

# Status Update of DATCON

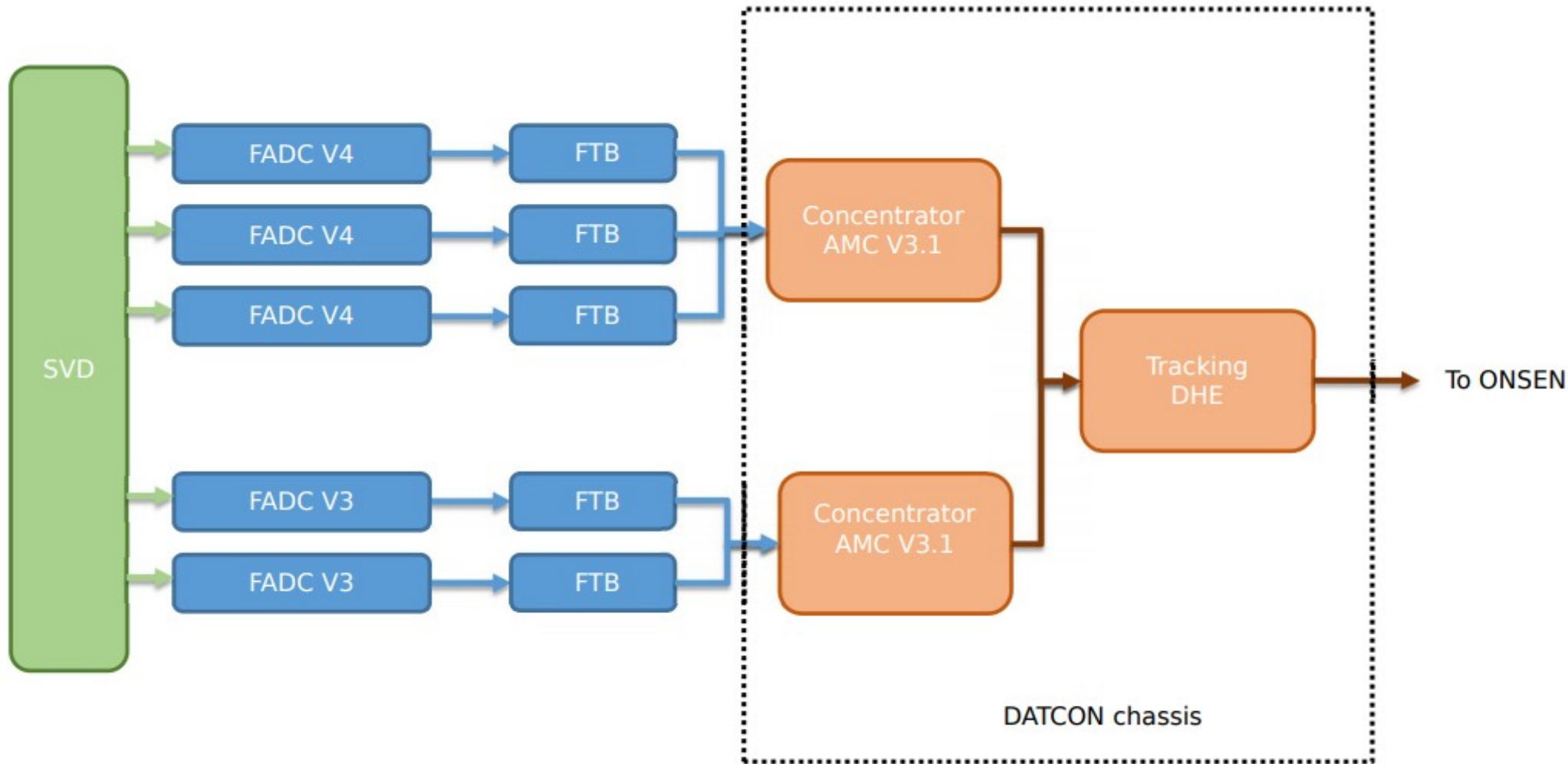
Belle II PXD Meeting 10.04.2018

Christian Wessel, Bruno Deschamps,  
Jochen Dingfelder, Carlos Marinas

Universität Bonn

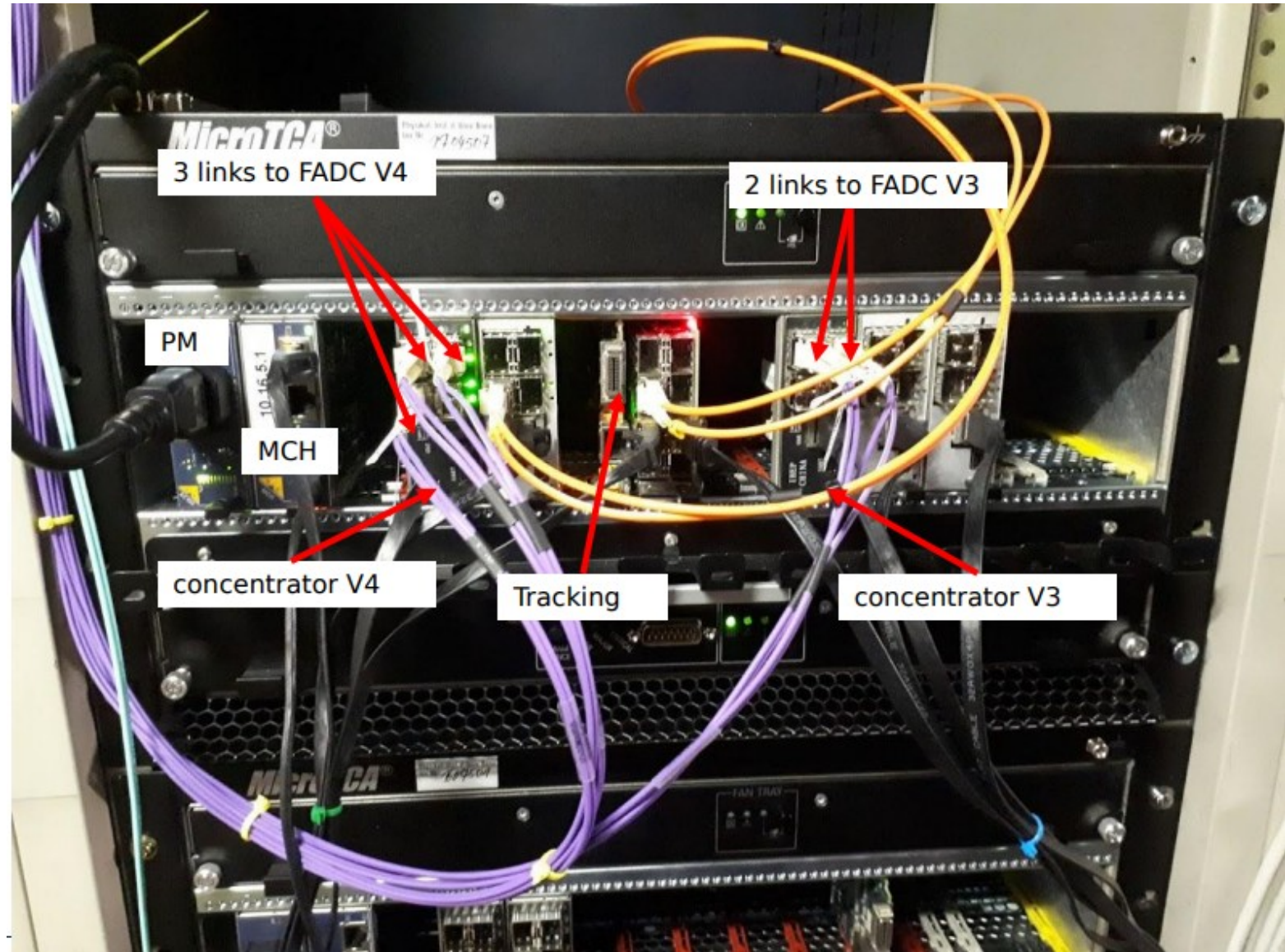
# Hardware Status Update

# Phase 2 Setup



On behalf of Bruno

# Phase 2 Setup



On behalf of Bruno

- From January DESY workshop:
  - Excluding tracking (tracking, extrapolation, ROI calculation)
  - Validate event building
  - Send fixed ROI to ONSSEN with current event number
- At KEK:
  - Most of the time crashing
  - Very unstable
- Symptoms:
  - FTB link having different trigger number
  - Slow control showed different SOF counter value for FTB link
  - Link up sometimes going down in slow control
  - Shifted data in DATCON
  - Missing end\_of\_event signal

On behalf of Bruno

- Diagnostic:
  - Spent lot of time trying to understand which part of DATCON firmware was causing the issues with no success
  - Investigating the link stability we discovered the link going down very often, even while transmitting data (very bad!)
- Solution
  - We had a look into FTB firmware, more precisely, in the sender to DATCON
  - The clock compensation was only made once, thus the link was always lost
  - Change FTB firmware
  - FTB-DATCON link always stable, SOF counters have the same value
  - Every event build correctly, transmitted to the tracking boards and ROI is sent to ONSEN
  - Tested many times for a maximum of 8 hours. Every event received was sent to ONSEN
- Now including tracking part, plan on testing on hardware this week

On behalf of Bruno



- Where and how can we store DATCON ROI for later analysis?
  - HLT ROI don't need to be stored, since they can easily be re-calculated using the very same algorithms, but DATCON ROIs can't (even if we would have the very same algorithms in BASF2, we still don't know if the ROI really are the same when calculated in BASF2 compared to FPGA)
- We need some mechanism for storing DATCON ROI for later analysis
  - Needed for testing whether everything still works fine
    - From ROI we can calculate extrapolated hit and thus residuals
  - Checking performance in case of firmware changes / updates
  - Comparison with BASF2 ROI and extrapolated hits
  - Not all and not always, but time by time with a frequency of O(kHz) or so
    - Could use ONSEN "sendROI" flag, but then also HLT ROI are saved
    - OR: could maybe send ROI to BonnDAQ from DATCON directly
      - What would be the impact on BonnDAQ (storage, ethernet, ...)

# Simulation Status Update



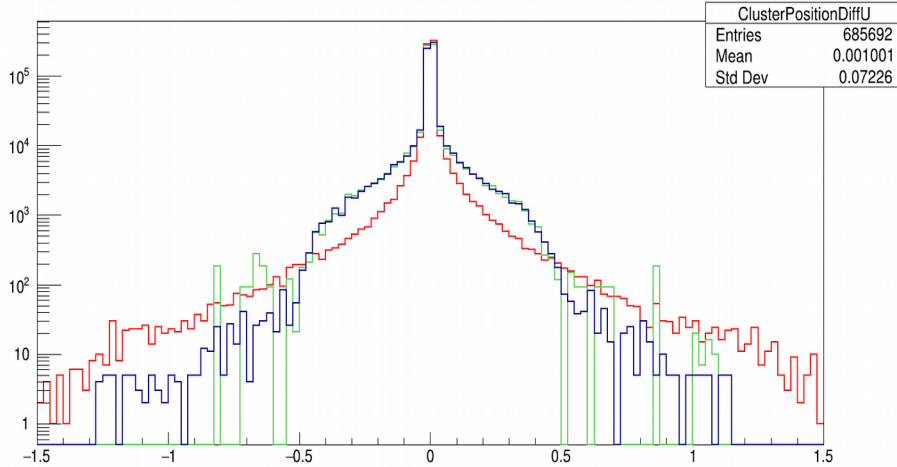
- Until last update, tracking within the DATCON basf2 implementation was performed based on SVDTrueHit information (w/o background) or SVDSimHit information (w/ background)
- The performance was good and the results were promising, but input data were not what will be available for the FPGA implementation → needed to change this!
- Now: simplified clustering of SVD data with very simple noise filter and SpacePoint creation based on this
- Clusterizer and SpacePointCreator are adapted / simplified versions of SVDSimpleClusterizerModule and SVDSpacePointCreatorModule
  - should be easy to convert DATCONSVDCluster / DATCONSVDSpacePoints into SVDCluster / SpacePoint for comparison / debugging / RecoTrack creation

- 5000 Y(4S) events without background
- Simulations with background to follow
- Using very simple noise filter for SVD data
- Self-made analysis function for clusters and SpacePoints
  - I tried to do some kind of truth matching, probably I didn't do it the best way possible, but it works and the results look reasonable
  - (Maybe) strange thing: DATCONSVDCluster and DATCONSVDSpacePoints look better in terms of residuals with SVDTrueHits compared to SVDSpacePoints (see next slides)
- Parameters used in the tracking algorithm only were optimised for old version relying on SVDTrueHits / SVDSimHits
  - Optimisation for custom clusters and SpacePoints has to be done in future

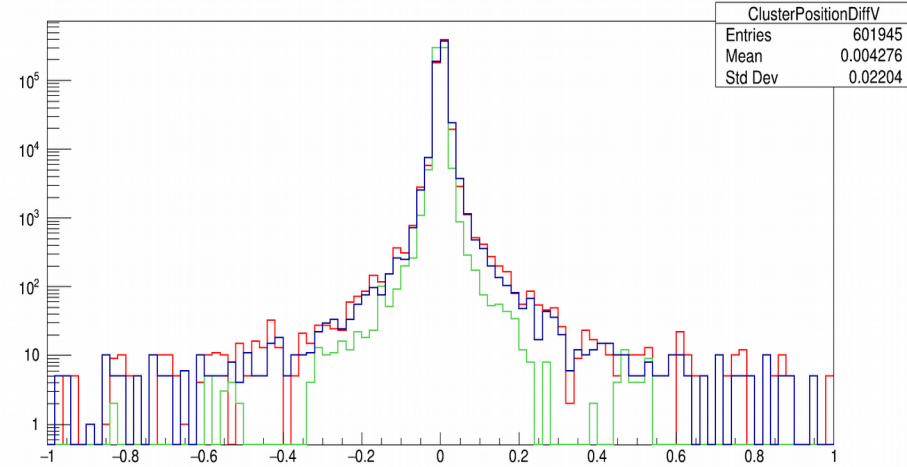
# Simulation Results - Clusters

red = (SVDCluster-DATCONCluster), green = (SVDTrueHit-SVDCluster), blue = (SVDTrueHit-DATCONCluster)

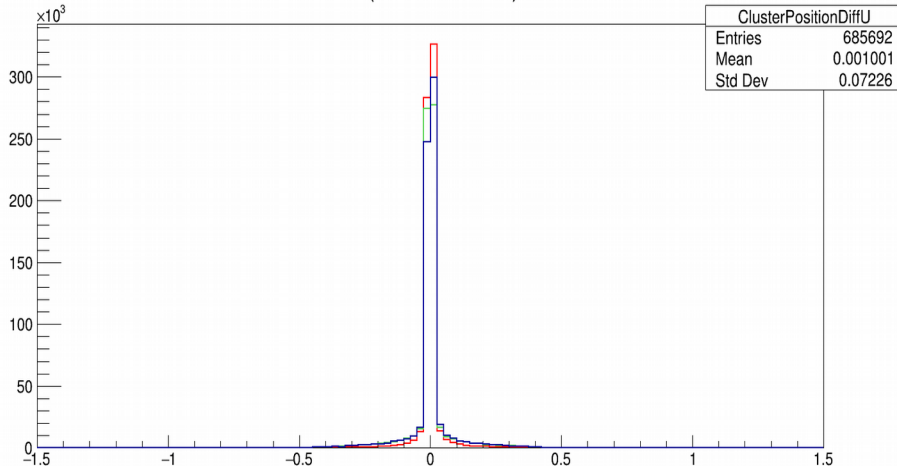
Cluster difference (SVD-DATCON) of U-side in cm



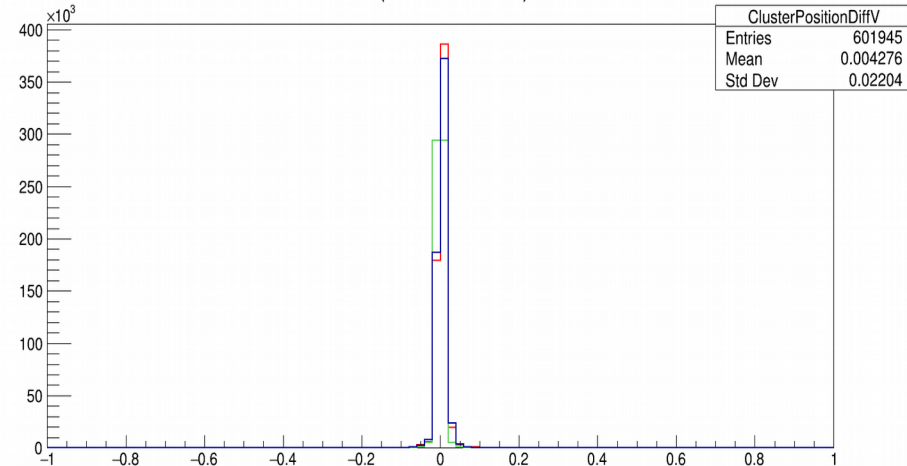
Cluster difference (SVD-DATCON) of V-side in cm



Cluster difference (SVD-DATCON) of U-side in cm

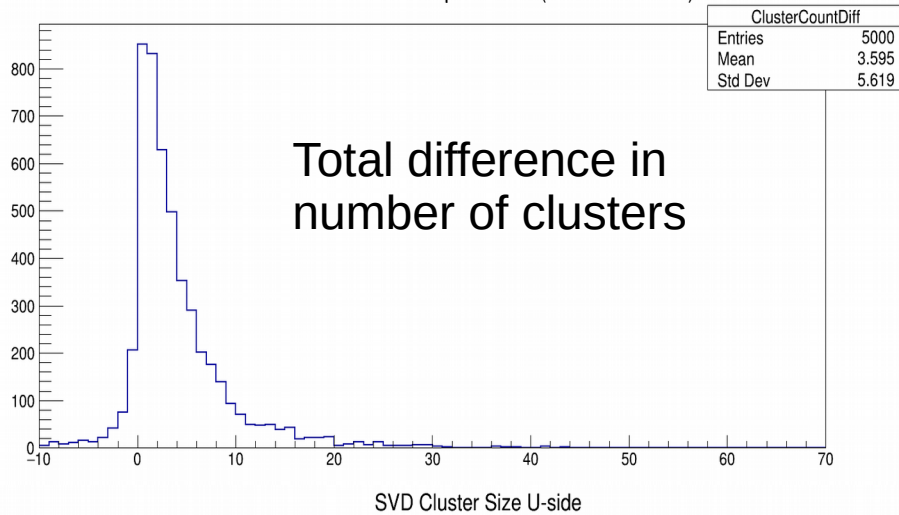


Cluster difference (SVD-DATCON) of V-side in cm

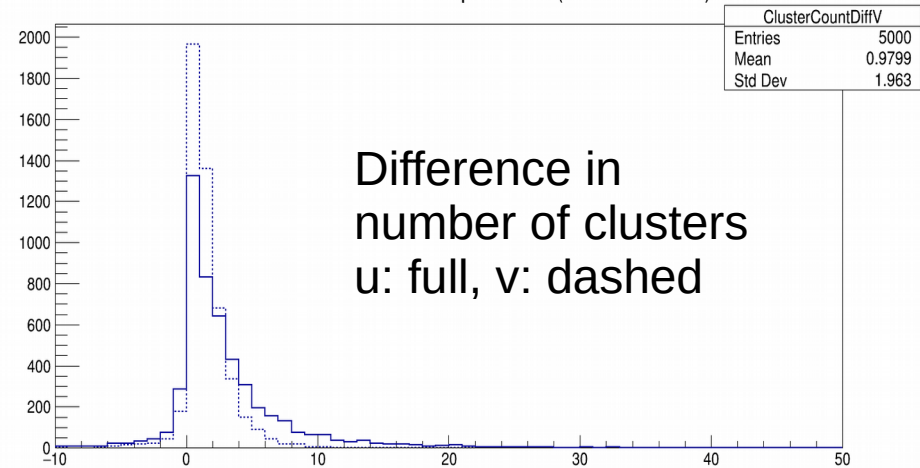


## Difference in number of clusters and cluster size

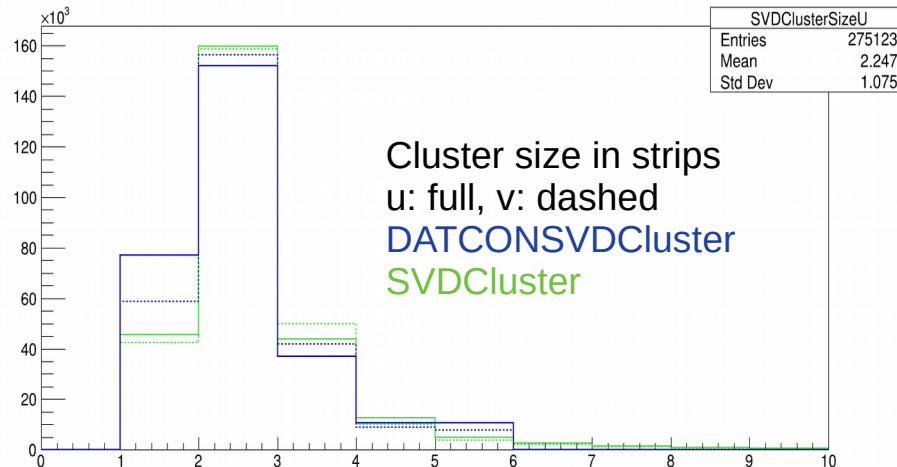
Difference in cluster number per event (DATCON-SVD)



Difference in cluster number per event (DATCON-SVD)



SVD Cluster Size U-side

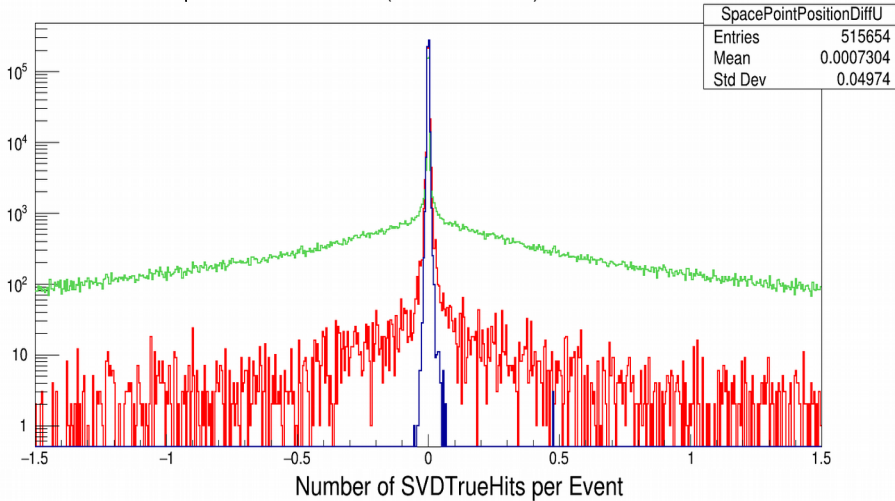


- Maximum cluster size of DATCONSVDClusters is 5 strips
- Usually more DATCONSVDClusters than SVDClusters
- Strange cases with less DATCONSVDClusters than SVDClusters (unexpected)

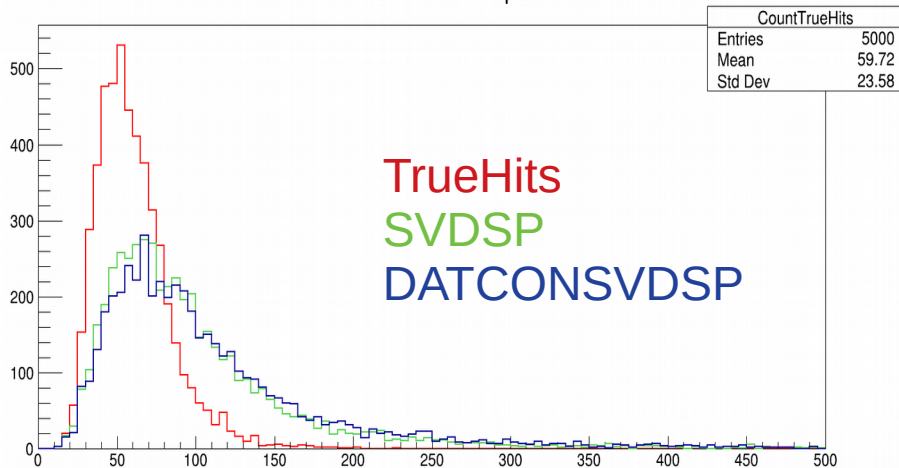
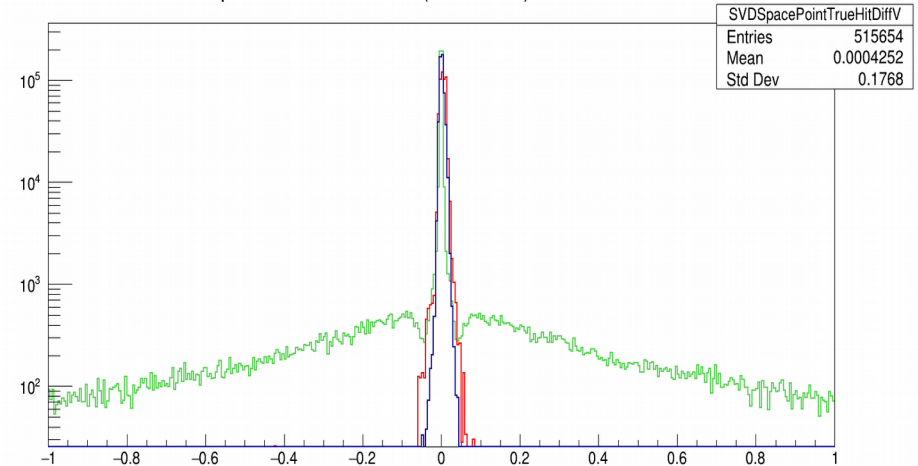
# Simulation Results - SpacePoints

Difference of SpacePoint positions (u and v) and number of SpacePoints / TrueHits per Event  
red = (SVDSP-DATCONSVDSP), green = (SVDTrueHit-SVDSP), blue = (SVDTrueHit-DATCONSVDSP)

SpacePoint difference (SVD-DATCON) of U-side in cm

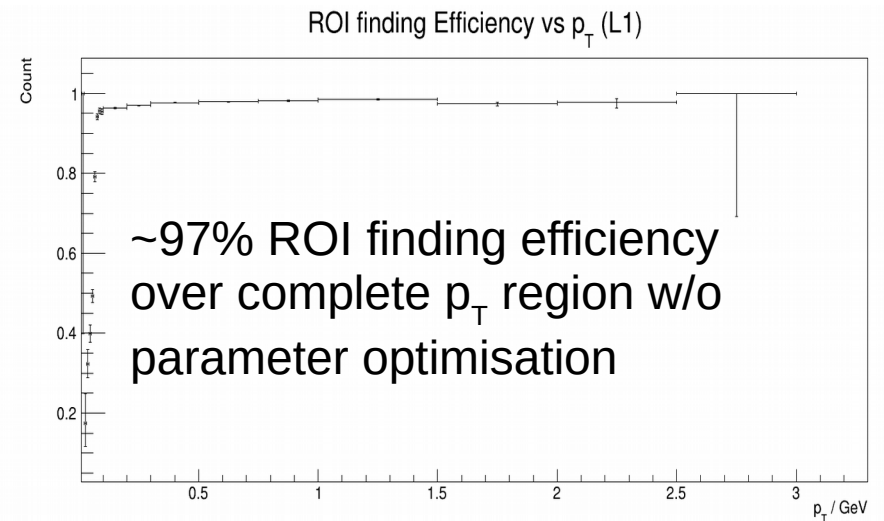
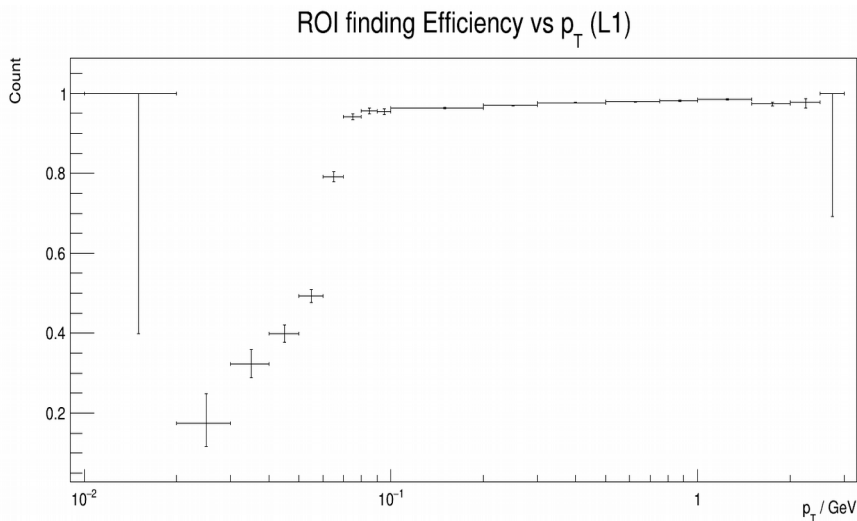
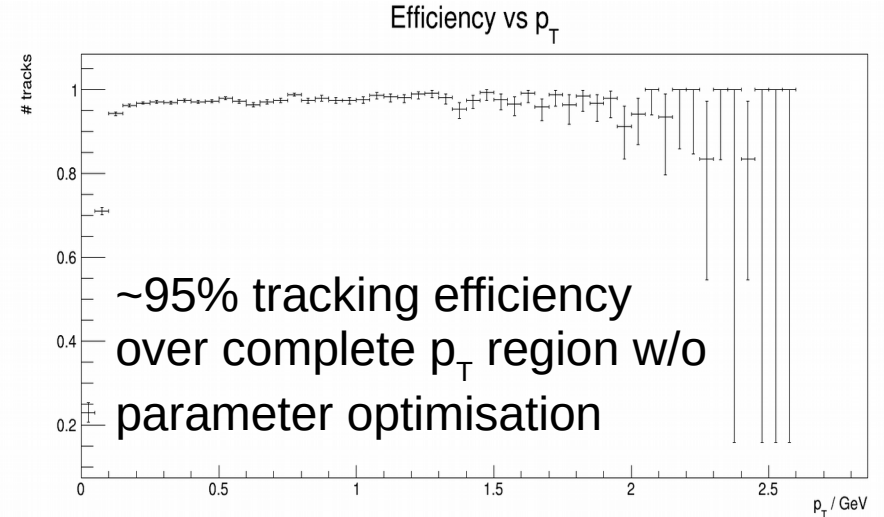
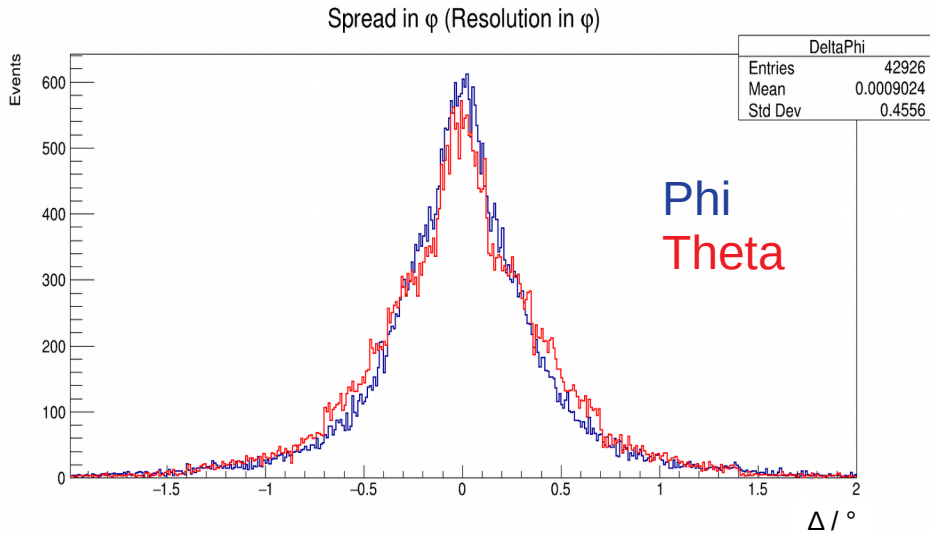


SpacePoint difference (True-SVD) of V-side in cm



- Number of DATCONSVDSpacePoints and SVDSpacePoints almost the same
- Per event more SpacePoints than SVDTrueHits

# Simulation Results – Tracking & ROI



- Very good performance of finding SVD clusters even with simplified methods for clustering and noise filtering and having no timing information
- Very good performance of SpacePoint estimation
- But: My “truth matching” might be wrong, the residuals look “too good”
  - Still: the results are fine
- Even with the new input for the Hough Trafo (DATCONSVDSpacePoints) the tracking efficiency is ~95% and the ROI finding efficiency is ~97% (without parameter optimisation)
- Simulations with real SVDSpacePoints show very similar performance for tracking and ROI creation → DATCONSVDClusters and DATCONSVDSpacePoints seem to really be compatible :)



- Simulations with background and comparison
- Optimisation of parameters!
- Still to be implemented and tested: calculation of local and global coordinates of clusters and SpacePoints without framework methods
  - Recent results look quite promising, no big performance decrease expected
  - Should only be some geometrical problems
- Save ROI with global classes implemented by Giulia → already in contact with Björn
  - Use global functions for performance estimation
- Save tracks as RecoTracks
  - Do performance studies for these RecoTracks (but ROI performance is more important)
- Clean up code and do documentation
- Remove / replace existing analysis functions
- .....



# Thank You!



**bmb+f** - Förderschwerpunkt

Elementarteilchenphysik

Großgeräte der physikalischen  
Grundlagenforschung