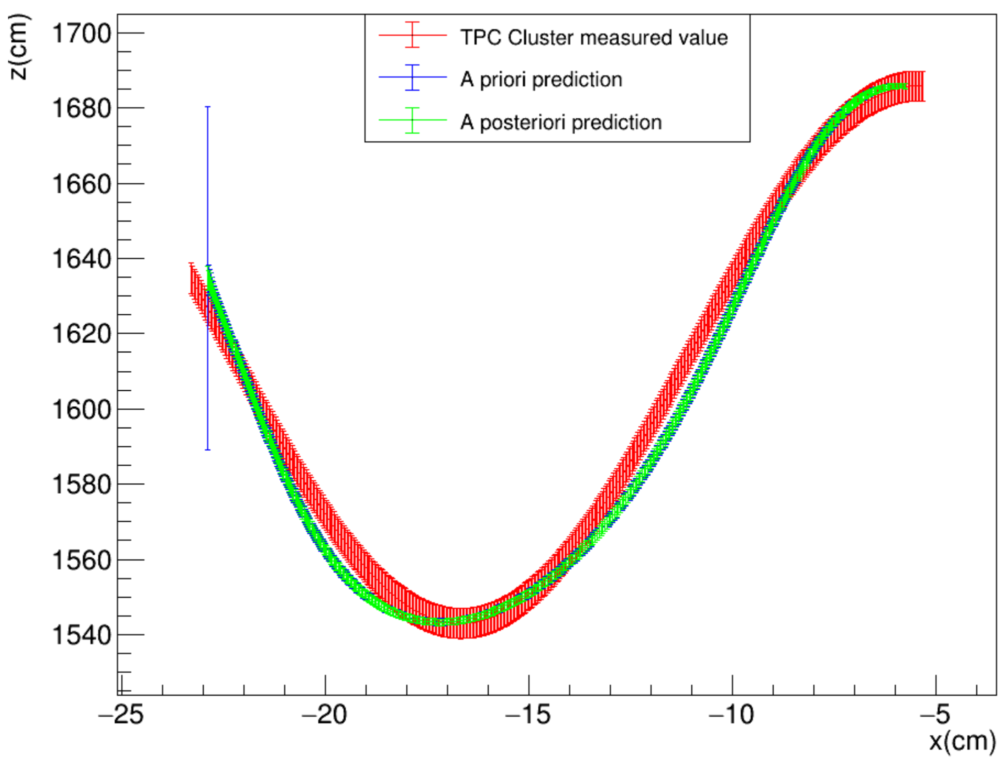
# TOY MONTECARLO

- Plots describing the evolution of the estimate of the measured quantities $(y, z)$ as a function of the free parameter $x$ for the perfect helix. Note that the fake TPC points have on the x coordinate the fake measured value, while for the estimates we use the estimated x (i.e. $x_k = x_0 + k \times dx_k$)

Y




Z



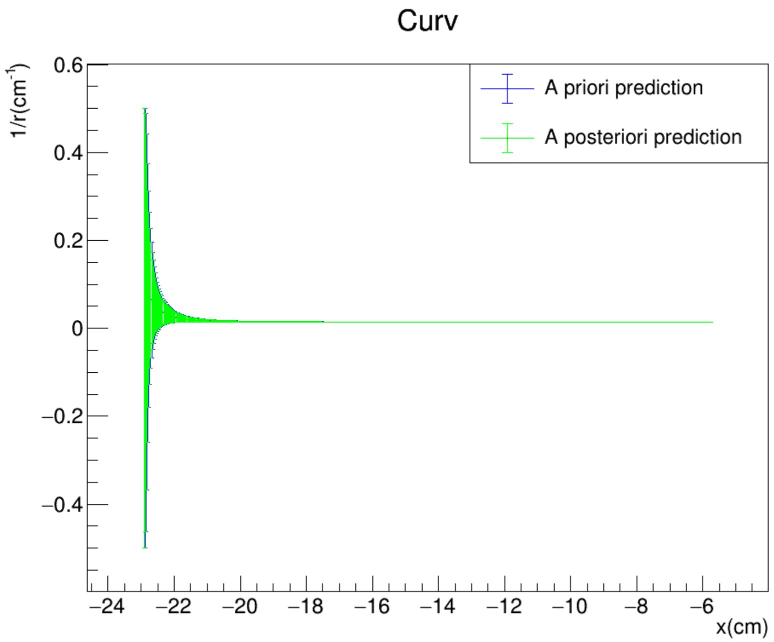

Note: the ‘TPC Cluster measured values’ are from a Toy Montecarlo, so the red error bars are just dummy values

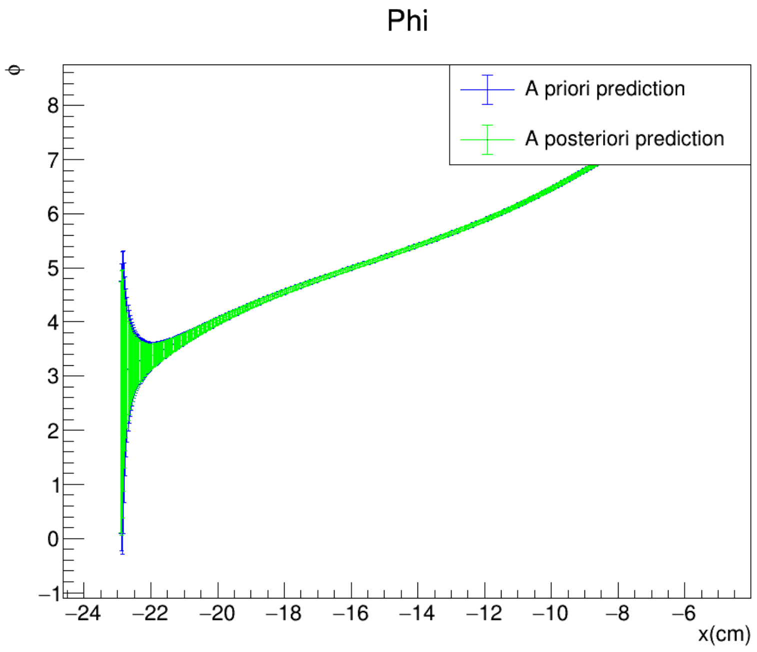
FITTER PROPAGATION DIRECTION

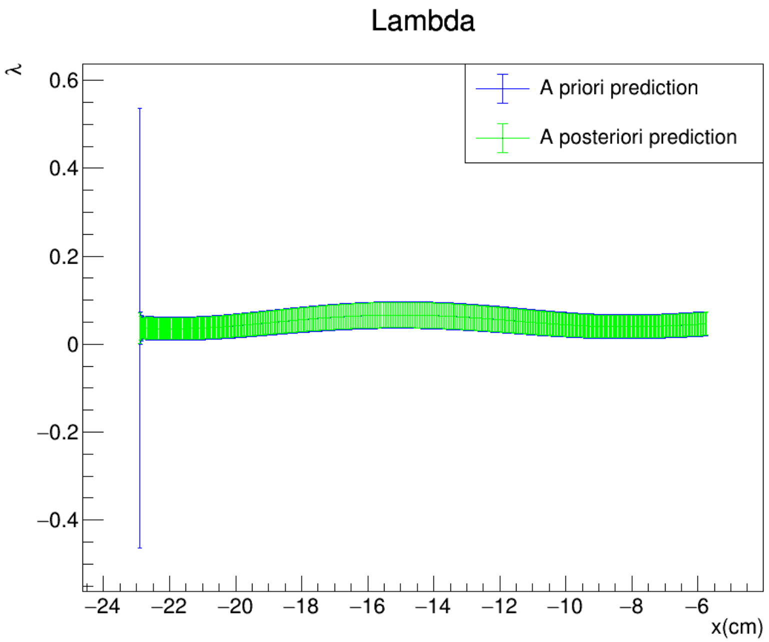
$(\sigma_x, \sigma_{yz}) = (0.5cm, 1cm)$

# TOY MONTECARLO

- Plots describing the evolution of the estimate of the non measured quantities $(\frac{1}{r}, \phi, \lambda)$ as a function of the free parameter $x$ for the perfect helix



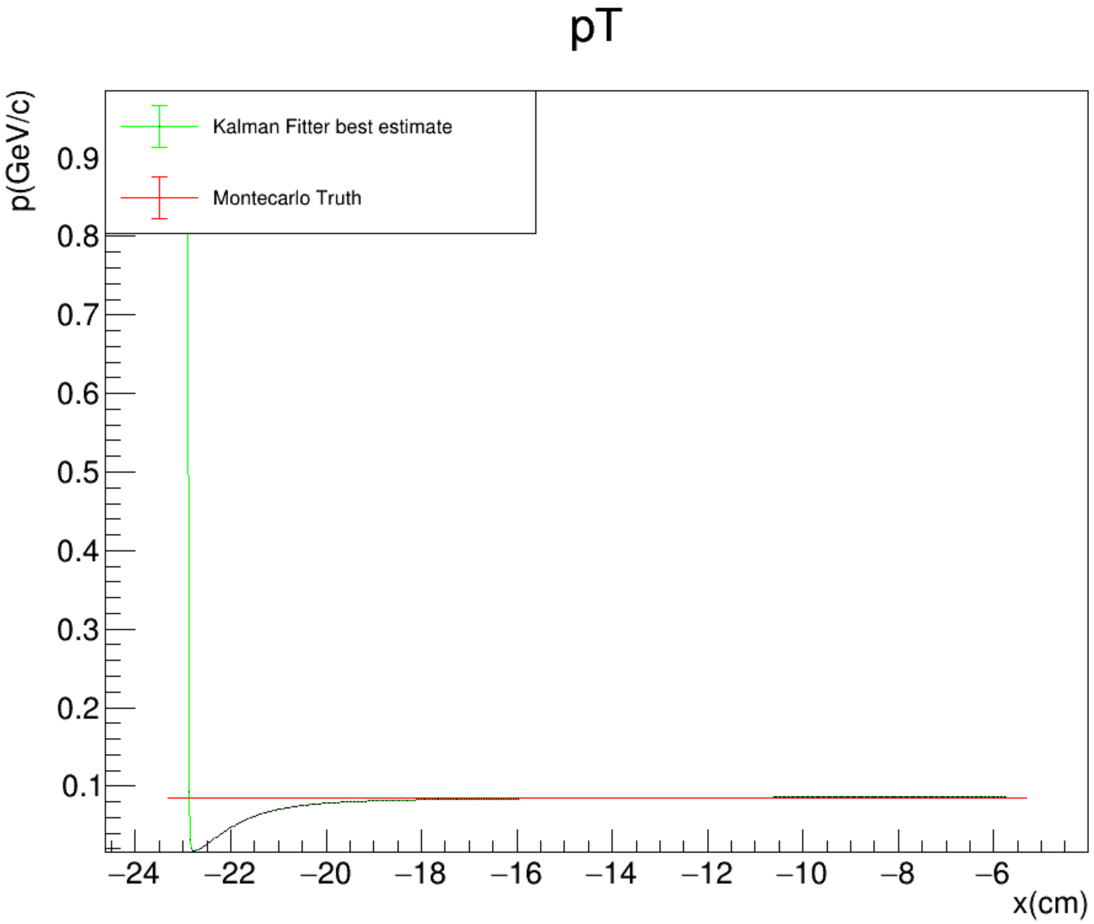



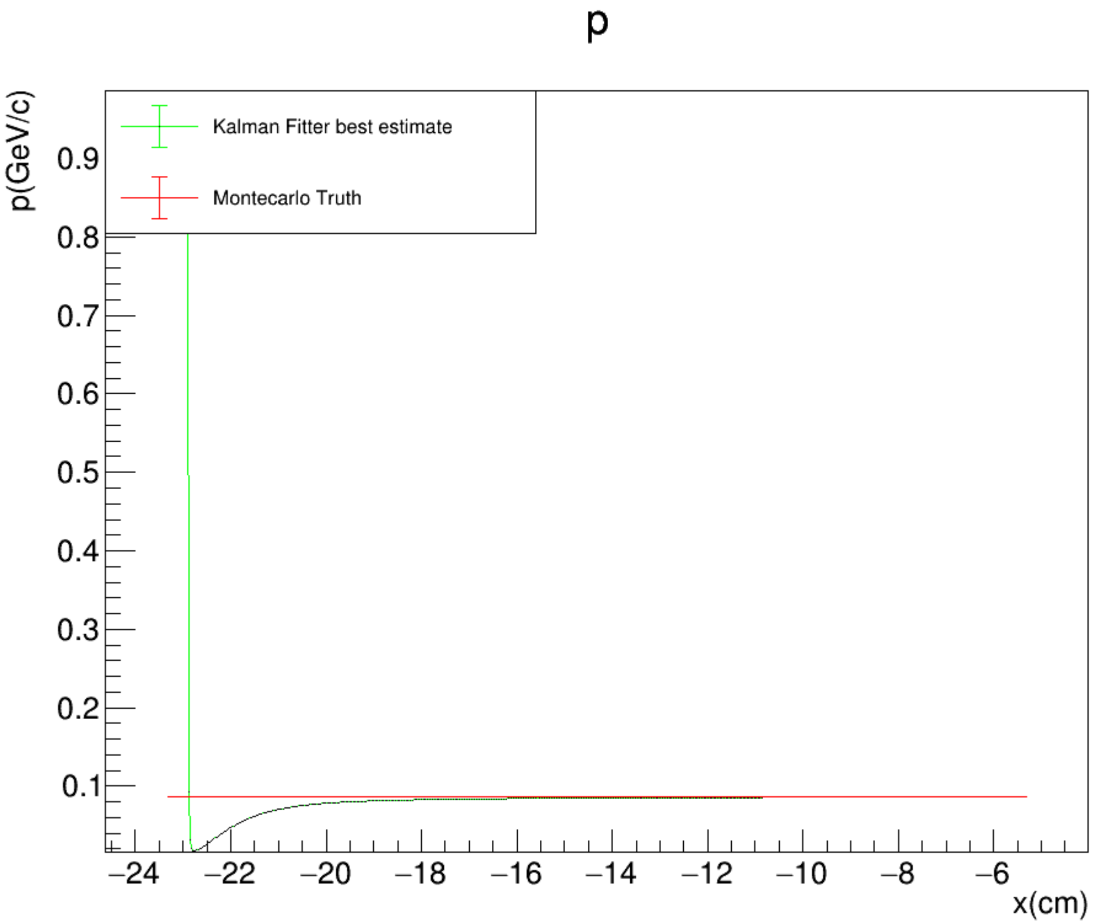

$(\sigma_x, \sigma_{yz}) = (0.5cm, 1cm)$

FITTER PROPAGATION DIRECTION

# TOY MONTECARLO

- Plots comparing the reconstructed tansverse momentum $p_T$ and the reconstructed total momentum $p_T$ from the Kalman fitter algorithm to the MC truth as a function of the free parameter $x$



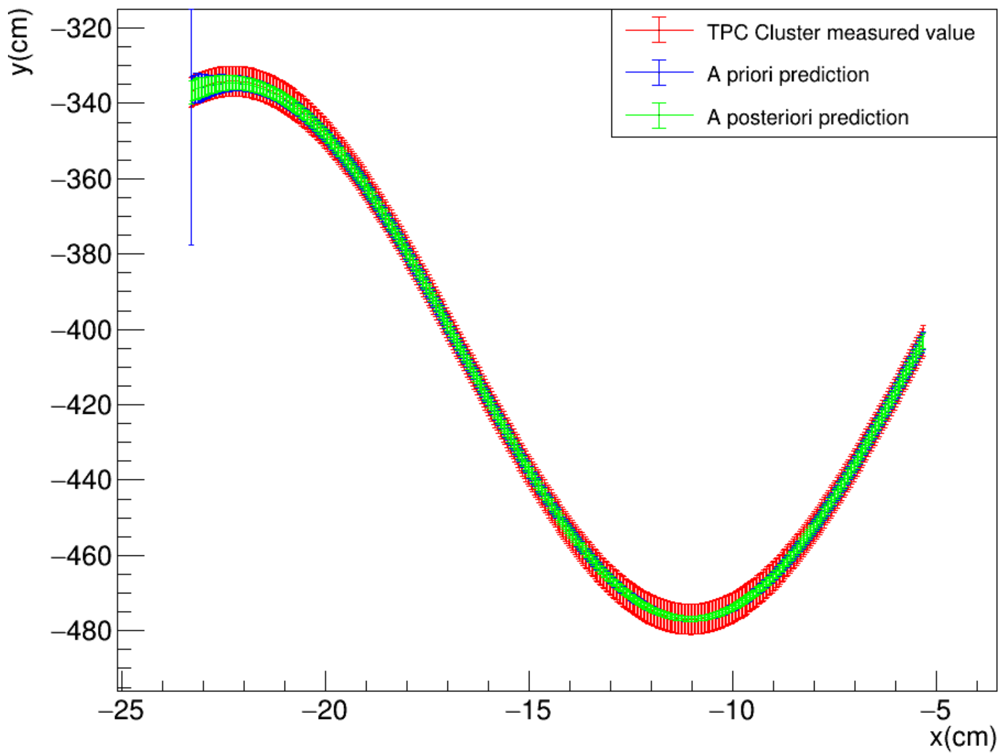

$$(\sigma_x, \sigma_{yz}) = (0.5cm, 1cm)$$

FORWARD FIT

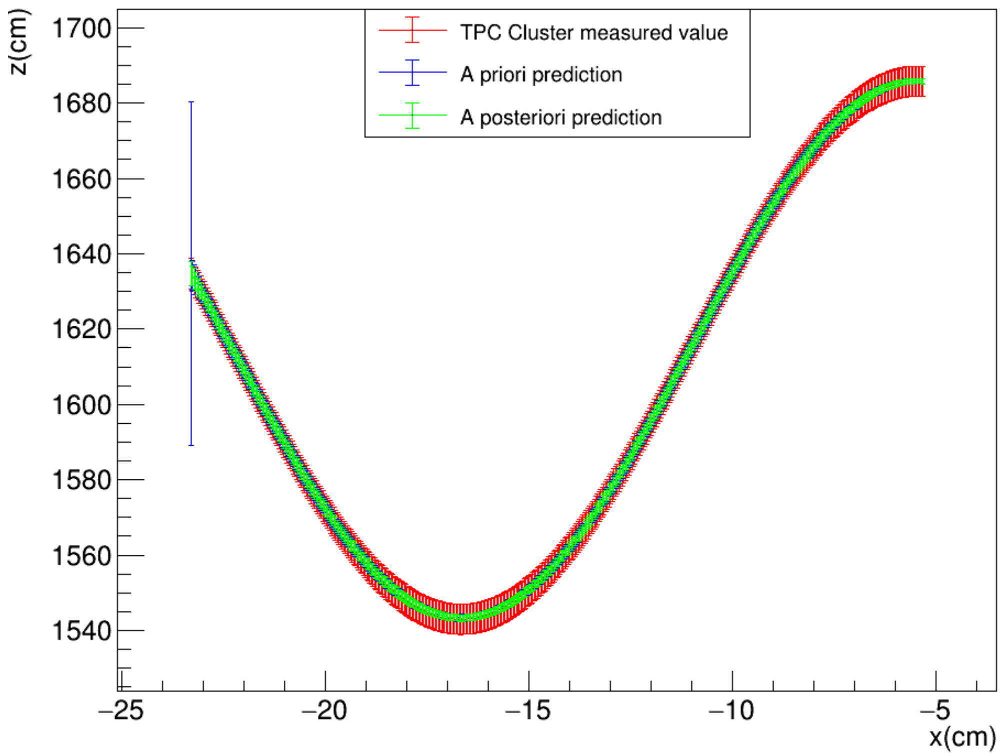
# TOY MONTECARLO

- Plots describing the evolution of the estimate of the measured quantities $(y, z)$ as a function of the free parameter $x$ for the perfect helix. This time both the fake TPC points and the predictions have on the x coordinate the fake measured value: the predictions for the state vector are working fine, while the kinematic fit prediction for the free parameter is failing

Y




Z




$(\sigma_x, \sigma_{yz}) = (0.5cm, 1cm)$

FITTER PROPAGATION DIRECTION

# TOY MONTECARLO: UNDERSTANDING STEP DETERMINATION

- The $dx_k$ update formula can be devided into two parts: one that depends on the evolution of the $x$ free parameter, and one on the measured quantities $y$ and $z$

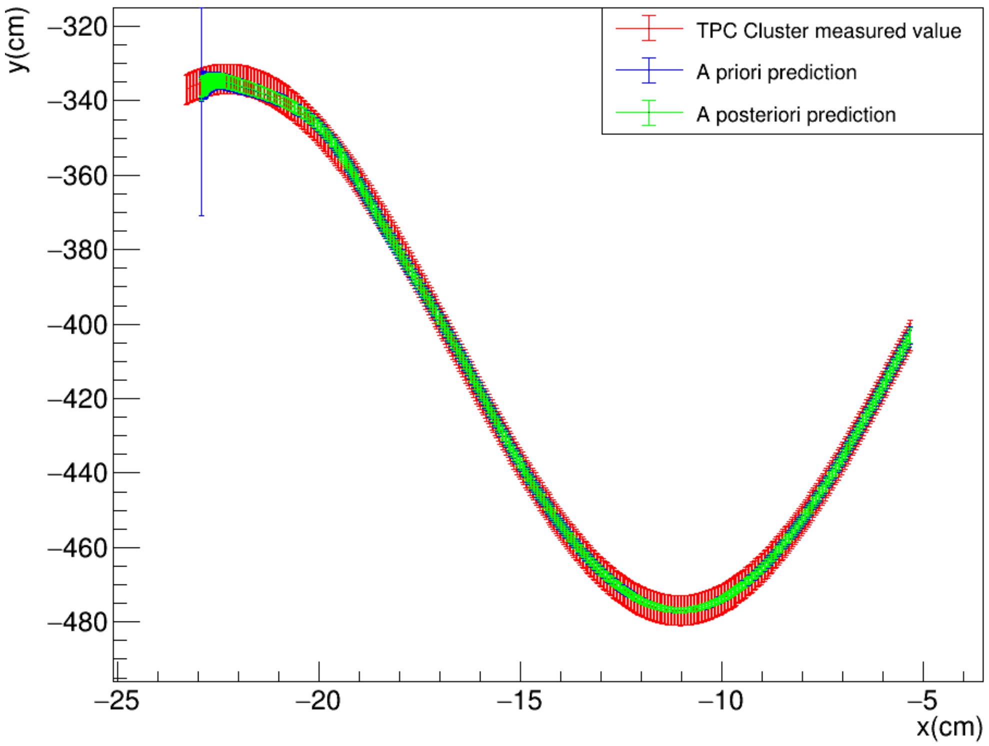
$$dx_k = \frac{\frac{\cot \hat{\lambda}^+_{k-1}}{\sigma^2_{yz}}\left(\left(y^h_k - \hat{y}^+_{k-1}\right)\sin\hat{\phi}^+_{k-1} + \left(z^h_k - \hat{z}^+_{k-1}\right)\cos\hat{\phi}^+_{k-1}\right)}{\cot^2 \hat{\lambda}^+_{k-1} \big/ \sigma^2_{yz} + 1 \big/ \sigma^2_x} + \frac{\frac{\left(x^h_k - \hat{x}^+_{k-1}\right)}{\sigma^2_x}}{\cot^2 \hat{\lambda}^+_{k-1} \big/ \sigma^2_{yz} + 1 \big/ \sigma^2_x} = dx_{yz} + dx_x$$

- The values of $\sigma_x$ and $\sigma_{yz}$ determine how much the update is influenced by the measured values of $x$ or of $y$ and $z$ respectivelly
- With the current sigma values ($\sigma_x = 0.5cm$ and $\sigma_{yz} = 1cm$ ) $dx_{yz}$ completely dominates, being often 2 or 3 orders of magnitude larger than $dx_x$: this gives us fairly accurate predictions for $y$ and $z$, but completely wrong ones for $x$, becuase the fit can never recover from a bad initial estimate for the free parameter

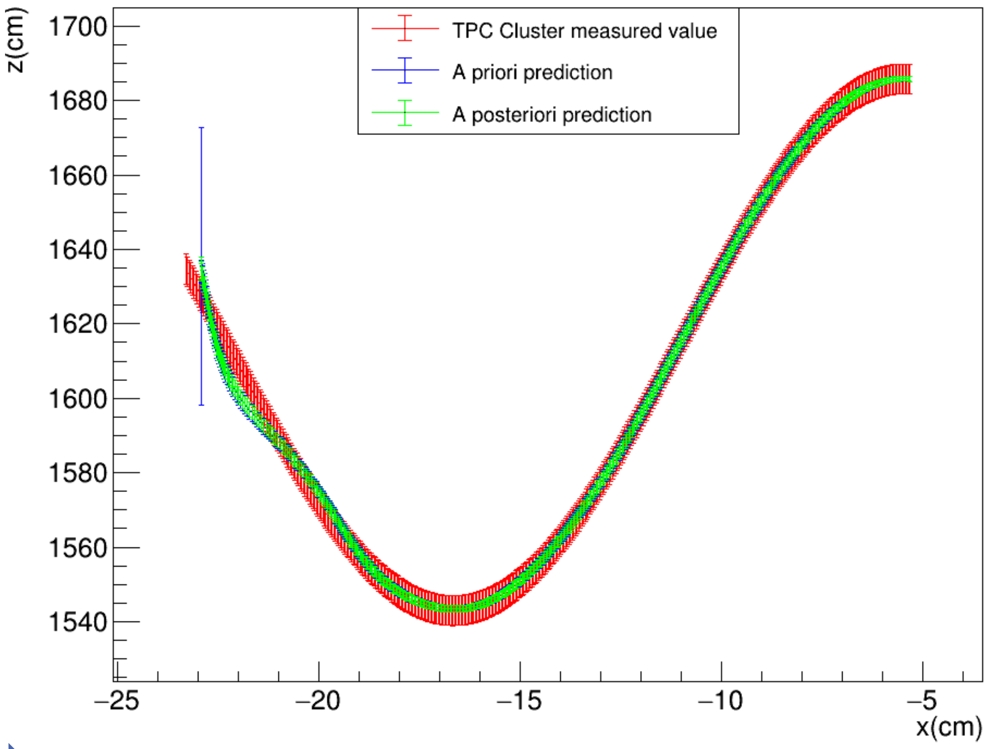
# TOY MONTECARLO: UNDERSTANDING STEP DETERMINATION

- In order for the x parameter to have more weight in the update of $dx_k$, changed the value of $\sigma_{yz}$ from 1cm to 4cm, leaving $\sigma_x = 0.5cm$
- The fit initially concentrates on fixing the $x$ prediction until that becomes accurate, and then it focuses on yz, reaching similar levels of precision by the end

Y




Z

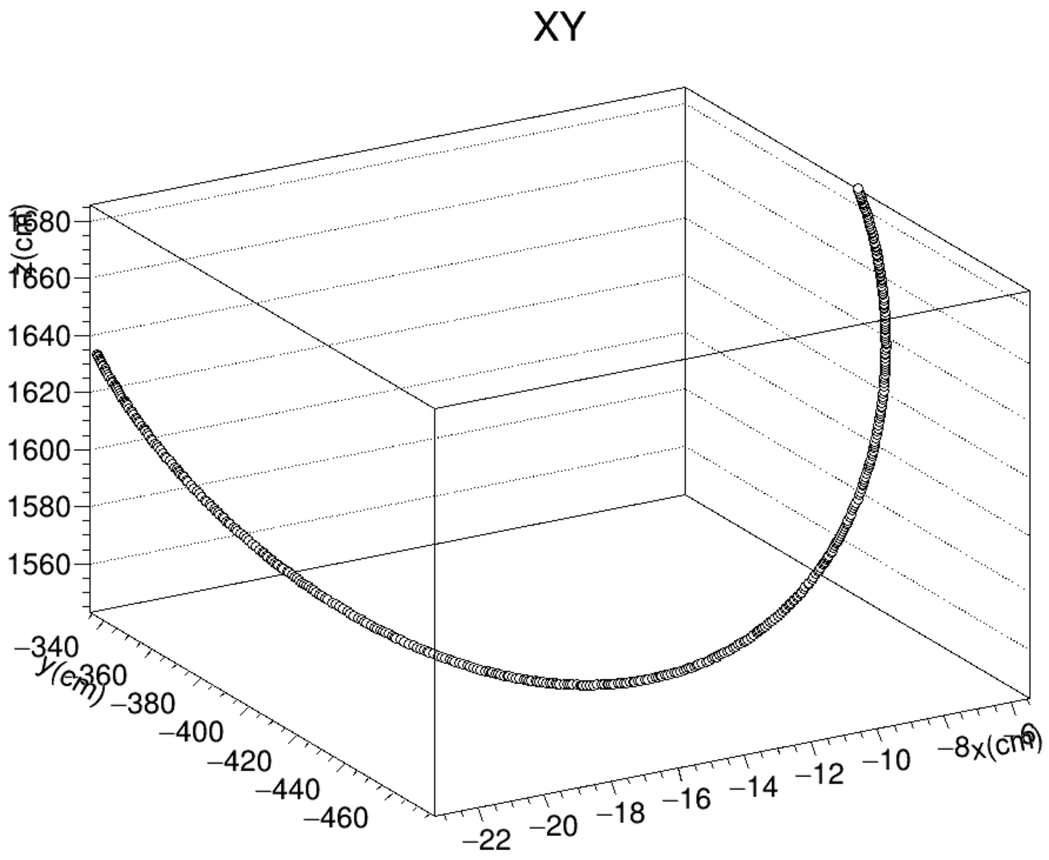



$(\sigma_x, \sigma_{yz}) = (0.5cm, 4cm)$

FITTER PROPAGATION DIRECTION

# RANDOMIZED X STEP

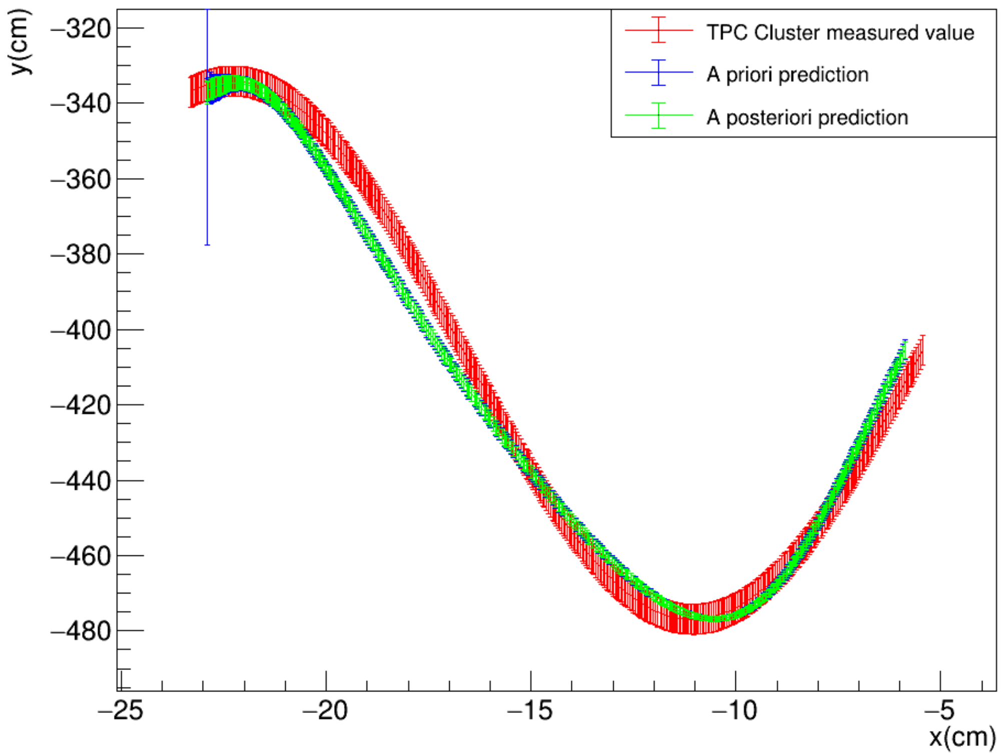
- Now we try applying Kalman Filter to ideal measurements following perfectly an helix with initial coordinates: $x_i^T = (y_i \quad z_i \quad 1/r_i \quad \phi_i \quad \lambda_i) = (y_1^h \quad z_1^h \quad 0.014\,cm^{-1} \quad 6\text{ rad} \quad -0.05\text{ rad})$ and the free parameter $x_i = x_1^h$ but with a randomized step $dx$ uniformely distributed between 0.02cm and 0.06cm



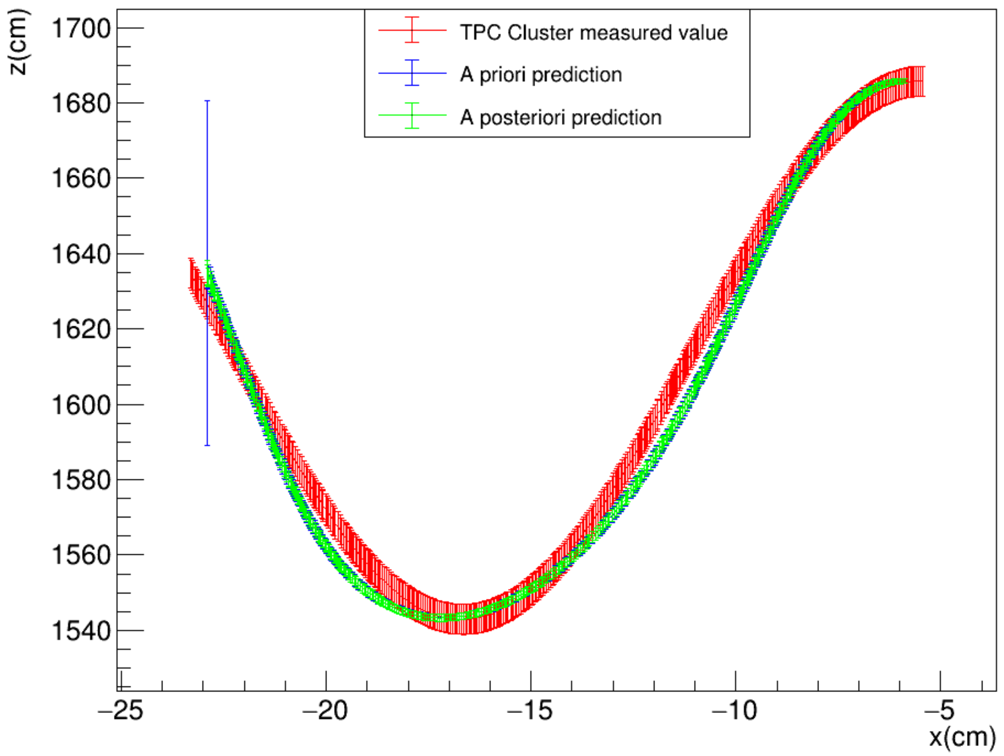
# RANDOMIZED X STEP

- Plots describing the evolution of the estimate of the measured quantities $(y, z)$ as a function of the free parameter $x$ for the perfect helix. Note that the the fake TPC points have on the x coordinate the  fake measured value, while for the estimates we use the estimated x (i.e. $x_k = x_0 + k \times dx_k$)

Y




Z

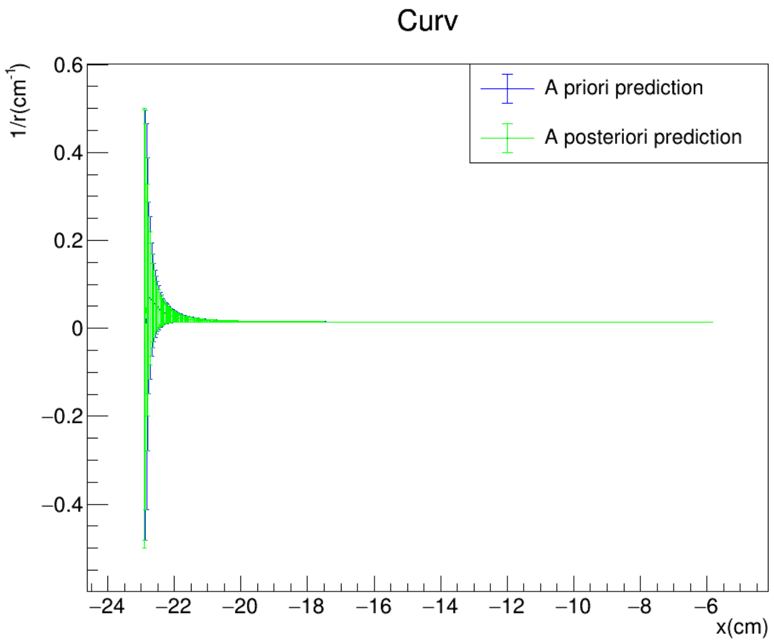


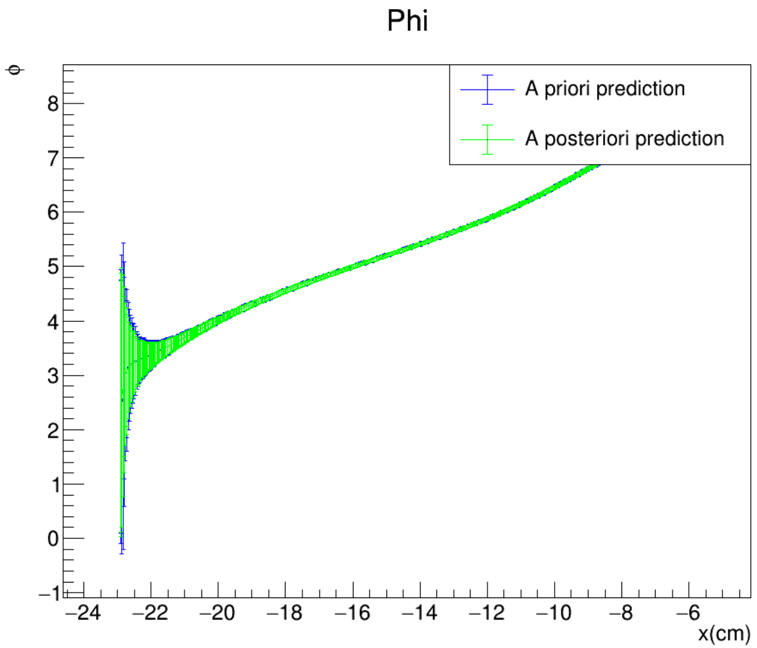

$(\sigma_x, \sigma_{yz}) = (0.5cm, 1cm)$

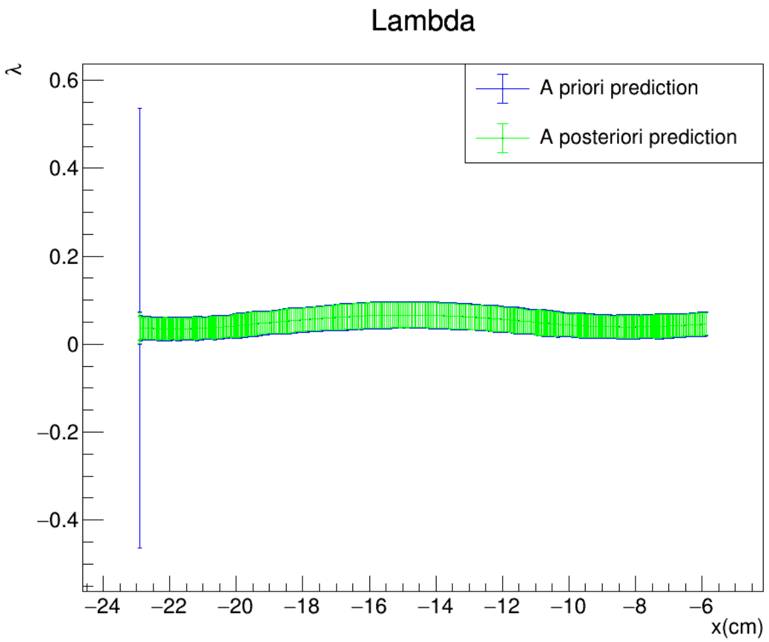
FITTER PROPAGATION DIRECTION

# RANDOMIZED X STEP

- Plots describing the evolution of the estimate of the non measured quantities $(\frac{1}{r}, \phi, \lambda)$ as a function of the free parameter $x$ for the perfect helix



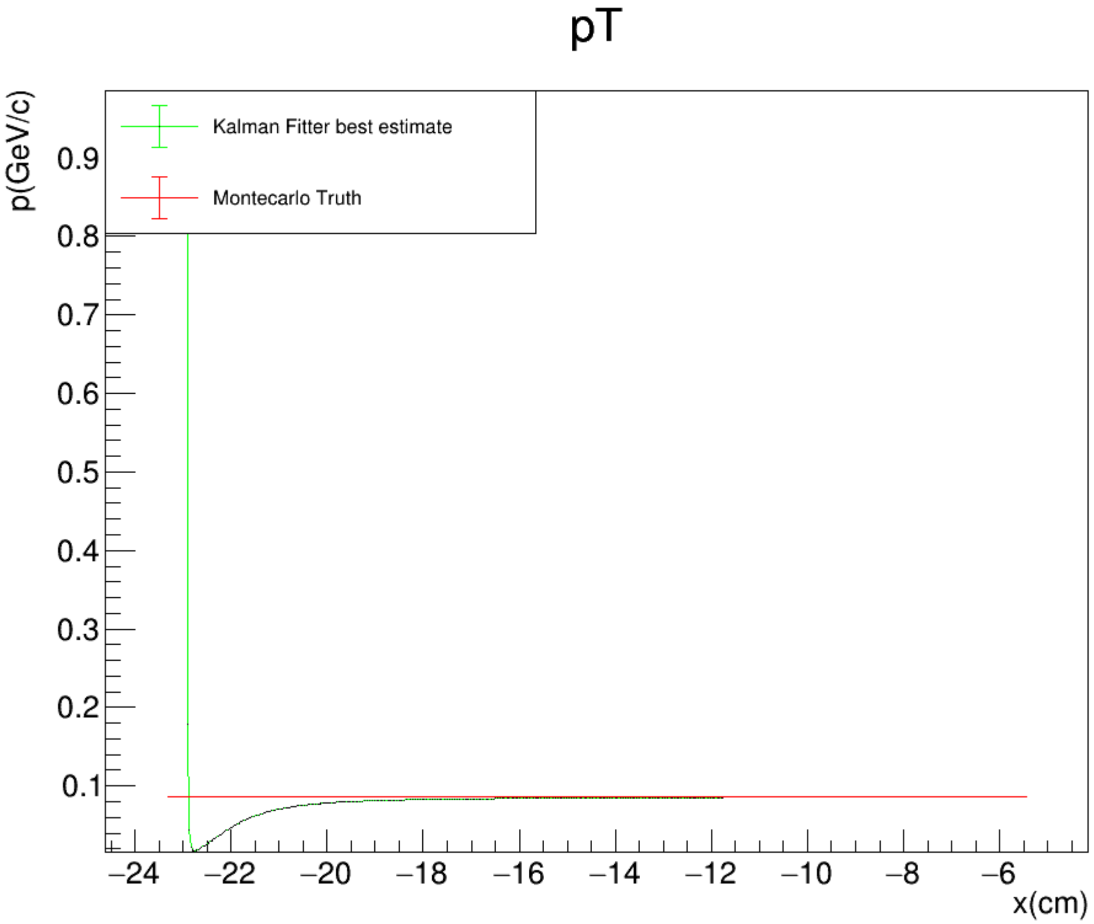



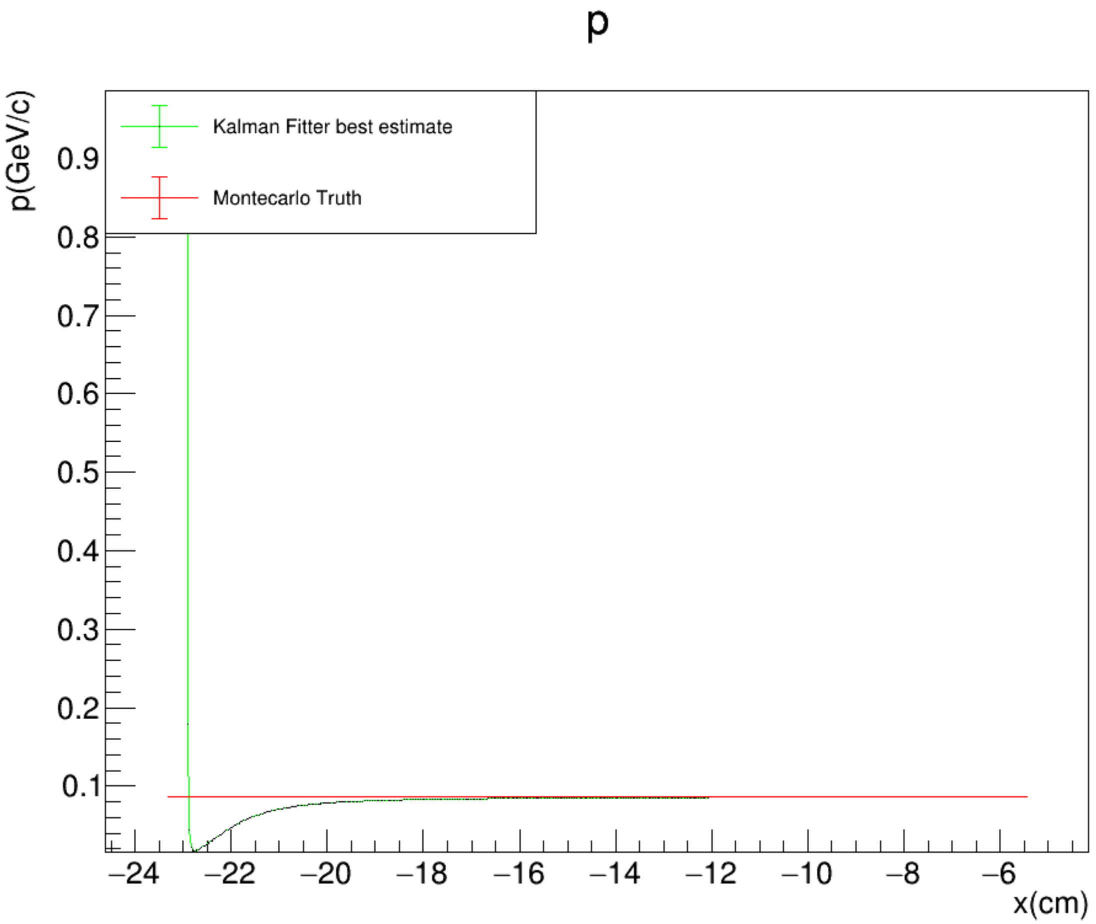

$(\sigma_x, \sigma_{yz}) = (0.5cm, 1cm)$

FITTER PROPAGATION DIRECTION

# RANDOMIZED X STEP

- Plots comparing the reconstructed tansverse momentum $p_T$ and the reconstructed total momentum $p_T$ from the Kalman fitter algorithm to the MC truth as a function of the free parameter $x$



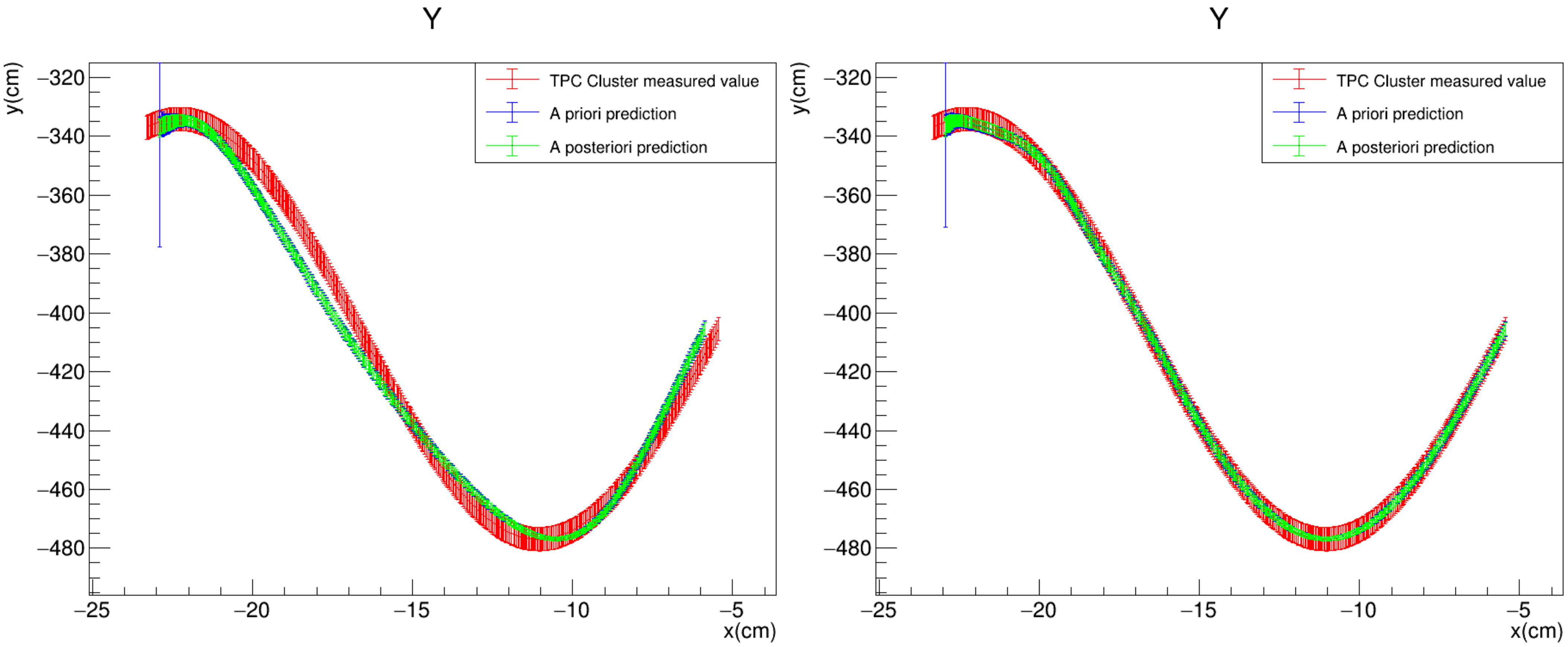




$$\left(\sigma_x, \sigma_{yz}\right) = (0.5cm, 1cm)$$

FORWARD FIT

# RANDOMIZED X STEP

- Reapplying the fit with the new $\sigma_{yz} = 4$cm we see that the 3D predictions are more in line with the Montecarlo truth, past the first few steps



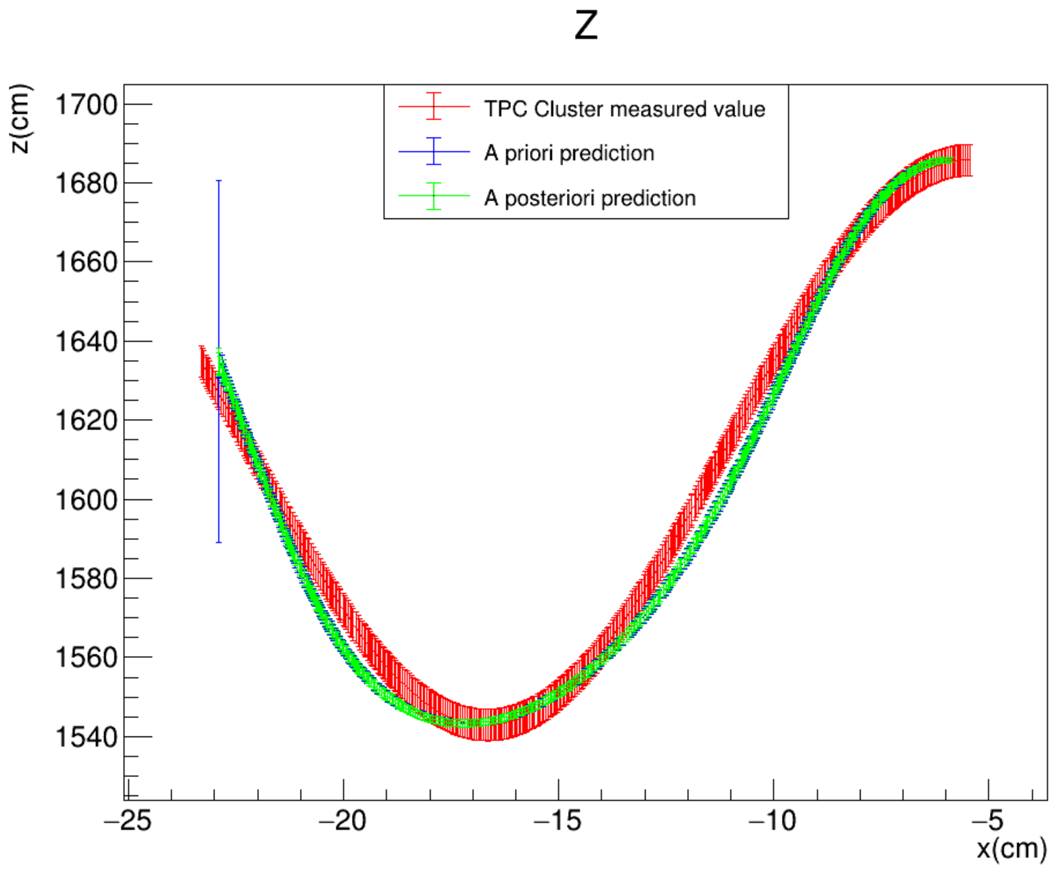

$(\sigma_x, \sigma_{yz}) = (0.5cm, 1cm)$

$(\sigma_x, \sigma_{yz}) = (0.5cm, 4cm)$

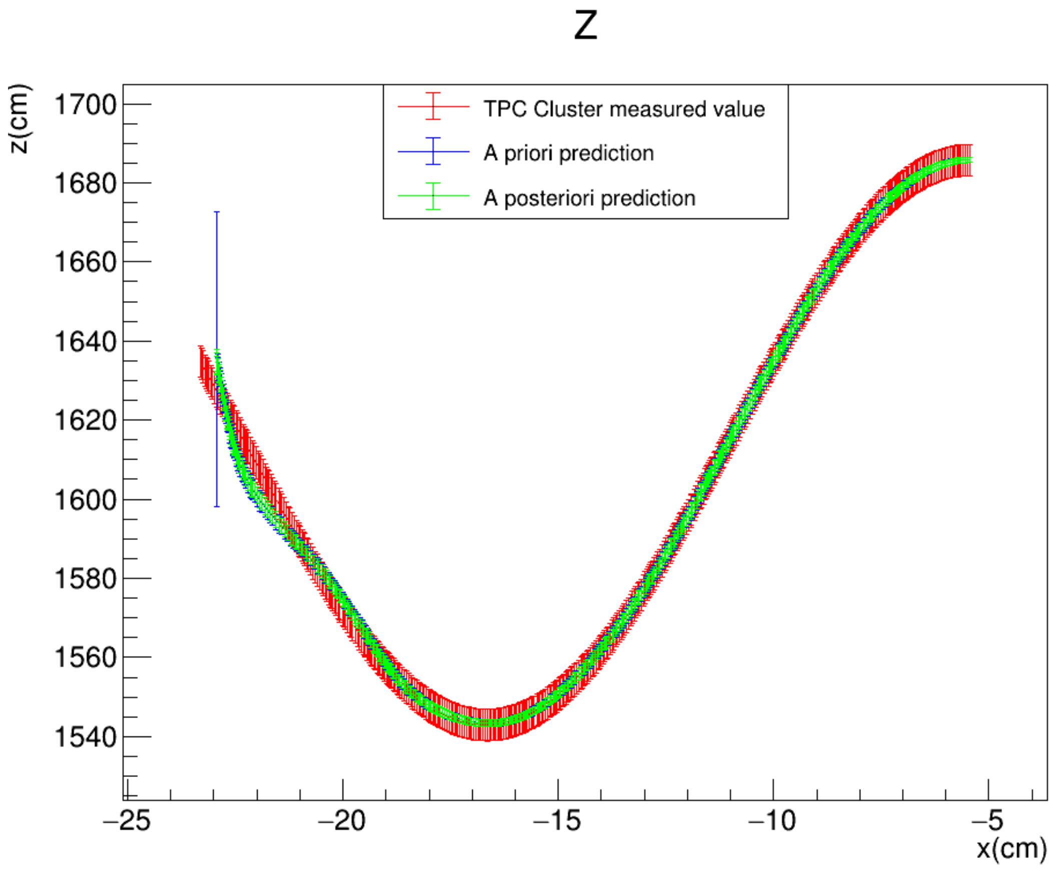
FITTER PROPAGATION DIRECTION

# RANDOMIZED X STEP

- Reapplying the fit with the new $\sigma_{yz} = 4\text{cm}$ we see that the 3D predictions are more in line with the Montecarlo truth, past the first few steps




$(\sigma_x, \sigma_{yz}) = (0.5cm, 1cm)$

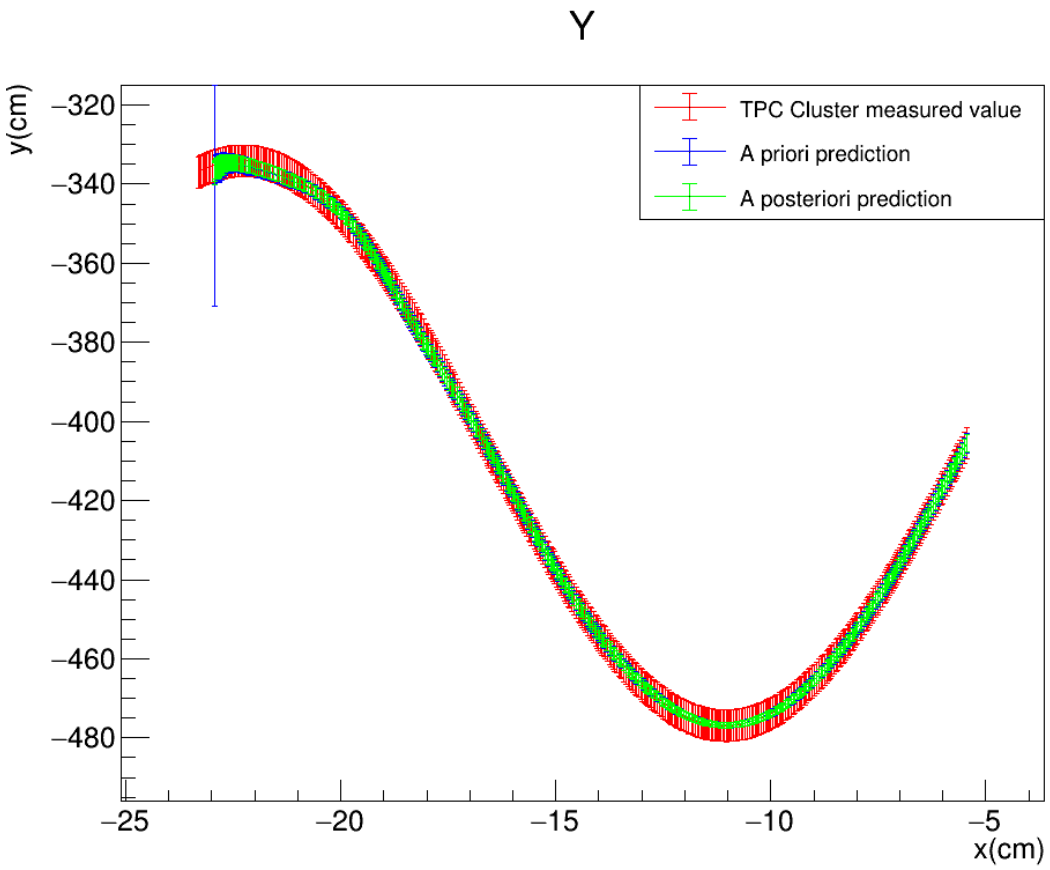



$(\sigma_x, \sigma_{yz}) = (0.5cm, 4cm)$

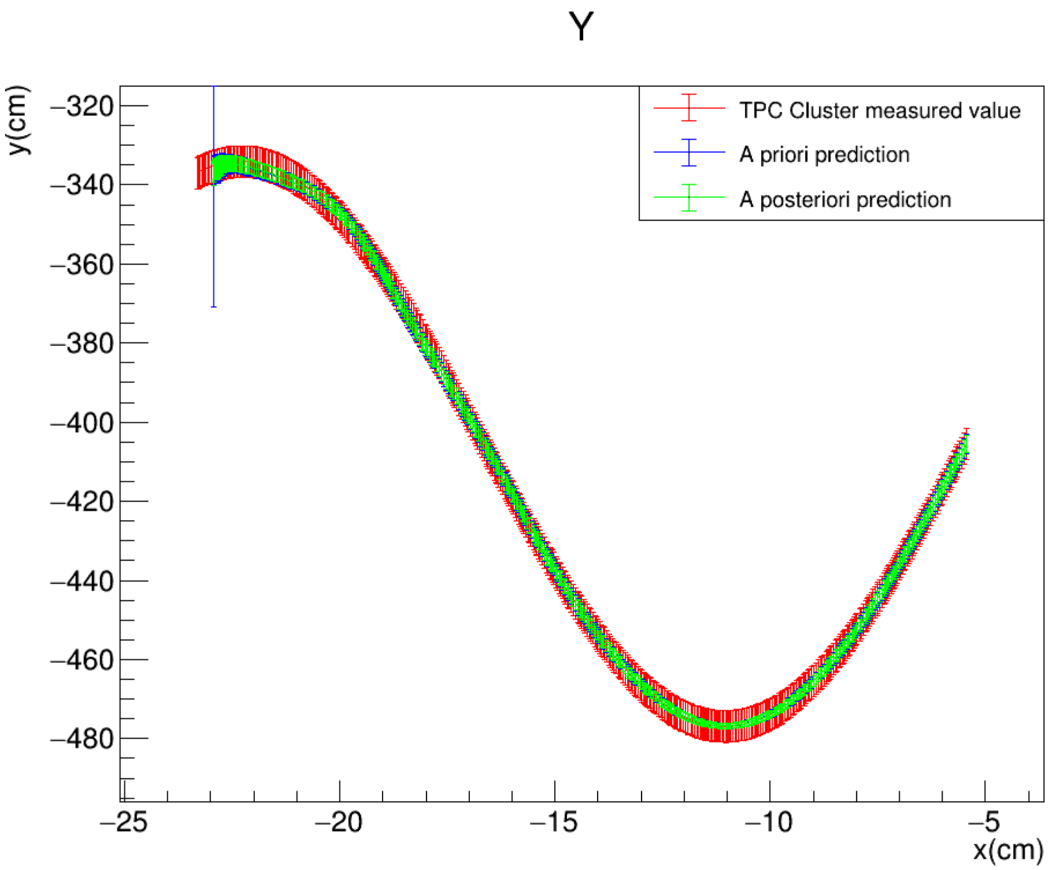
FITTER PROPAGATION DIRECTION

# RANDOMIZED X STEP: TESTING THE RATIO

- Since $\sigma_x$ and $\sigma_{yz}$ are essentially weights, only their ratio $\sigma_x/\sigma_{yz}$ should matter. To test this we try the pair $(\sigma_x, \sigma_{yz}) = (1cm, 8cm)$ which has the same ratio as $(\sigma_x, \sigma_{yz}) = (0.5cm, 4cm)$




$(\sigma_x, \sigma_{yz}) = (0.5cm, 4cm)$

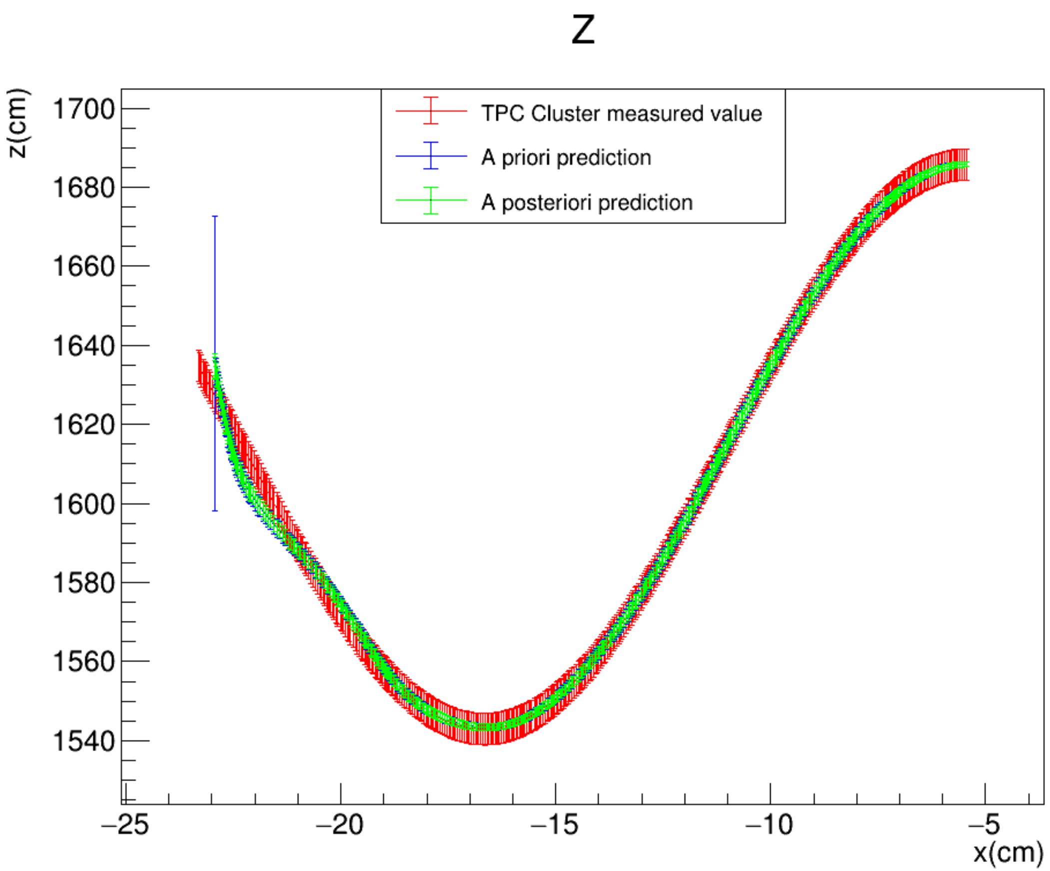



$(\sigma_x, \sigma_{yz}) = (1cm, 8cm)$

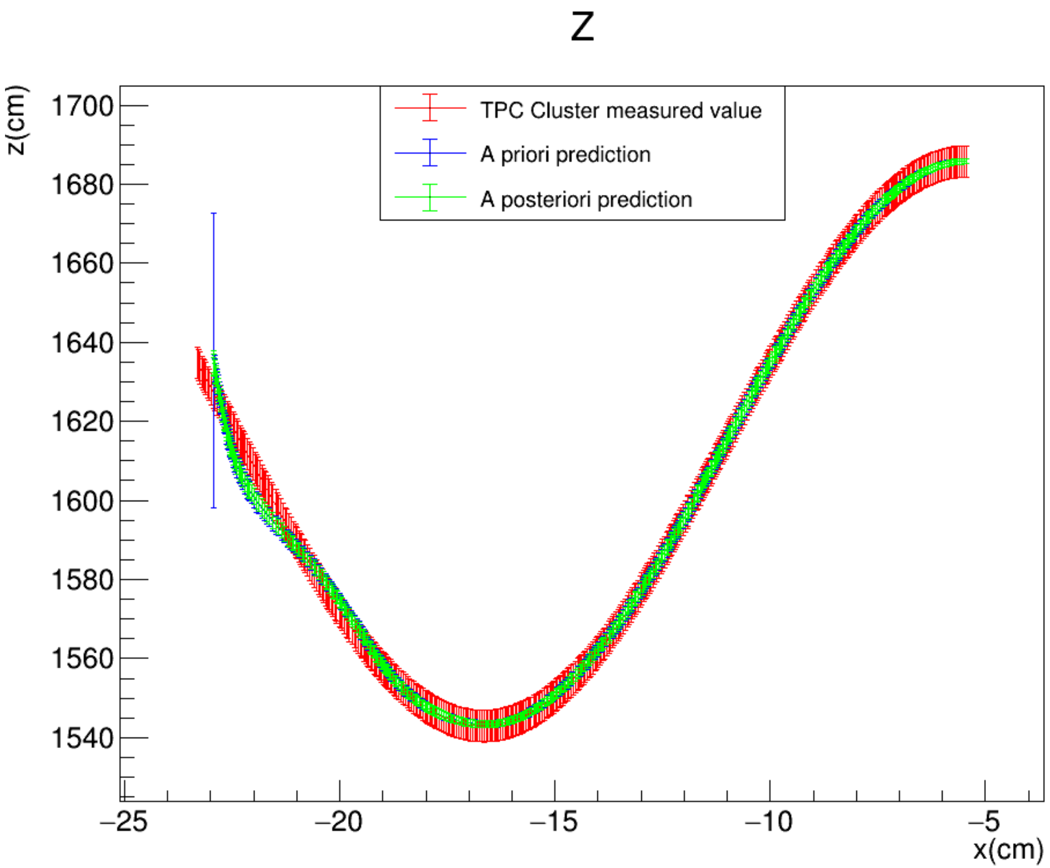
FITTER PROPAGATION DIRECTION

# RANDOMIZED X STEP: TESTING THE RATIO

- Since $\sigma_x$ and $\sigma_{yz}$ are essentially weights, only their ratio $\sigma_x/\sigma_{yz}$ should matter. To test this we try the pair $(\sigma_x, \sigma_{yz}) = (1cm, 8cm)$ which has the same ratio as $(\sigma_x, \sigma_{yz}) = (0.5cm, 4cm)$




$(\sigma_x, \sigma_{yz}) = (0.5cm, 4cm)$




$(\sigma_x, \sigma_{yz}) = (1cm, 8cm)$

FITTER PROPAGATION DIRECTION

# SUMMARY AND FUTURE STEPS

- The Kalman Filter now in use works in parallel with a Kinematic Fit which determines the evolution of the free parameter, minimizing the distance between the measured value and the a priori prediction

- The Kinematic fit is regulated by two weights $\sigma_{yz}$ and $\sigma_x$ whose ratio determines weather the fit is dominated by corrections related to the $yz$ or the $x$ measurements and predictions

- With the previous values of $(\sigma_x, \sigma_{yz}) = (0.5cm, 1cm)$ the Kalman Fit disregarded the $x$ measurements in the update
  - Increasing the value of $\sigma_{yz}$ ,bringing the ratio from $\sigma_{yz}/\sigma_x = 2$ to $\sigma_{yz}/\sigma_x = 8$ the fit better takes into account the x measurements and follows the Montecarlo truth more closely

- Future steps:
  - The optimal ratio of $\sigma_{yz}/\sigma_x$ needs to be found
  - A smeared helix Toy Montecarlo needs to be investigated
  - Other options for the free parameter need to be studied