Status on the 3D fitter

TRG/DAQ Workshop
Jan 26 2011
Jaebak Kim(*),
Eunil Won, Byeongrok Ko, Kyungtae Kim
Korea University

Last B2GM

- Things to do
  - New basf2 CDC geometry has to be in
  - Move C++ program into integer space
  - Write VHDL
  - Use timing information to reach z0 resolution ~O(4cm)
  - Write identical C++ into tsim cdc

Last B2GM

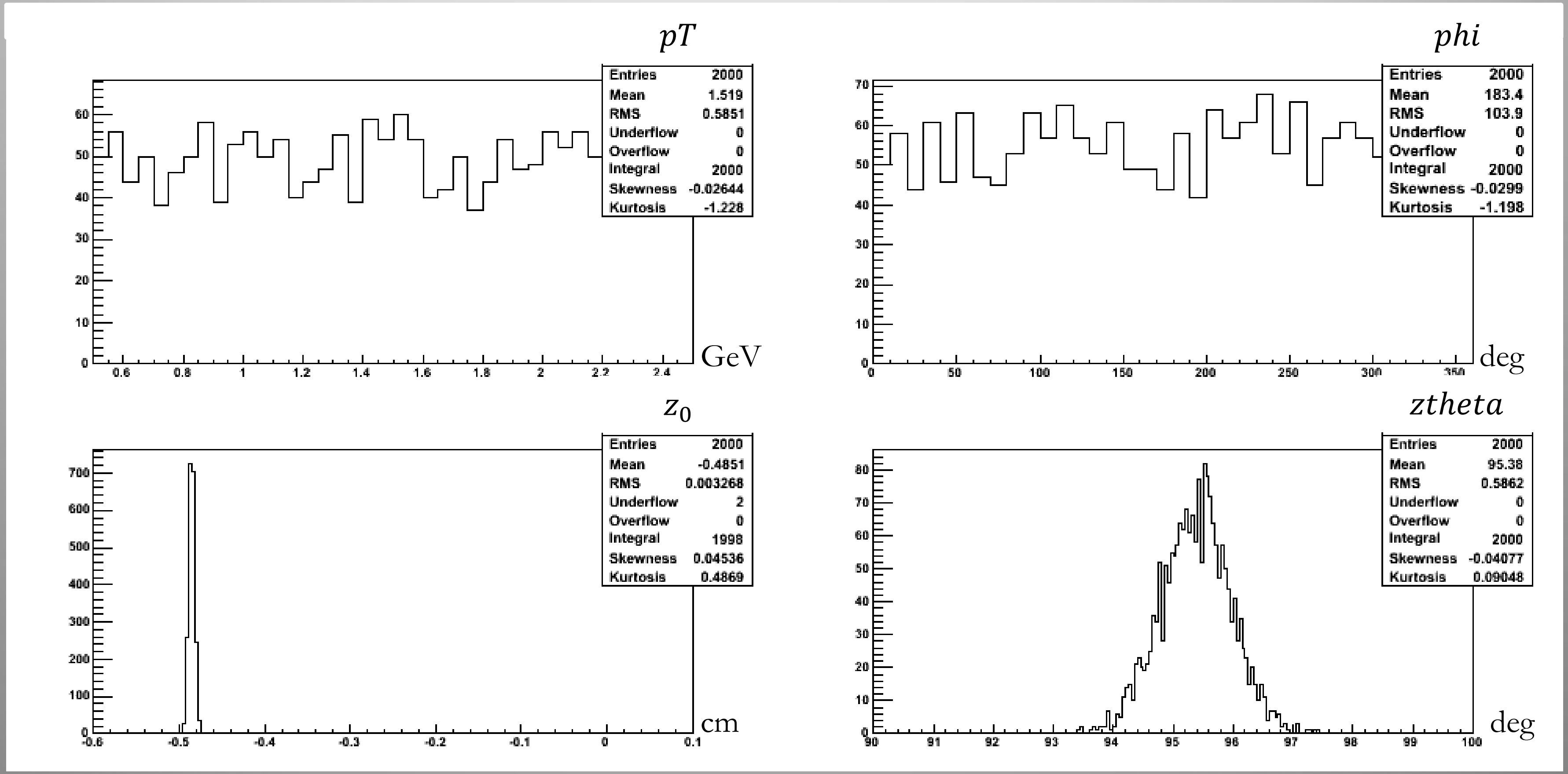
- Things to do
  - ~~New basf2 CDC geometry has to be in~~ **DONE**
  - Move C++ program into integer space **IN PROGRESS**
  - Write VHDL **IN PROGRESS**
  - Use timing information to reach z0 resolution ~O(4cm) **IN PROGRESS (KKT)**
  - Write identical C++ into tsim cdc **IN PROGRESS (KKT)**

New basf2 CDC geometry

- I calculated multiple perfect tracks using the geometry in the basf2 for the CDC given by KKT. (in order to test the 3d-fitter)
- I put the track data into the 3d-fitter to see if the fitter worked correctly with the BELLE2 geometry
  - (In doing so, I was able to improve the fitter.)
- Track info
  - p_T: [0.5~2.5]GeV
  - Track angle: [0~360]deg
  - z/r slope: Gaussian (mean: -0.09359,sig: 0.01)

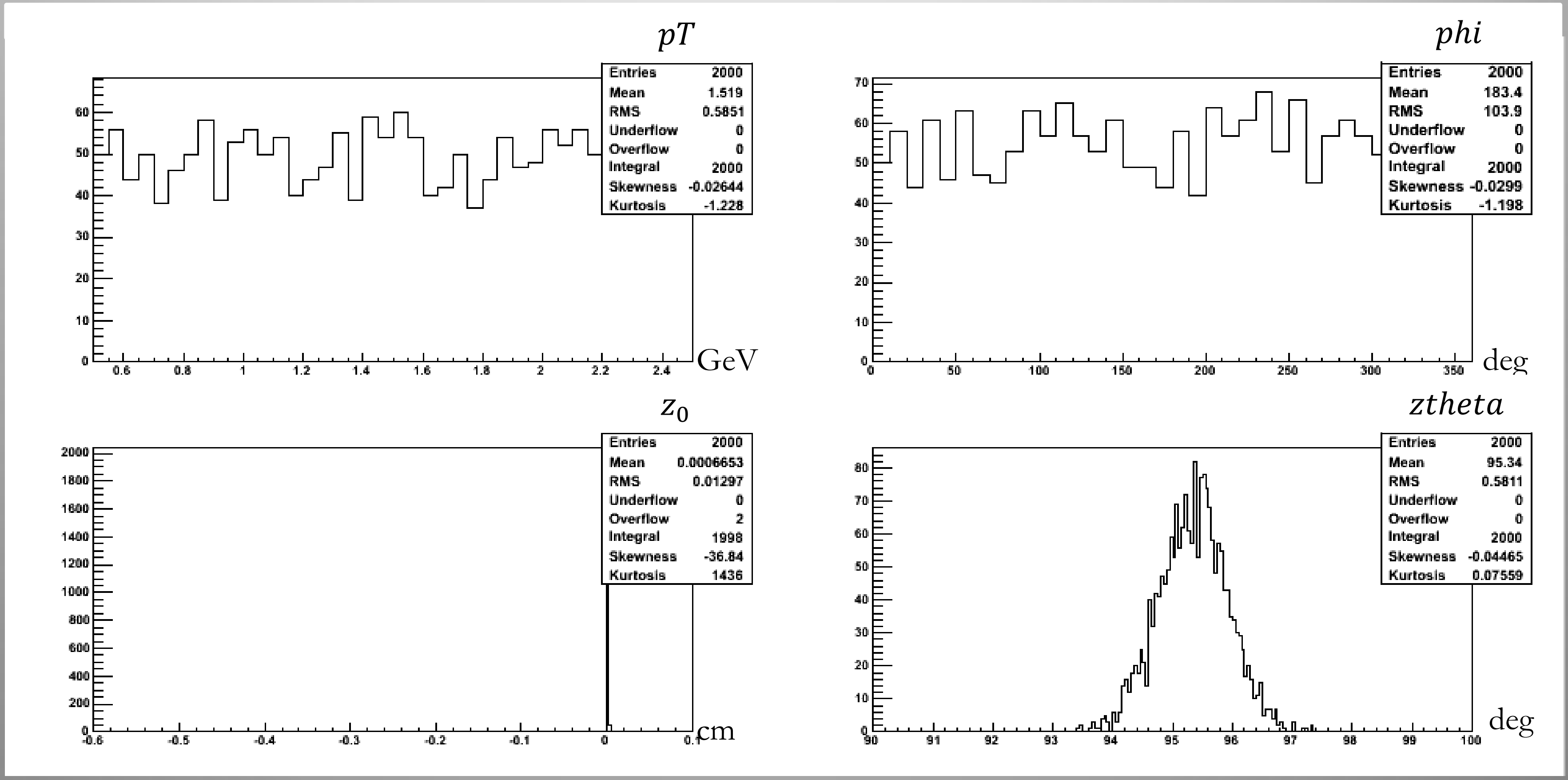
New basf2 CDC geometry(cont'd)

- Previous fitter (Results using basf2 geometry)



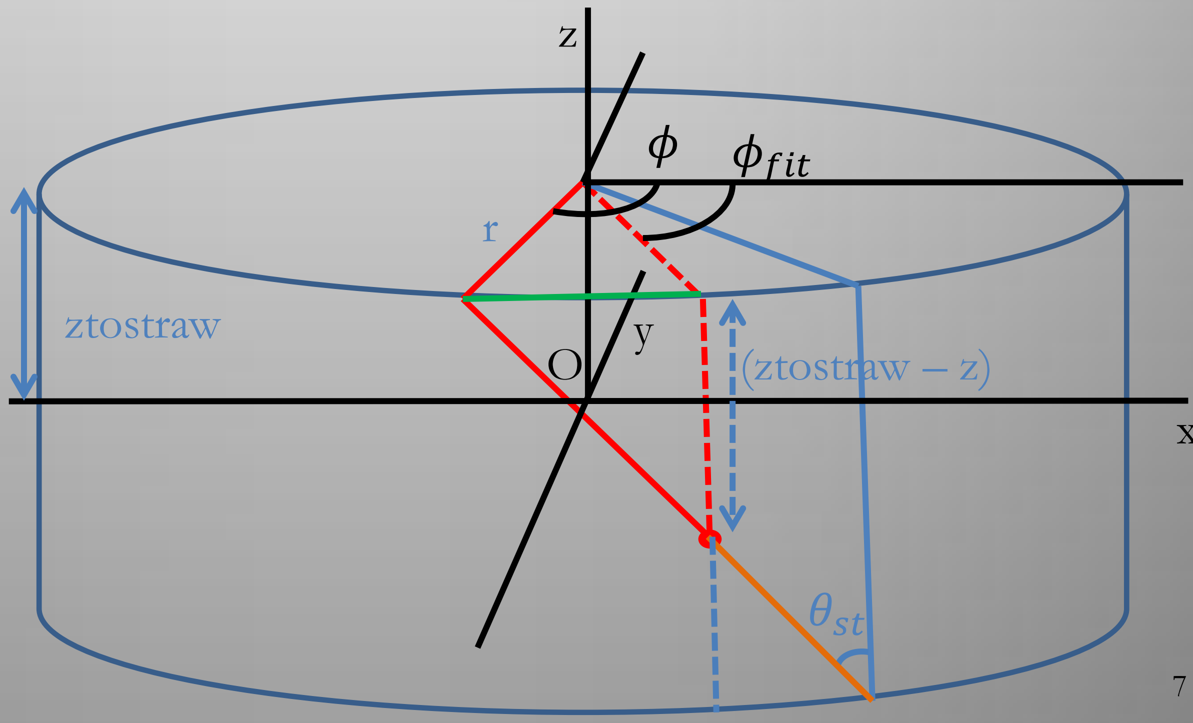
New basf2 CDC geometry(cont'd)

- Improved fitter (Results using basf2 geometry)



New basf2 CDC geometry(cont'd)

- The current code uses this relation
  - $(ztostraw - z) * tan\theta_{st} = r * \tan(|\phi_{fit} - \phi|)$
  - $(ztostraw - z) * tan\theta_{st} = r * 2 * \sin(\frac{|\phi_{fit}-\phi|}{2})$ ⇐ More Accurate



New basf2 CDC geometry(cont'd)

- We will do this test again using the data from tsim, and also when the C++ code is changed into integer space.

Move C++ to integer + VHDL

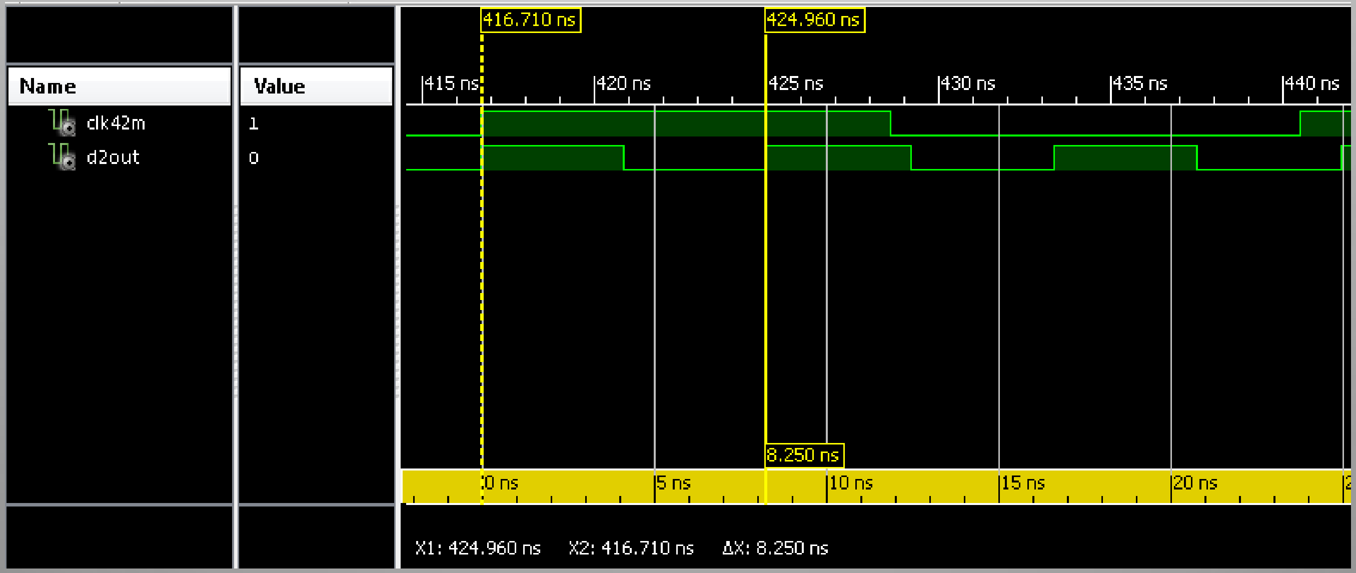
- I am planning to move the C++ code into integer space step by step, while writing VHDL for each step I go.
- In doing so, I can made sure that the C++ code and VHDL will give the same output.
- We have two C++ versions. I will change one of them to VHDL and see which one will be better later.

Move C++ to integer + VHDL (cont'd)

- I have made some toy components for the 3D fitter firmware at a 130Mhz clock on a Virtex 5 220T chip. (UT2 board)
  - Speed the 42Mhz board clock to 130Mhz using DCM (To match with other trigger logic boards)
  - An LUT(Look up table) using BRAM(Block RAM)
  - An LUT using CLB(Configurable Logic Block) (Distributed RAM)
  - Multiplication using DSP slices in Virtex chip

Move C++ to integer + VHDL (cont'd)

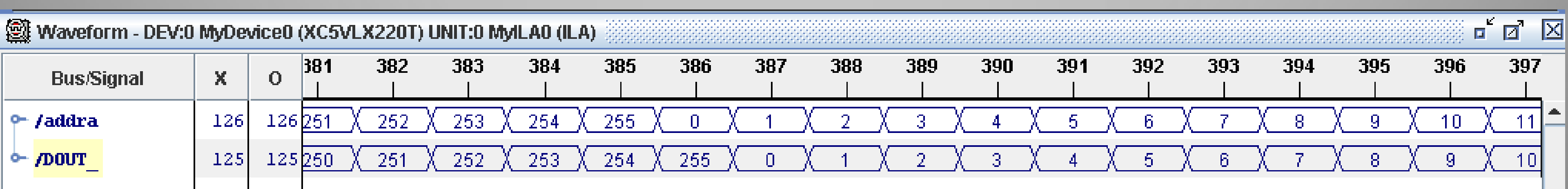
- The board clock is 42Mhz so I speeded it up to 120Mhz using DCM.
  - From 23.81ns => 42 Mhz,
  - To 8.3ns => 120 Mhz



From iSim

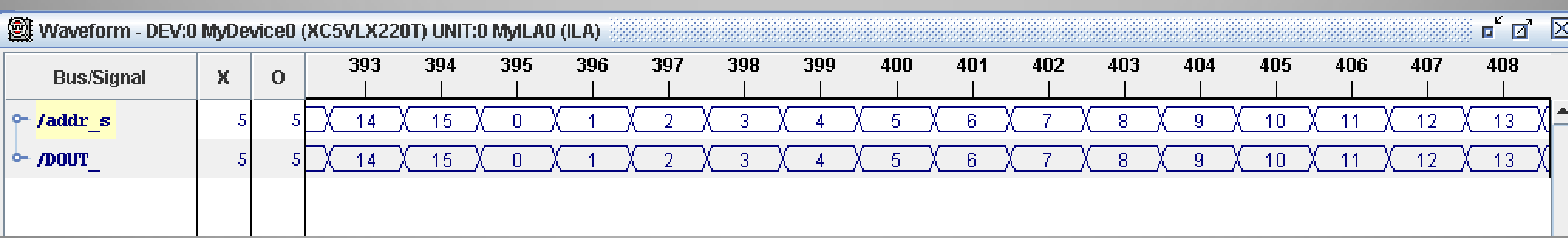
Move C++ to integer + VHDL (cont'd)

- LUT using BRAM
- These LUT's will be used for atan, cos, sin functions.



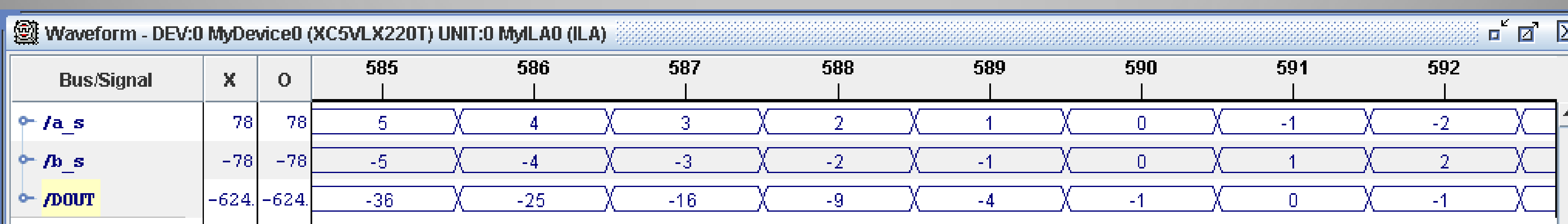
Move C++ to integer + VHDL (cont'd)

- LUT using CLB (Distributed RAM)
- These LUT will be used for storing constants that change for each super layer.



Move C++ to integer + VHDL (cont'd)

- Signed integer multiplication using DSP slices in Virtex Chip
- Multiplication can be done in one clock



Move C++ to integer + VHDL (cont'd)

- I am changing the C++ code into integer space step by step while writing the VHDL for each step.
- I will show how I transformed the phi into integer space.

Move C++ to integer + VHDL (cont'd)

- I need to change the phi input into integer space.
  - To do this I need to know the Max and Min values of the phi input.
  - I need to choose how many integers I will use.
- $\emptyset_{int} = \frac{\emptyset_{int\,max}}{\emptyset_{float\,max}} * \emptyset_{float}$
  - I round the $\emptyset_{int}$ to make them integers.
  - I am just quantizing $\emptyset_{float}$.

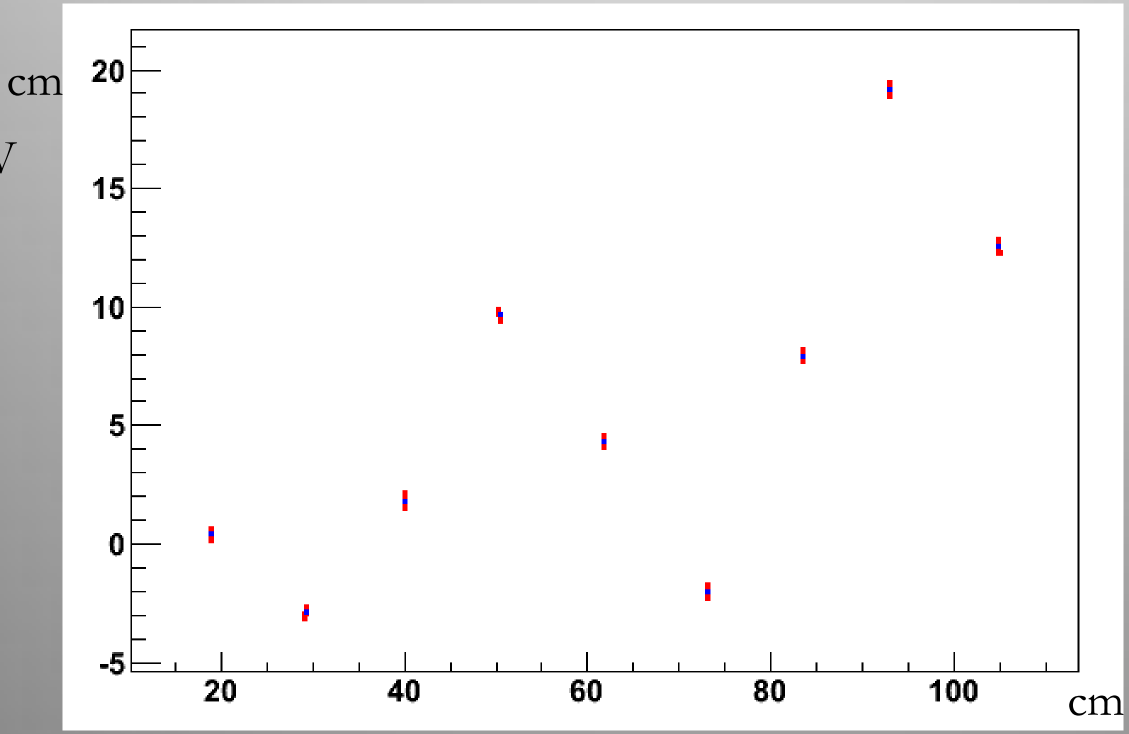
Move C++ to integer + VHDL (cont'd)

- Also, to have better resolution for the integer values, we transform $\emptyset_i$ into $\emptyset_{i\,rel} = \emptyset_i - \emptyset_{A3}$ ($\emptyset_{A3}$-Third superlayer for axial wires)
- If we do so the $\emptyset_{float\,max}$ will be smaller. Then $\emptyset_{int}$ will have better resolution.
- That is, the space between quantized $\emptyset$ will be smaller.

Use timing information to reach z0 resolution ~o(4cm)

- To estimate the needed timing information resolution for z0 resolution to be 4cm, I made an input track for the 3d trigger and smeared them 2000 times.

pT = 2 GeV



Red: Smeared track
Blue: Actual track

Use timing information to reach z0 resolution ~o(4cm) (cont'd)

- Results (when drift speed is $40\mu m/ns$)

Pt = 2 GeV/c

Time_rms(ns)	z0_rms(cm)
22.5	2.310
27.5	2.823
35	3.594
40	4.107

Pt = 0.5 GeV/c

Time_rms(ns)	z0_rms(cm)
22.5	2.349
27.5	2.871
35	3.655
40	4.177

Use timing information to reach z0 resolution ~o(4cm) (cont'd)

- The results show we need ~38ns time information resolution when the drift speed is $40\mu m/ns$
- This was done with float numbers in the fitter code, so we will do it again after the code is changed into integer space.
- Further studies on how to get the time information resolution is being done by KyungTae.

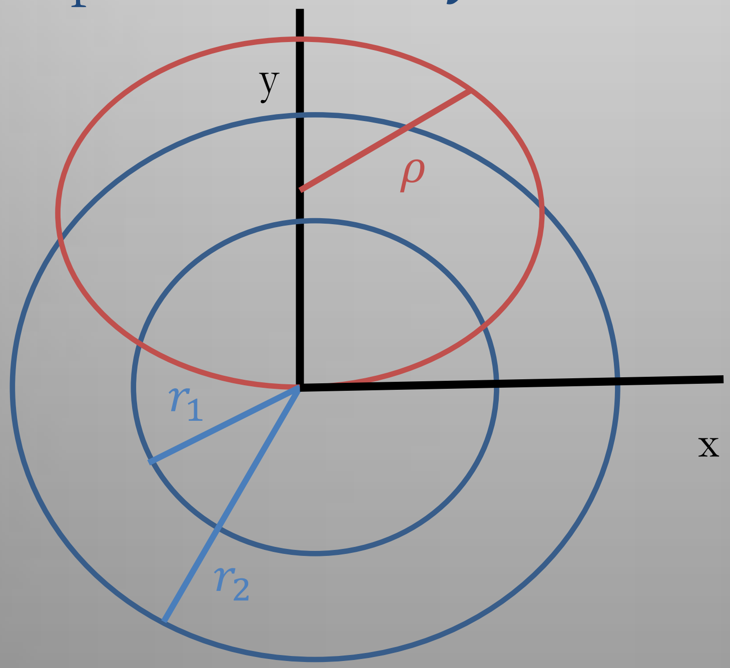
Summary

- Basf2 CDC geometry has been implemented and tested in the 3D fitter.
- We were able to improve the 3D fitter.
- We made toy VHDL modules for the fitter.
- We are changing the code into integer space.
- We were able to calculate the needed time resolution for z0 to be ~O(4cm).

Backup slides

New basf2 CDC geometry(cont'd)

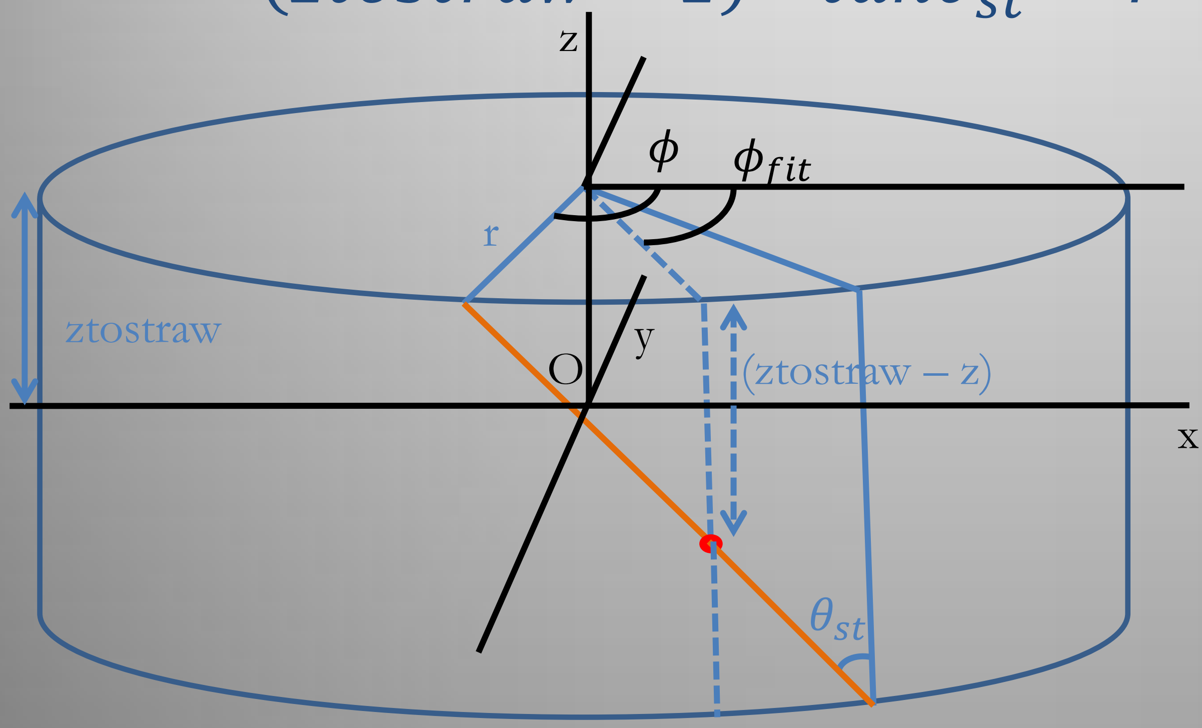
- Basic equations to get axial wire position.
  - $p_T(\frac{GeV}{c}) = 0.3 * z(e) * B(T) * \rho(m) = 0.3 * 1.5 * \rho$
  - Simple perfect track : $x^2 + (y-\rho)^2 = \rho^2$
  - CDC TS position: $x^2 + y^2 = r^2$



Blue: CDC super layers
Red: Perfect Track

New basf2 CDC geometry(cont'd)

- Equation to calculate stereo wire position
  - $(ztostraw - z) * tan\theta_{st} = r * 2 * \sin(\frac{|\phi_{fit} - \phi|}{2})$

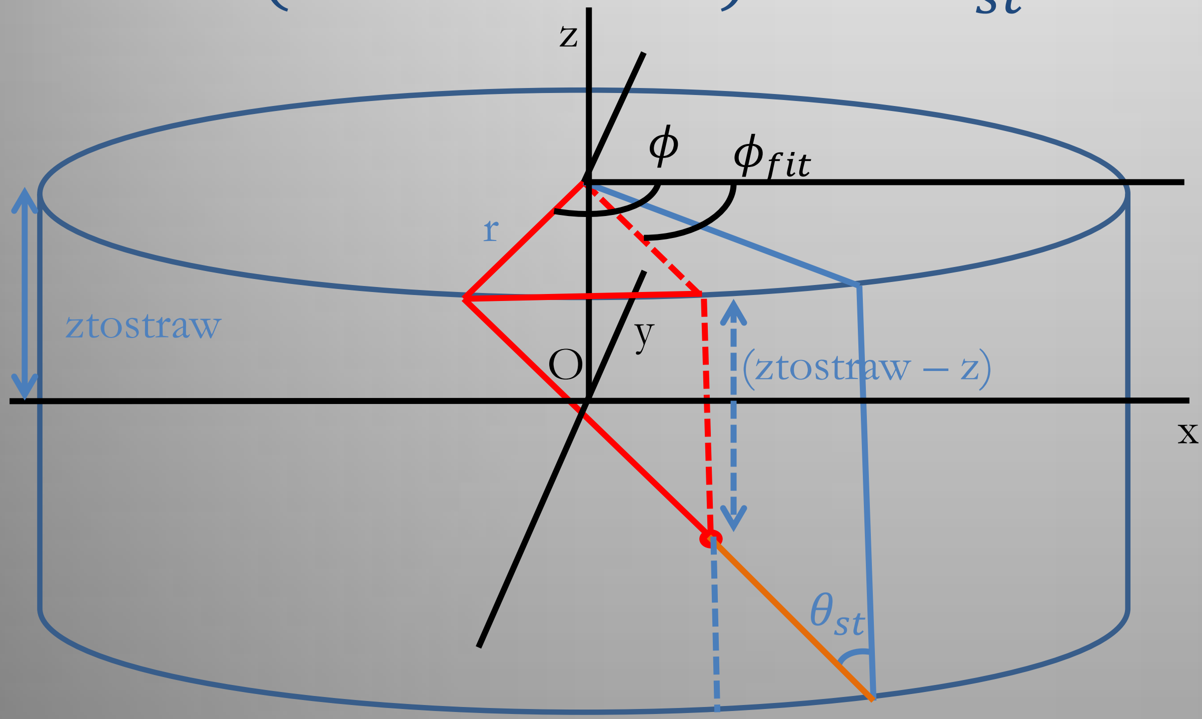


θ_{st} is greatly exaggerated

Blue Cylinder: CDC
Gold: Sense wire
Dash line: The wire if the stereo wire was an axial wire.

New basf2 CDC geometry(cont'd)

- Equation to calculate stereo wire position
  - $(ztostraw - z) * tan\theta_{st} = r * 2 * \sin(\frac{|\phi_{fit} - \phi|}{2})$

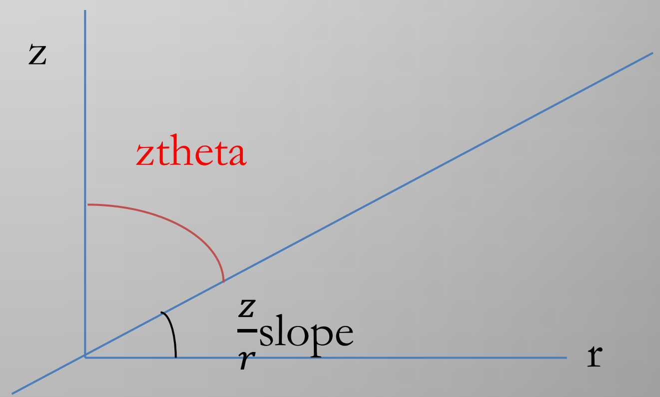


θ_{st} is greatly exaggerated

Blue Cylinder: CDC
Gold: Sense wire
Dash line: The wire if the stereo wire was an axial wire.

New basf2 CDC geometry(cont'd)

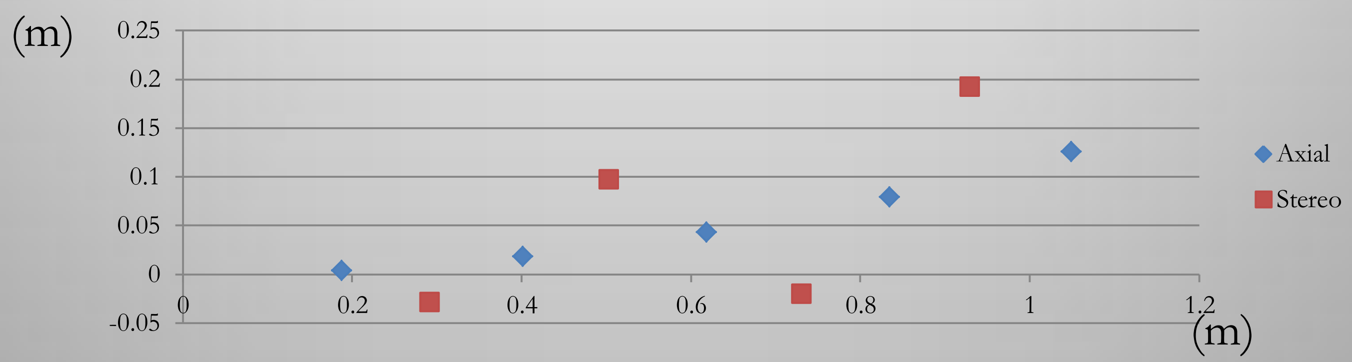
- Equation to get z.
  - Using data from the Belle tsim, we have found the relation between r and z.
  - z=-0.09358*r
  - ztheta= $\frac{\pi}{2} - \tan(\frac{z}{r} slope)$



- Now we can calculate the stereo wire position.

New basf2 CDC geometry(cont'd)

- Results
  - 2 GeV



  - 0.5 GeV