SoLID Tracking

Weizhi Xiong
Duke University

SoLID Collaboration Meeting

Aug 26th ~27th, 2016

Outline

- SoLID-SIDIS tracking with 1 time-sample from APV25
- Kalman Filter as a track finding algorithm
  - Seed finding
  - Tracking following
  - Final selection
- Preliminary track finding and fitting results
- Conclusion
- Appendix
  - Other candidate algorithm for SoLID Tracking
  - GEM Occupancy for SIDIS and J/Psi

SIDIS Tracking with One Time Sample From APV

- Data size limitation allows only one time sample from the APV, if we aim for 100k Hz
  - Run APV in deconvolution-mode and take one deconvoluted time sample (still worth testing and considering)
  - Or one raw time sample (assumed for this study)
- With only one time sample, we cannot apply noise cut, the most effective cut so far for rejecting out-of-time noise (how this cut behave under trigger jitter and noise still need to be tested)
- Threshold cut on APV signal amplitude becomes essentially the only tool to suppress noise
- Fairly high occupancy and hit multiplicity after threshold cut (ADC = 120):

	GEM 1	GEM 2	GEM 3	GEM 4	GEM 5	GEM 6
Occupancy	2.5%	9.7%	4.1%	2.6%	2.0%	1.5%
Hit Multi.	420	5048	1860	1136	460	424

- Hit multiplicity contains false hits
- number will go up if consider 20 GEM sectors (currently assume 30)
- Lead to large amount of combinations ~10^{17} to 10^{18} currently

SIDIS Tracking with One Time Sample From APV

- At this level, Progressive tracking doesn't work
- Large angle can still hold up thanks to the LAEC
- Forward angle breaks down first due to high hit multiplicity and lack of enough support from downstream detectors (ECal, MRPC… are about 3m away downstream from the last GEM)
- Execution time also increase dramatically ($O(n^5)$ algorithm)

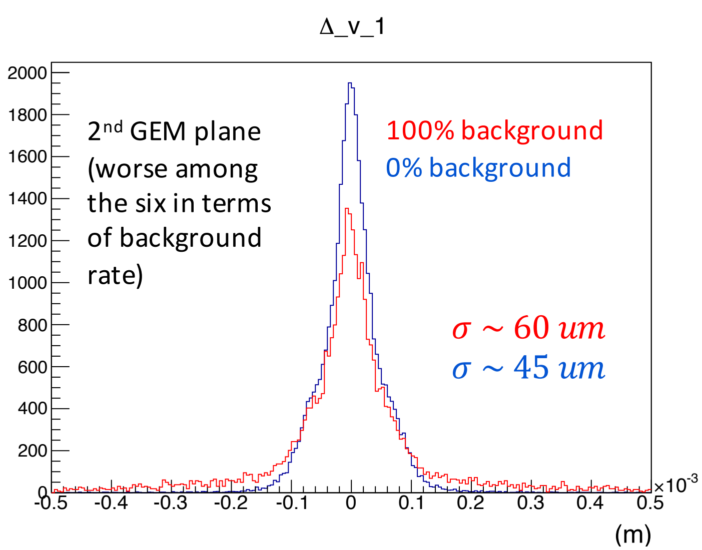
Previous result with Progressive tracking and one time sample from APV (Forward Angle)
Single electron signal track

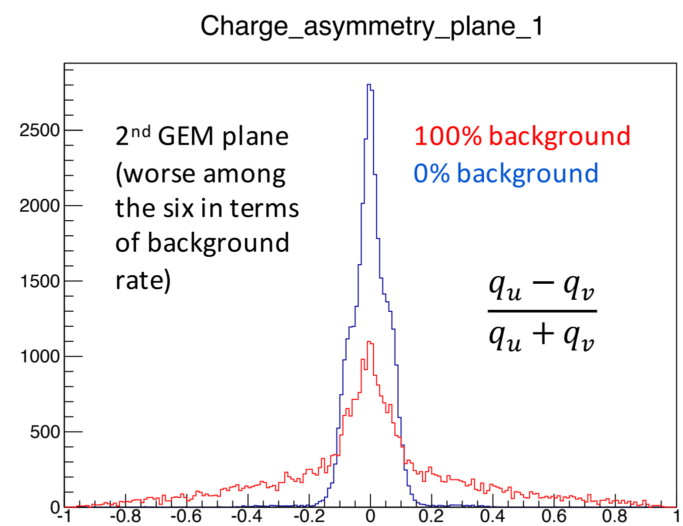
	Zero track	Single track	Multi track
Efficiency	2.1%	38.4%	59.5%

	0	1	2	3	4	5
# of misidentified hit per track	28.7%	8.6%	2.8%	2.6%	55.4%	1.9%

Some Small Modifications on GEM Digitization

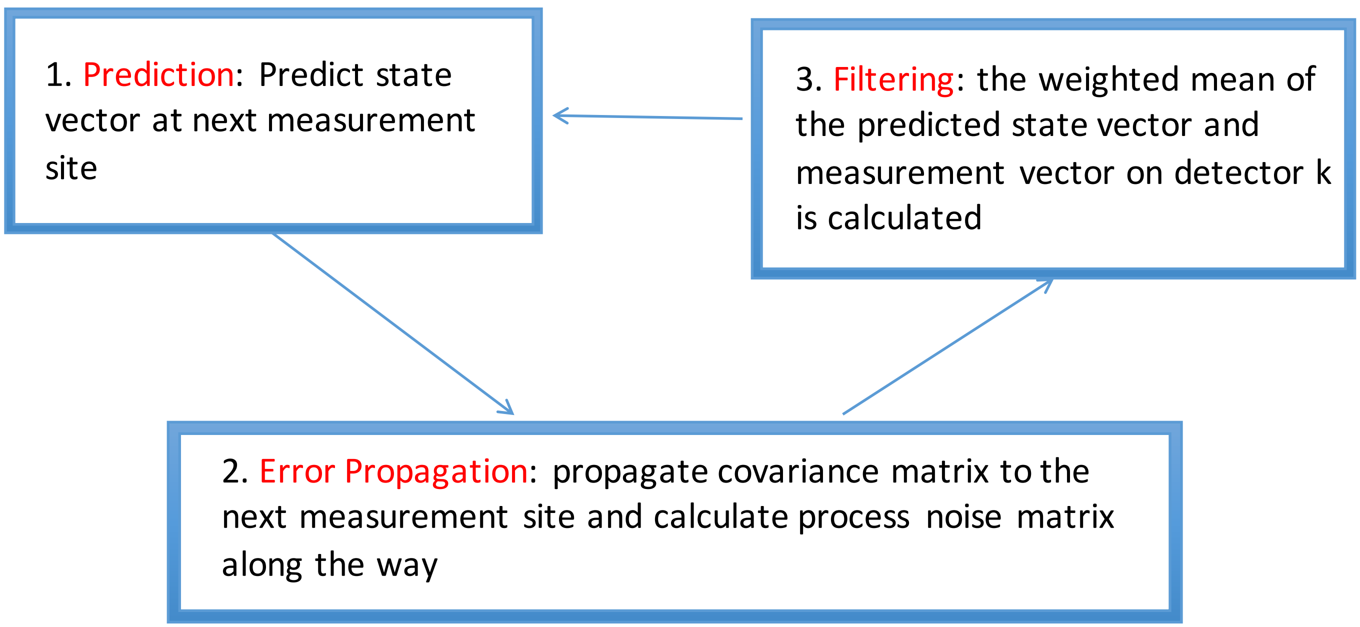
- Add Gaussian noise to digitized output (width = 20 ADC values, rough estimate for long strip for SoLID)
- Position resolution changes from ~30um to 45um at 0% background level (with one sample)
- Due to lack of statistics of background event, in addition to randomizing signal arrival time, also rotate the space with random azimuthal angle
- In order to keep the correlation for high energy background track, each background event has only one randomized azimuthal angle





Kalman Filter Algorithm

Kalman Filter: a recursive fitting algorithm based on χ^2 minimization



Kalman Filter Algorithm

Kalman Filter: a recursive fitting algorithm based on χ^2 minimization

1. Prediction: Predict state vector at next measurement site

3. Filtering: the weighted mean of the predicted state vector and measurement vector on detector k is calculated

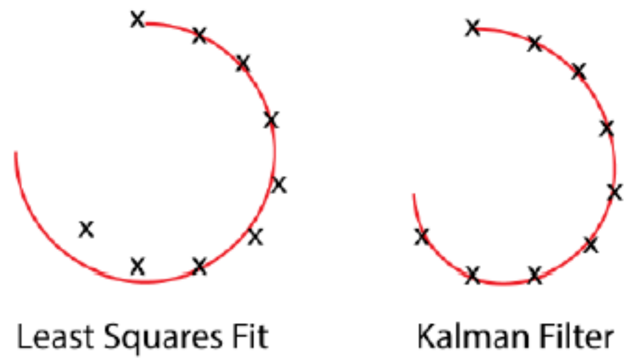
Using Kalman Filter as track finder

Arbitration: decide whether we want to accept the hit based on the prediction, the propagated covariance matrix and χ^2 increment if accept

2. Error Propagation: propagate covariance matrix to the next measurement site and calculate process noise matrix along the way

Kalman Filter Algorithm

- Track representation or state vector $(x, y, t_x, t_y, q/p)$
  - Allow smooth transition between uniform and fringe field
  - Rely completely on accurate field map measurement

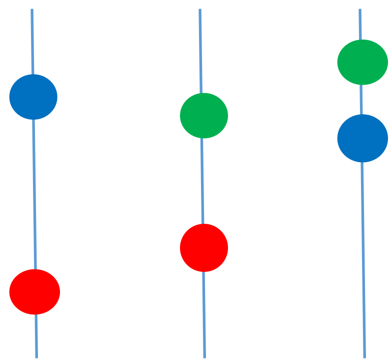


- Kalman Filter track finder advantages:
  - Evolution of track parameters, favors local information
  - Concurrent track finding and fitting
  - Discriminating power improved as more hits added
- Kalman Filter track finder disadvantages:
  - Relatively slow due to field propagation and large computation power requirement (5-D matrices propagation, multiplication and inversion)
  - Weak discriminating power at early stage
  - Rely on efficient seed finding

First Step in KF Track Finding -- Seeding

- Seed for Kalman Filter: the minimum number of hits that can be used to give initial estimation of the track parameters
- Requirement for seed finding:
  - Selection rules too rigorous -> lost of true seed at the beginning, almost impossible to recover the true track
  - Selection rules too loose -> large amount of seeds, greatly increase execution time
  - Need to consider multiple seed pattern: the true hit may lost due to GEM inefficiency
  - This can by itself an independent tracking algorithm
- Current seed finding strategy:
  - Look for three types of doublet seeds from the three most downstream GEMs detectors
  - Using analytic formulae to estimate initial track parameters
  - Use sanity cuts on the estimated momentum and angles,
  - Use Runge-Kutta propagation to check it (come from target and lead to a hit on EC)
  - Merge doublet seeds to form triplet seeds (requires three types of double seed joint at front, middle and back GEM)
  - Once a triplet seed is form, deactivate the corresponding doublets (avoid repetition of track finding)

Three types of doublet seeds

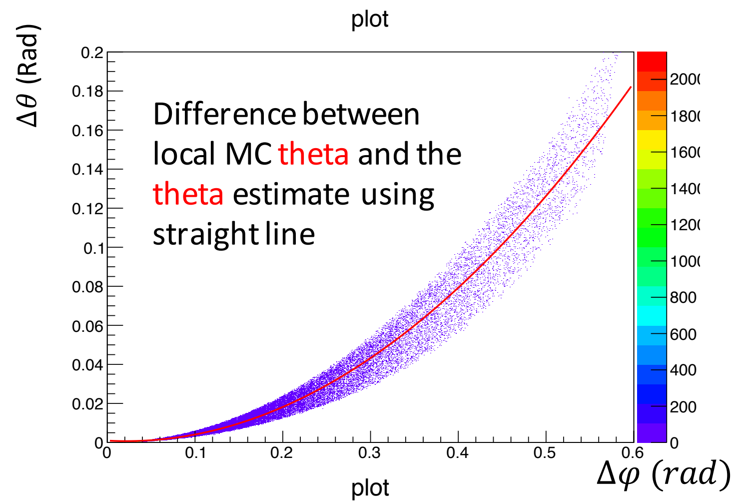


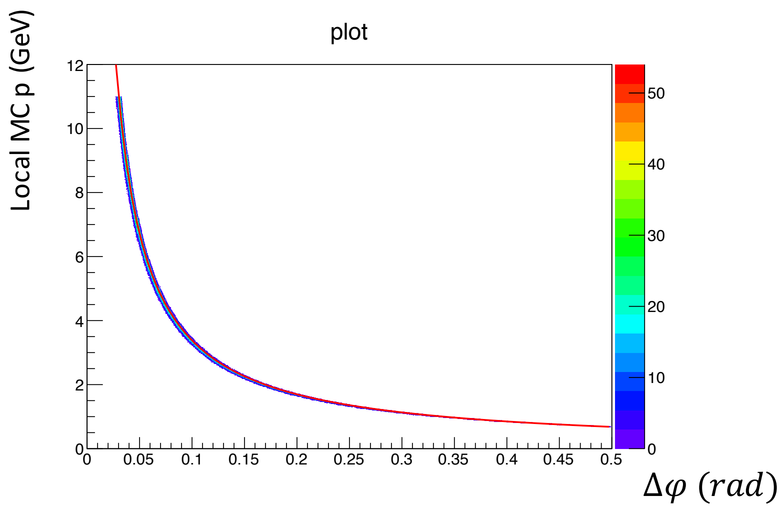
First Step in KF Track Finding -- Seeding

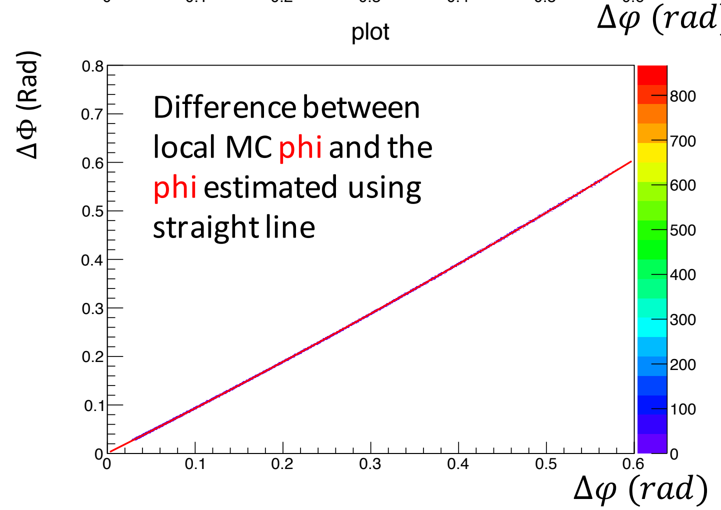
- Using coordinate difference between two hits to estimate the local momentum and angle info of the track (Kalman Filter needs local info)

- Assume the track is straight at first so:
  $\theta = \mathrm{atan}(\Delta r/\Delta z)$ and $\Phi = \mathrm{atan}(\Delta y/\Delta x)$

- Using local MC information to correction functions to estimate the real theta and phi of the track

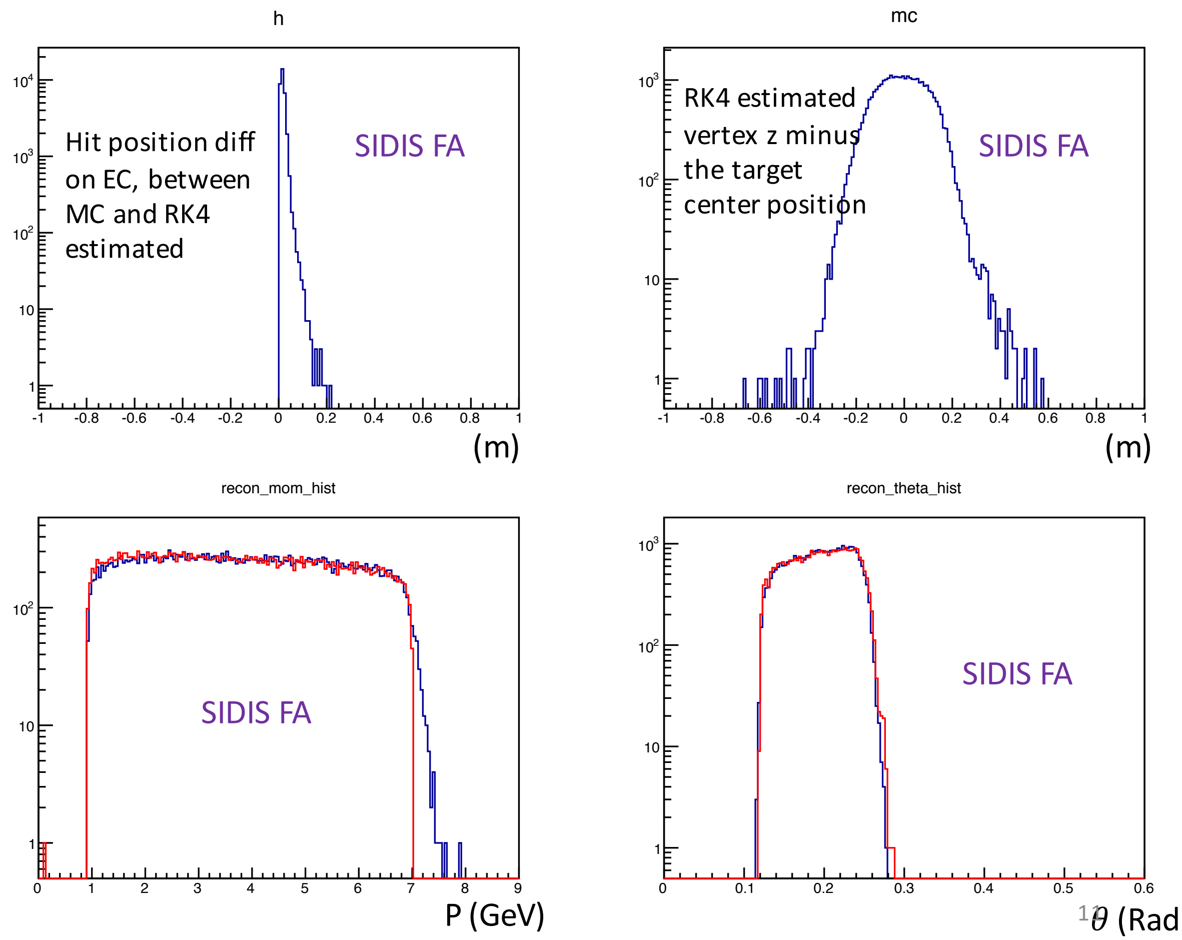






First Step in KF Track Finding -- Seeding

- Using Runge-Kutta for the doublet seed to check whether it can connect to EC and target
  - Distance between predicted EC hit and actual EC hit < 20 cm for FA (6 cm for LA)
  - Difference between Predicted vertex z and target center < 60 cm for FA (50 cm for LA)
- Using analytic formulae and two hits to estimate the initial local momentum and angles of the track
  - Red distribution – MC local info
  - Blue distribution – Estimated local info



Second Step in KF Track Finding – Track Following

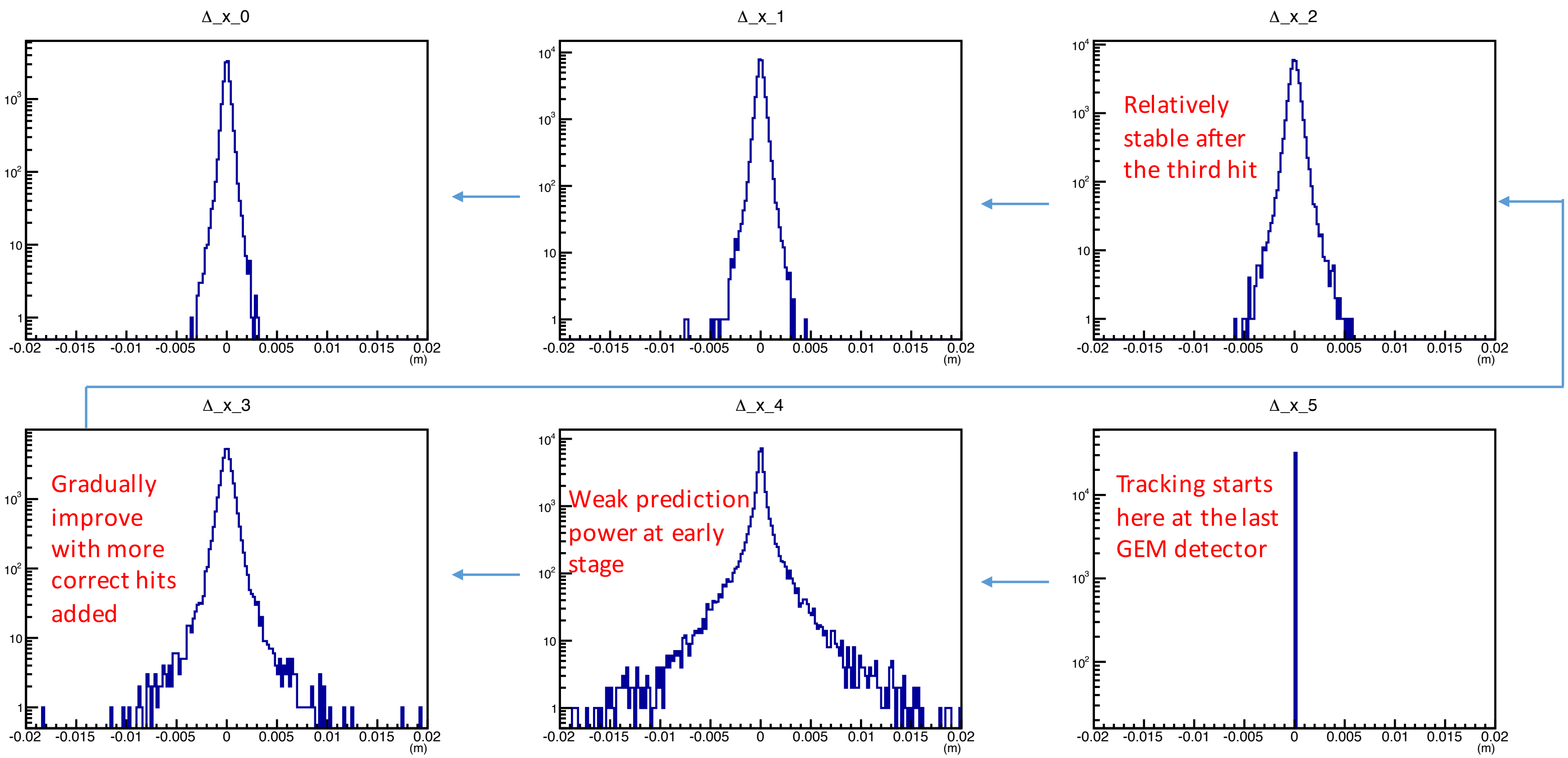
- Follow the direction of seed, predict next measurement

- Add if a hit is found, which satisfies the following requirements:
  - Fall within the prediction window (window size fixed at early stage, auto-adjustable later on based on the covariance matrices of the track parameters)
  - χ^2 increment less than 60

- Can tolerate no more than 1 missing hit:
  - Triplet seed can have one more missing hit later on
  - Doublet seed cannot miss anymore
  - Tracks have at least 4 hits in FA, and 3 hits in LA

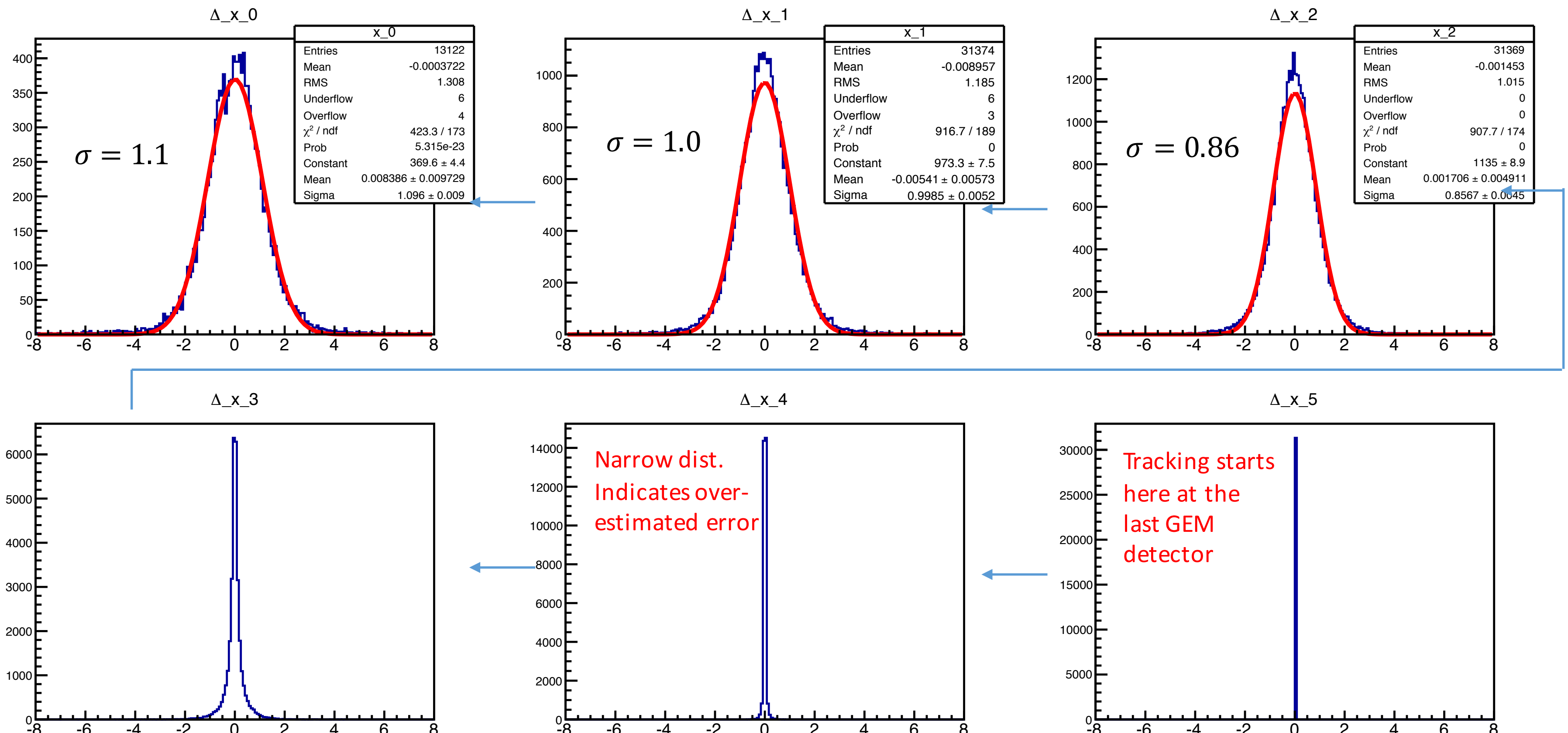
Second Step in KF Track Finding – Track Following

Kalman Filter predicted x minus actual measured x coordinate (FA, 0% background)



Second Step in KF Track Finding – Track Following

Kalman Filter predicted x minus actual measured x coordinate, weighted by the error in prediction (FA, 0% background)

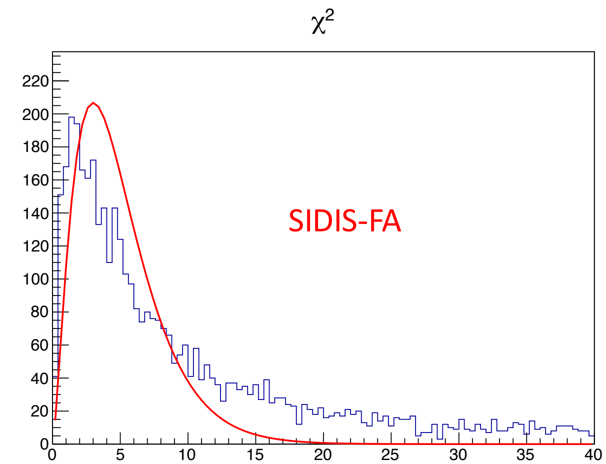


Third Step in KF Track Finding – Final Selection

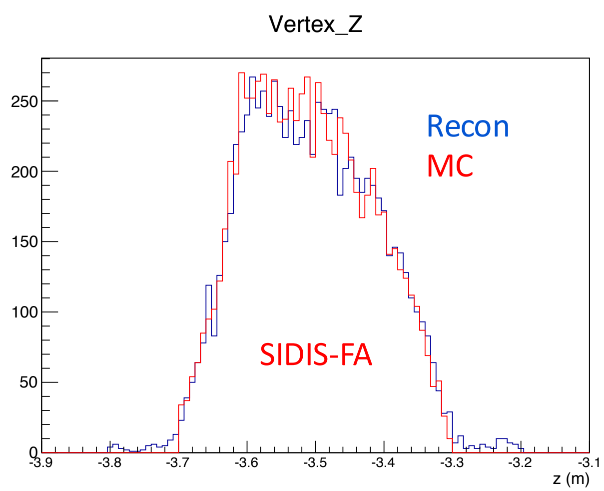
- Charge asymmetry: require at least three hits have good charge asymmetry
  - $q_u - q_v / q_u + q_v < 0.5$
- χ^2/ndf < 30
- Tracks that satisfied the above condition will be propagated to the target and downstream detectors
  - Finer cut for the vertex z position
  - Finer cut for the match between predicted EC hit and actual EC hit position
  - EC Cluster energy match with the track energy (+/- 50%), only for LA
- Rank tracks with certain rules
  - Track with more hits ranks higher
  - Track with smaller χ^2/ndf ranks higher
  - Select the best track if share common hits

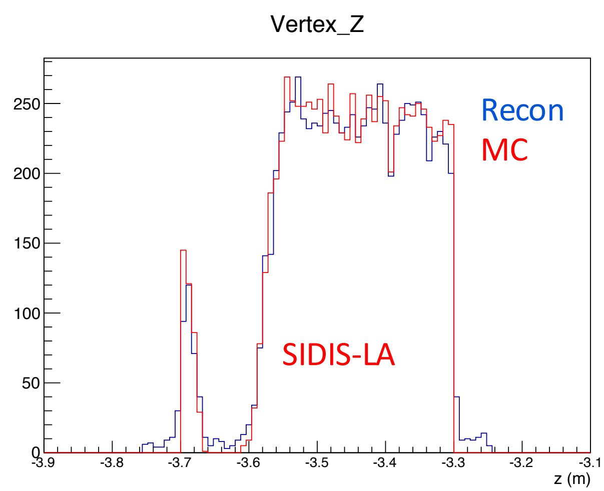
Third Step in KF Track Finding – Final Selection

- χ^2distribution doesn't quite follow the expected curve, likely due to overestimation of error at early stage. Need some fine-tuning



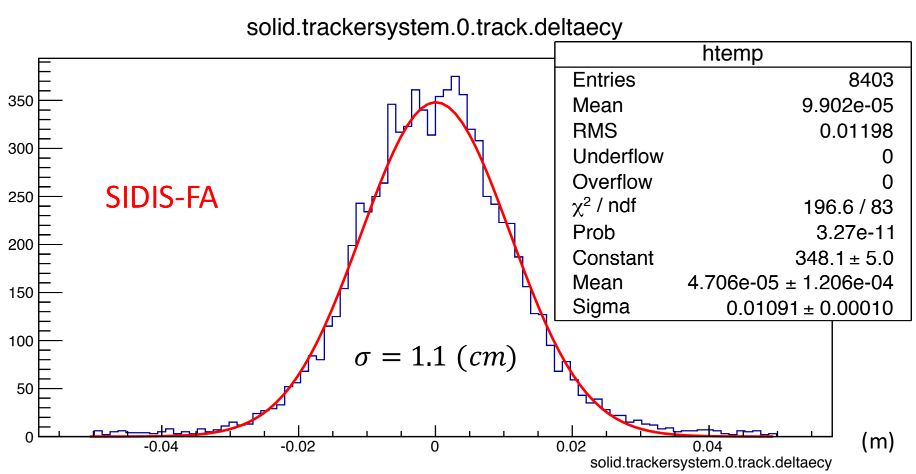
- +/- 5cm from the edge of the target cell (10 cm for FA as vertex z resolution will be worse)

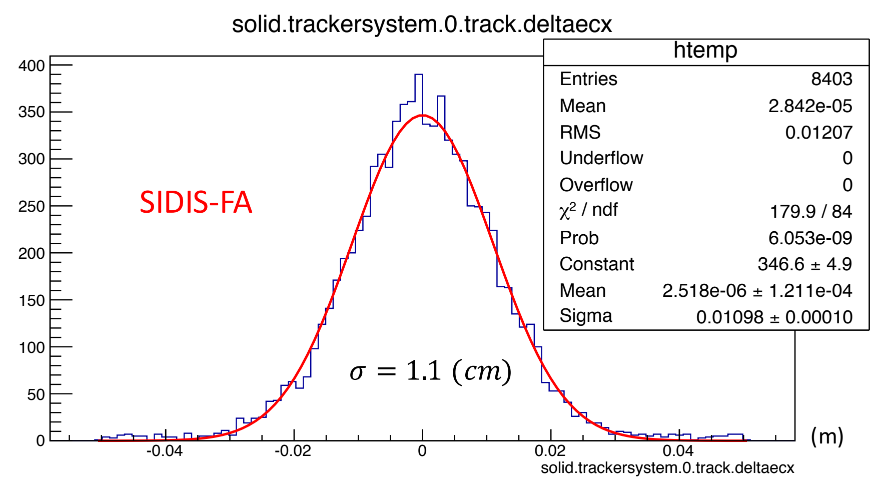




Third Step in KF Track Finding – Final Selection

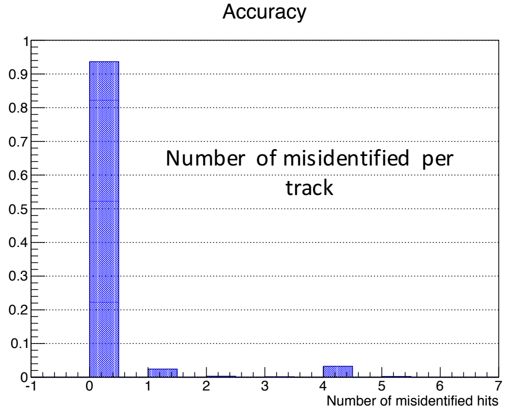
- Optimal state at the vertex at the moment
- Past sites do not contain all the hit information
- Before we do a final match between the track and downstream detectors, we can either:
  - Refit the track forward
  - Using Kalman Filter smoothing technique (re-evaluate the past site based on saved state vectors and covariance matrices so far)
- Cut at +/- 5cm from the EC hit

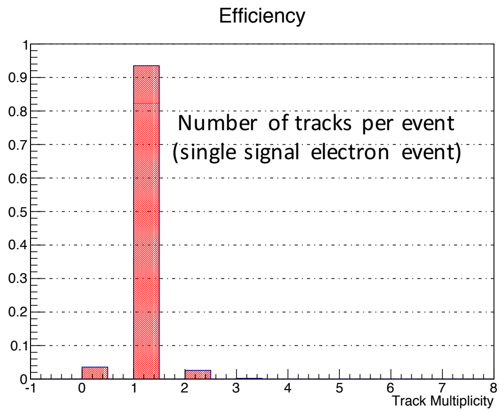




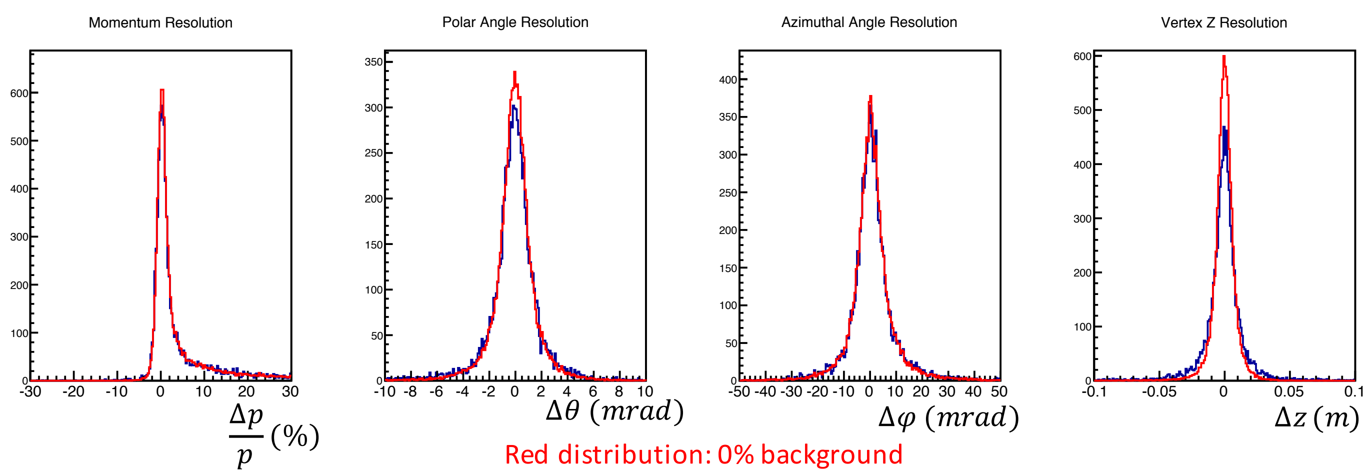
KF Track Finding – FA Preliminary result

- Condition: (1) Single electron signal. (2) 100% background. (3) GEM resolution (45um at 0% background) (4) EC resolution (1cm for position, $10\%/\sqrt{E}$ for energy) (5) p of signal track 0.9 ~ 7 GeV





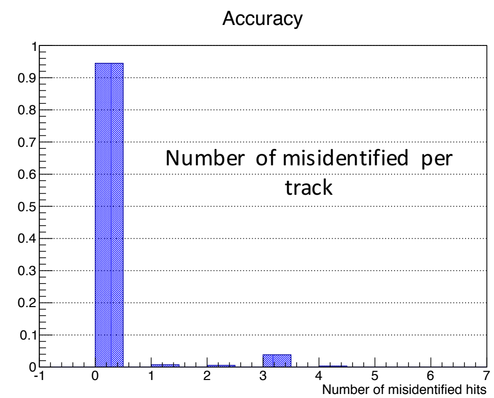
	$\Delta p/p$ (%)	$\Delta\theta$ (mrad)	$\Delta\varphi$ (mrad)	Δz (mm)
0% background	1.23	1.13	5.31	6.32
100% background	1.28	1.22	5.36	8.60

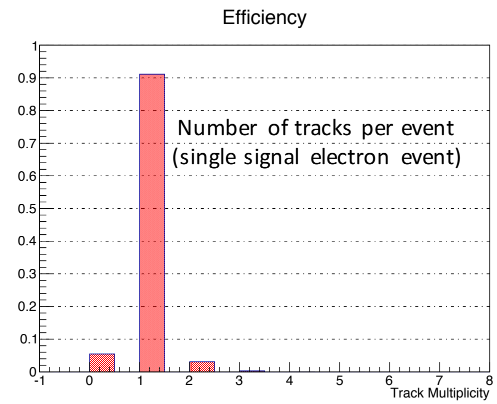


Red distribution: 0% background
Blue distribution: 100% background

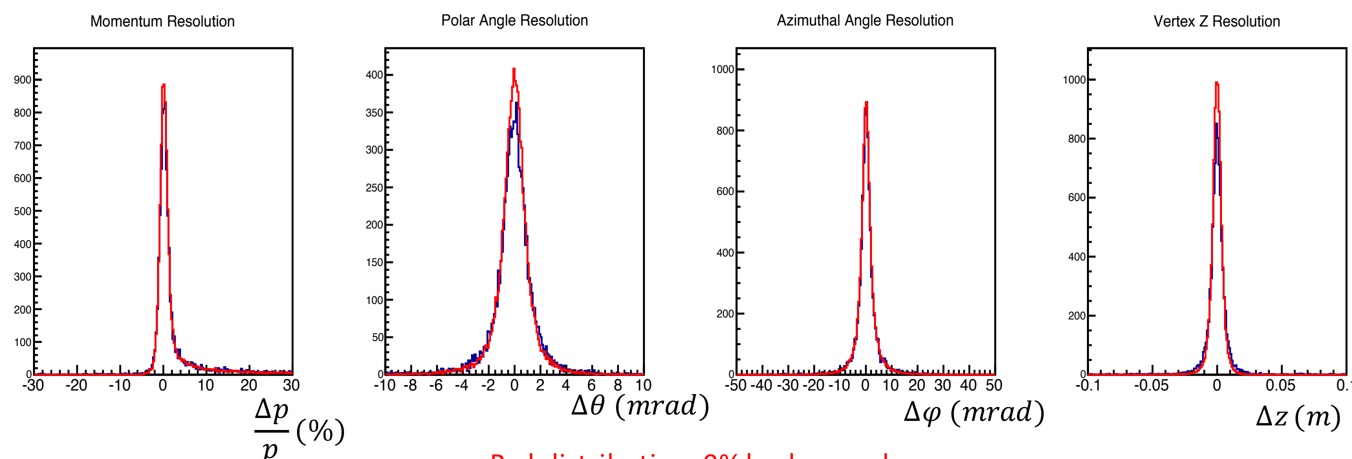
KF Track Finding – LA Preliminary result

- Condition: (1) Single electron signal. (2) 100% background. (3) GEM resolution (45um at 0% background) (4) EC resolution (1cm for position, $10\%/\sqrt{E}$ for energy) (5) p of signal track 0.9 ~ 7 GeV





	$\Delta p/p$ (%)	$\Delta\theta$ (mrad)	$\Delta\varphi$ (mrad)	Δz (mm)
0% background	0.99	0.95	2.16	3.70
100% background	1.05	1.08	2.19	4.51



Red distribution: 0% background

Blue distribution: 100% background

Conclusion

- Framework of Kalman Filter track finding has been developed
- For SIDIS:
  - Give acceptable result with 1 time sample for single electron event
  - To do: Add hadron and do coincident tracking (2~4 weeks)

- For PVDIS:
  - In principle the same algorithm works (has been used to do track fitting)
  - To do: digitize PVDIS events and make sure the program runs for the configuration (2~4 weeks). Test how the noise cut behave with trigger jitter and noise.
  - Certain modification may needed based on the characteristic of PVDIS tracks

- For J/Psi:
  - Will be extremely challenging if use only one sample (much higher GEM occupancy and low momentum of signal particle)

Conclusion

	GEM Digitization	GEM Decoding	Track Finding	Track Fitting
SIDIS-He^3	Yes	Yes	Only electron	Yes
SIDIS-Proton	Yes	Yes	No	No
PVDIS	Yes*	Yes*	Only Field Off	Yes
J/Psi	Yes	Yes	No	Yes

- GEM digitization need to keep up with simulation (requires continuous development)
- * For PVDIS, only old digitization data exist (readout strip arranged in favor of Tree search), should be easy to get new digitization

Other Tracking Algorithm for SoLID

- Straight track is easier to find than curved track
- If tracks are straight in one space, we should make the most out of it (PVDIS tracks in r-z space)
- If tracks are not straight, sometimes they can be mapped into other space where they are straight

Other Tracking Algorithm for SoLID

- Straight track is easier to find than curved track
- If tracks are straight in one space, we should make the most out of it (PVDIS tracks in r-z space)
- If tracks are not straight, sometimes they can be mapped into other space where they are straight
  - Conformal mapping – mapping circles that pass the origin into straight line

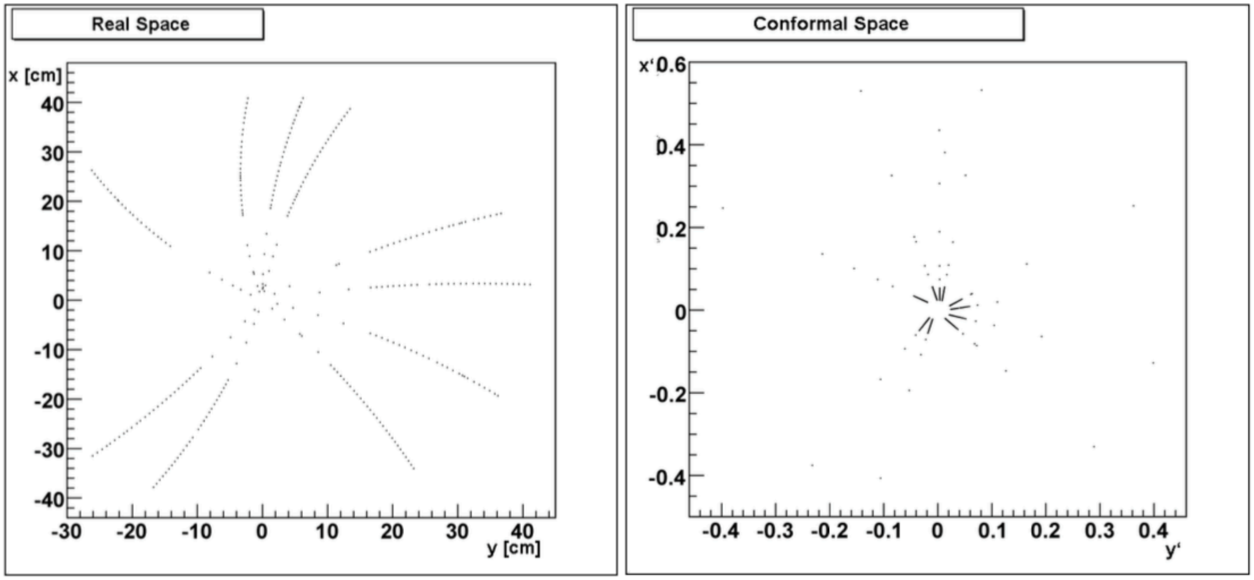
$$u = \frac{x}{x^2 + y^2}$$

$$v = \frac{y}{x^2 + y^2}$$

$$(x-a)^2 + (y-b)^2 = R^2$$

$$\downarrow$$

$$v = \frac{1}{2b} - u\frac{a}{b}$$



See http://nuclear.gla.ac.uk/twiki/pub/Main/PandaFeDatWorkshop2009April/3-4.Muenchov_David.FPGA_tracking_at_HADES.pdf for more details

Other Tracking Algorithm for SoLID

- Straight track is easier to find than curved track
- If tracks are straight in one space, we should make the most out of it (PVDIS tracks in r-z space)
- If tracks are not straight, sometimes they can be mapped into other space where they are straight
  - Conformal mapping – mapping circles that pass the origin into straight line
  - Riemann track finder – mapping 2-D circle into 3-D plane
    - Using $w = x^2 + y^2$, turning 2D circle into 3D plane, thus slow and non-linear circle fit can be replaced by fast linear plenary fit
    - $(x - a)^2 + (y - b)^2 = R^2 \longrightarrow w - 2ax - 2by + a^2 + b^2 - R^2 = 0$
    - And then notice that for helix along z-axis: $z = R\varphi / \tan(\theta)$
    - Thus a complete helix fit is replace by a plenary fit plus a straight line fit
    - Basic step in track finding can be quite similar as Kalman Filter. However, requires relatively uniform field

See https://indico.gsi.de/getFile.py/access?contribId=3&resId=0&materialId=slides&confId=665 for more details

Other Tracking Algorithm for SoLID

- Straight track is easier to find than curved track
- If tracks are straight in one space, we should make the most out of it (PVDIS tracks in r-z space)
- If tracks are not straight, sometimes they can be mapped into other space where they are straight
  - Conformal mapping – mapping circles that pass the origin into straight line
  - Riemann track finder – mapping 2-D circle into 3-D plane

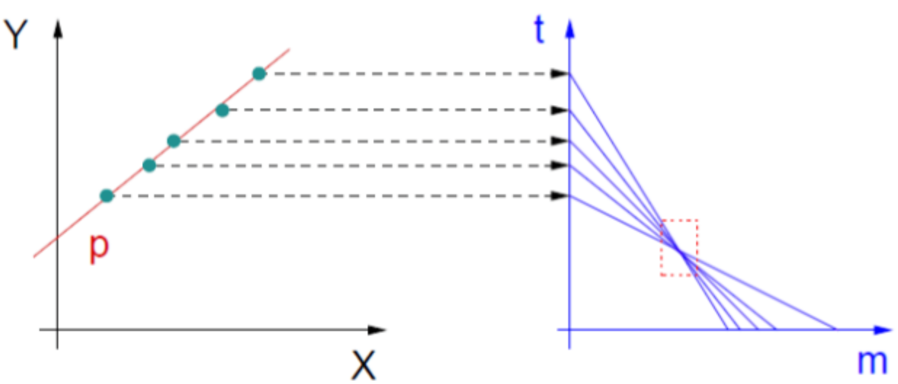
- Techniques in identifying straight track:
  - Straight line least square fit
  - Tree search – template matching algorithm

Other Tracking Algorithm for SoLID

- Straight track is easier to find than curved track
- If tracks are straight in one space, we should make the most out of it (PVDIS tracks in r-z space)
- If tracks are not straight, sometimes they can be mapped into other space where they are straight
  - Conformal mapping – mapping circles that pass the origin into straight line
  - Riemann track finder – mapping 2-D circle into 3-D plane

- Techniques in identifying straight track:
  - Straight line least square fit
  - Tree search – template matching algorithm
  - Hough transform – mapping points in physical space into lines in parameter space

  - In physical space $y = mx + t$
  - In parameter space $t = -xm + y$
  - Co-linear hits becomes intersecting lines in the parameter space
  - Turning straight track identification problem into problem of find maxima in histogram



Other Tracking Algorithm for SoLID

- Straight track is easier to find than curved track
- If tracks are straight in one space, we should make the most out of it (PVDIS tracks in r-z space)
- If tracks are not straight, sometimes they can be mapped into other space where they are straight
  - Conformal mapping – mapping circles that pass the origin into straight line
  - Riemann track finder – mapping 2-D circle into 3-D plane

- Techniques in identifying straight track:
  - Straight line least square fit
  - Tree search – template matching algorithm
  - Hough transform – mapping points in physical space into lines in parameter space
  - Artificial neutral network – favors straight track, but allows small banding angle

Other Tracking Algorithm for SoLID

- Straight track is easier to find than curved track
- If tracks are straight in one space, we should make the most out of it (PVDIS tracks in r-z space)
- If tracks are not straight, sometimes they can be mapped into other space where they are straight
  - Conformal mapping – mapping circles that pass the origin into straight line
  - Riemann track finder – mapping 2-D circle into 3-D plane

- Techniques in identifying straight track:
  - Straight line least square fit
  - Tree search – template matching algorithm
  - Hough transform – mapping points in physical space into lines in parameter space
  - Artificial neutral network – favors straight track, but allows small banding angle

- Kalman filter is rather popular fitter in tracking reconstruction, but it may not be the best in various situation
  - Not optimal if noise is non-Gaussian: Bremsstrahlung radiation for electrons
  - Gaussian Sum Filter: approximate noise with Gaussian mixture
  - Deterministic Annealing Filter: deal with competing measurements and reject wrong assignment of hits

- Various machine learning algorithm can be useful as well i.e. boosted decision tree may be useful for seeding and finally selection of tracks

GEM Occupancy

SIDIS

Threshold	GEM 1 (%)	GEM 2 (%)	GEM 3 (%)	GEM 4 (%)	GEM 5 (%)	GEM 6 (%)
0	5.00	17.23	8.90	5.00	4.83	3.72
60	3.49	13.22	6.04	3.57	2.98	2.22
80	3.09	11.92	5.28	3.18	2.58	1.91
100	2.75	10.75	4.63	2.84	2.25	1.66
120	2.47	9.70	4.08	2.55	1.99	1.46
140	2.21	8.78	3.63	2.31	1.78	1.30

J/Psi

Threshold	GEM 1 (%)	GEM 2 (%)	GEM 3 (%)	GEM 4 (%)	GEM 5 (%)	GEM 6 (%)
0	11.40	22.44	14.89	12.30	12.38	10.25
60	8.39	17.72	11.00	8.98	8.35	6.69
80	7.63	16.27	9.95	8.11	7.39	5.88
100	7.00	14.96	9.05	7.38	6.61	5.22
120	6.47	13.8	8.29	6.74	5.95	4.68
140	6.01	12.76	7.62	6.20	5.39	4.22