Status of Software & Simulation for CSC Trigger

Track-Finder, LCT, ORCA

D. Acosta, S.M. Wang

University of Florida

B. Tannenbaum

UCLA

Design Status

- Nearly complete conceptual design presented late March at SR/SP review
- Primary concern is the design of the Track Assembler, the heart of the Track-Finder
  - Digests extrapolation results and must determine the quantity and quality of trigger muons
  - Scheme proposed, but must be validated with physics simulation
  - Must avoid ghost tracks for multi-muon trigger, yet maintain high efficiency for high P_T single muons

Preparation for full SP review in July

- Finalize backplane and SR↔SP signals, including 4-station capability
- Determine connector space at backplane
- Estimate FPGA and RAM count
- Estimate board area (must fit on 9Ux400mm)
- Estimate cost
- Validate basic scheme with simulations

Track-Finder Simulation (Endcap)

- Written in Fortran (for the moment)
- Interfaces with the ntuple produced by CMSIM
- Simulate the Extrapolation Units of the Sector Processor (ME1-ME2, 2-3, 3-4, 1-3, 2-4)
  - ϕ_i and η_i are quantized using the appropriate number of bits of representation
  - $\mid \Delta\eta_{ij} \mid \leq 3, \quad \overline{\mathrm{AMB_i} \cdot \mathrm{AMB_j}} = 1$ ... to reject halo muons
    ($\Delta\eta = \eta_i - \eta_j, \quad$ AMB : Accelerator Muon Bit)
  - Comparators to determine low, medium, high Pt tracks (for ME1-ME2, 1-3 only)
  - Quality of extrapolation :
    - For ME1-ME2 and 1-3, the qualities are based on the determined Pt
    - For ME2-ME3, 3-4, 2-4, the qualities are based on the number of cathode strip layers with hits
- Simulate the Track Assembler Unit (Stream 1)
  - Assemble extrapolations into complete tracks
  - Output 3 best tracks

Results Single μ events :

Efficiency of Extrapolations :

Pt = 3 GeV

		%	
		ME1-ME2	ME1-ME3
successful extrapolation	=	93.4	81.3
Low Pt	=	77.1	58.1
Medium Pt	=	14.5	19.5
High Pt	=	1.8	3.7

Pt = 5 GeV

		%	
		ME1-ME2	ME1-ME3
successful extrapolation	=	99.1	98.1
Low Pt	=	44.8	46.4
Medium Pt	=	53.2	46.8
High Pt	=	1.2	4.9

Pt = 50 GeV

		%	
		ME1-ME2	ME1-ME3
successful extrapolation	=	99.4	99.4
Low Pt	=	0.02	0.01
Medium Pt	=	0.9	1.1
High Pt	=	98.4	98.2

Pt (GeV)

Low $\leftarrow$ 5 Medium 25 $\rightarrow$ High

Efficiency of complete track found at TAU (stream 1) :

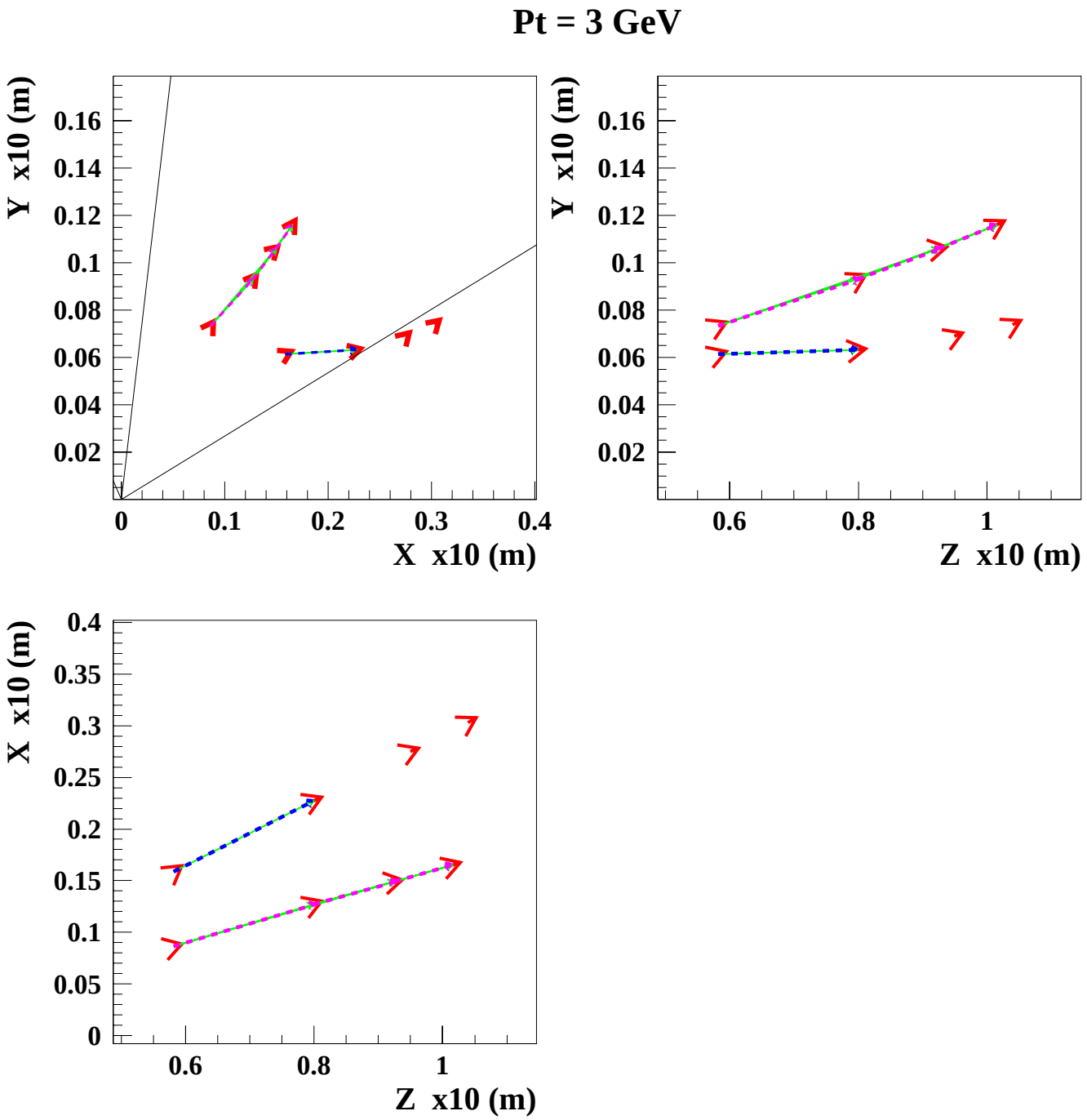
Pt = 50 GeV

0 track	=	0.6 %
1 track	=	93.9 %
2 tracks	=	5.2 %
3 tracks	=	0.3 %

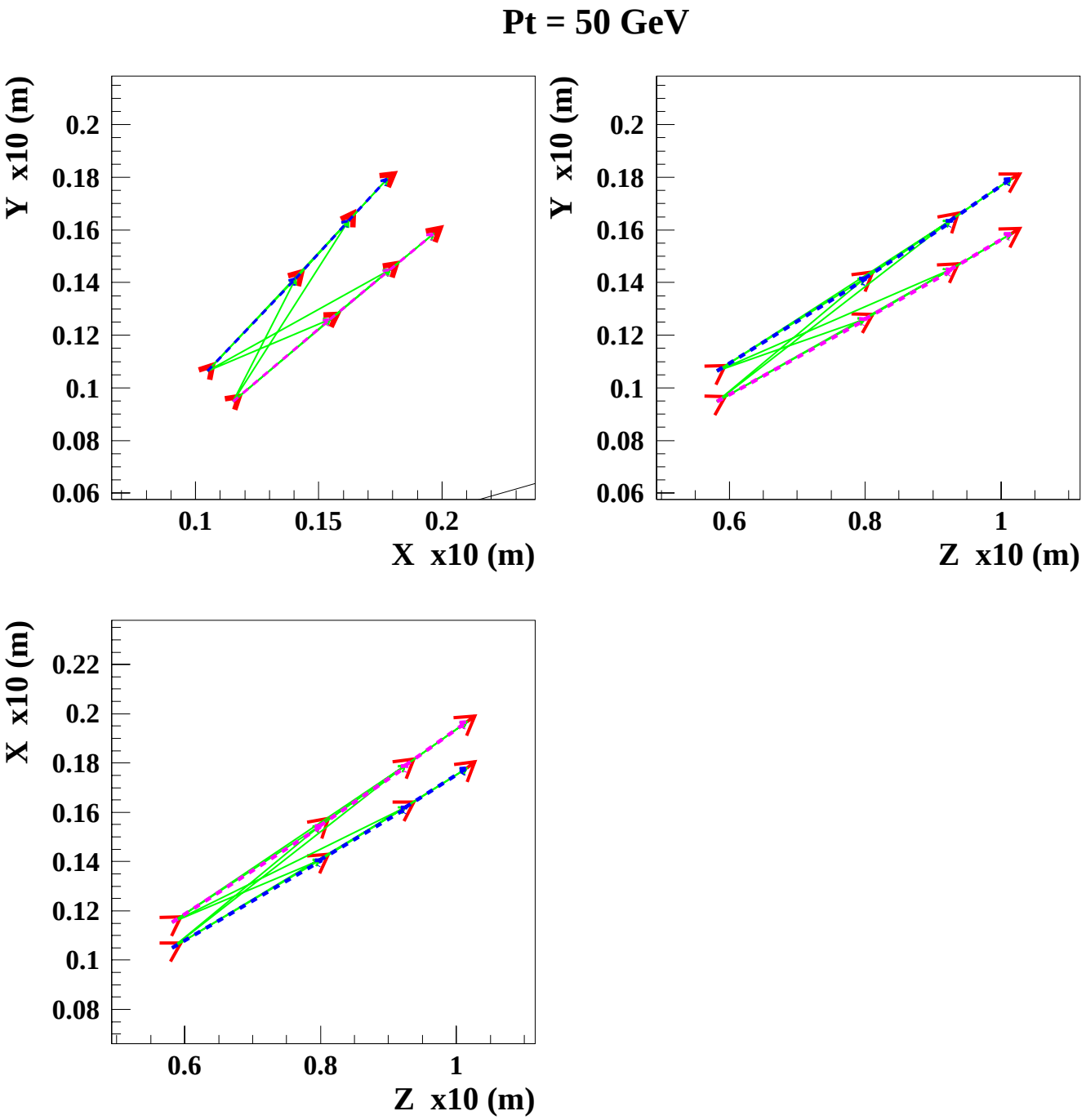
Results Single halo-μ events :

P = 100 GeV (Total # Evts = 9142)		%
successful extrapolation	=	0.01
Low Pt	=	0.00
Medium Pt	=	0.00
High Pt	=	0.01

2 μ in 60° Sector (Pt = 3 GeV)



2 μ in 60° Sector (Pt = 50 GeV)



LCT Simulation studies and status

Darin Acosta for

Benn Tannenbaum

UCLA

May 1999

Outline

- **Software Status**
- **Simulation progress**
- **Plans**

Software Status

- **The new LCT simulation software exists as a tarball for download and installation. Instructions are at www–collider.physics.ucla.edu/cms/cmsim/**
  - **Follow the link to 'New LCT software'**
  - **Includes preliminary write up by Igor Smirnov**
  - **Describes all common blocks used and word format**
- **Features include**
  - **Time based simulation of**
    - **Comparator**
    - **Anode and Cathode LCT boards**
    - **Motherboard**
    - **Muon port card**
  - **Finds LCTs**
  - **Generates pattern tables from CMSIM**
  - **Excellent clock-cycle by clock-cycle event display for debugging**

Software Status

- The output common block has all the information described in the trigger bit note (ftp://hepsun0.physics.ucla.edu/pub/cms/trigger/triggerbits.ps)
- 28-bit cathode word
  - Valid pattern flag (1 bit)
  - Pattern number (8 bits)
  - Left/Right Bend (1 bit)
  - 1/2 strip ID (8 bits)
  - BXN (5 bits)
- 28-bit anode LCT word
  - Valid pattern flag (1 bit)
  - Pattern quality (2 bits)
  - Accelerator muon (1 bit)
  - Wire gang ID (7 bits)
  - BXN (5 bits)

Software Status

- 44-bit MPC word
  - Valid pattern flag (1 bit)
  - Cathode pattern number (8 bits)
  - Cathode left/right bend (1 bit)
  - Cathode 1/2 strip ID (8 bits)
  - Anode pattern quality (2 bits)
  - Accelerator muon (1 bit)
  - Anode wire gang ID (7 bits)
  - CSC ID (4 bits)
  - BXN match (2 bits)
  - Anode bunch crossing number (5 bits)
  - TMB Status A (2 bits)
  - TMB Status B (2 bits)
  - Sync error (1 bit)

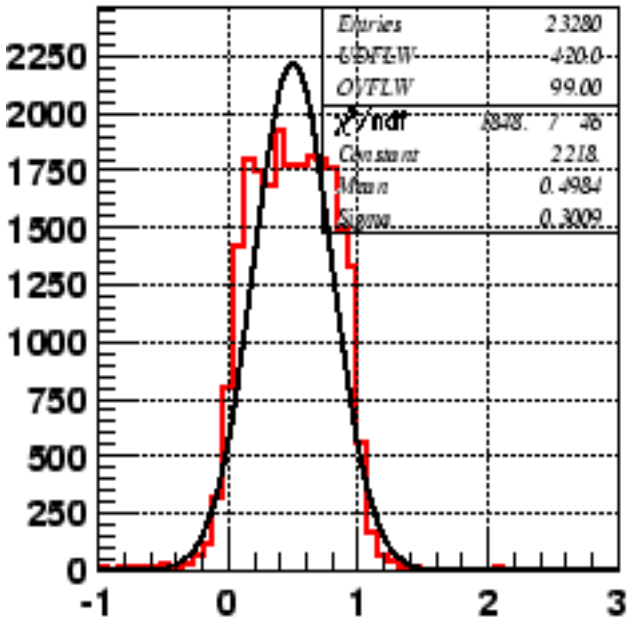
- **Has been installed at UCLA and UFL. Once it's used by a few more people, I'll release it to CERN.**

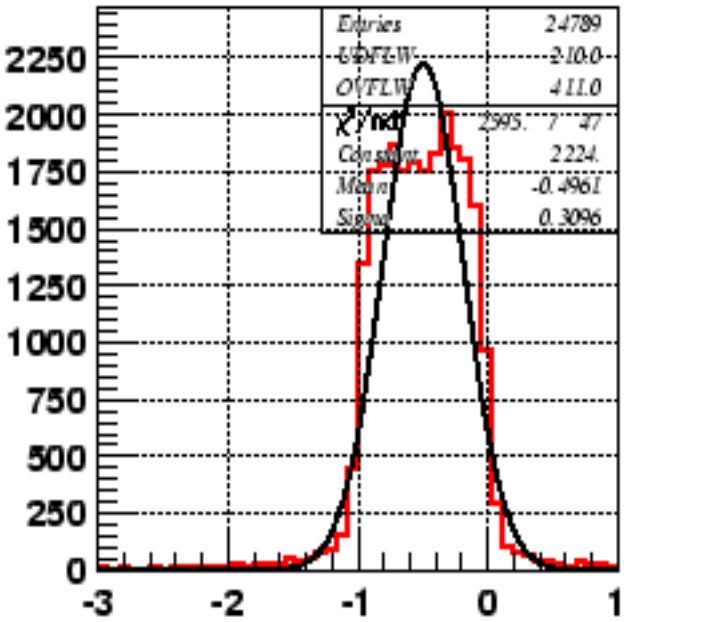
Simulation Status

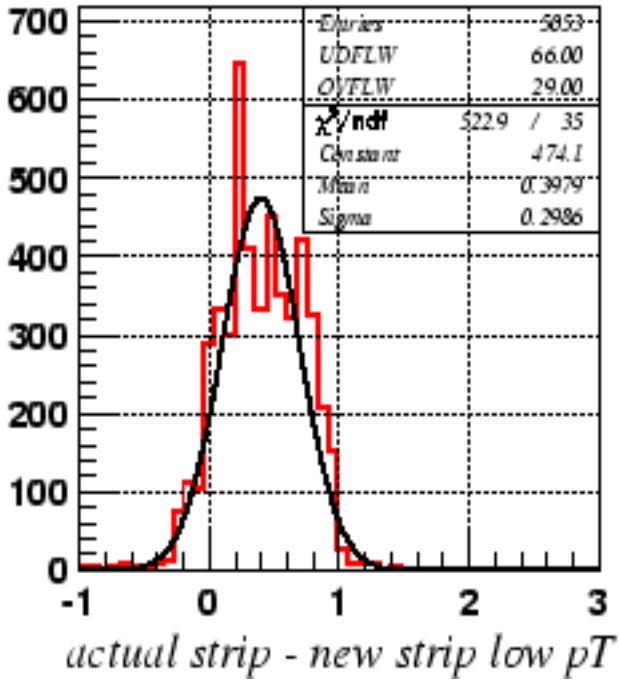
- **Can generate pattern tables with 5 000 - 10 000 events (about 2-3 days of running at U. Florida)**
- **Can generate 1000 single muon events for study overnight**
- **At present the LCT finding efficiency is 90%.**
- **Am studying resolution of LCT finding**
  - **Compare exact location of muon passing through a given station to that found by the LCT simulation on layer 3**

Strip Resolution

2.5-25 GeV/c muons





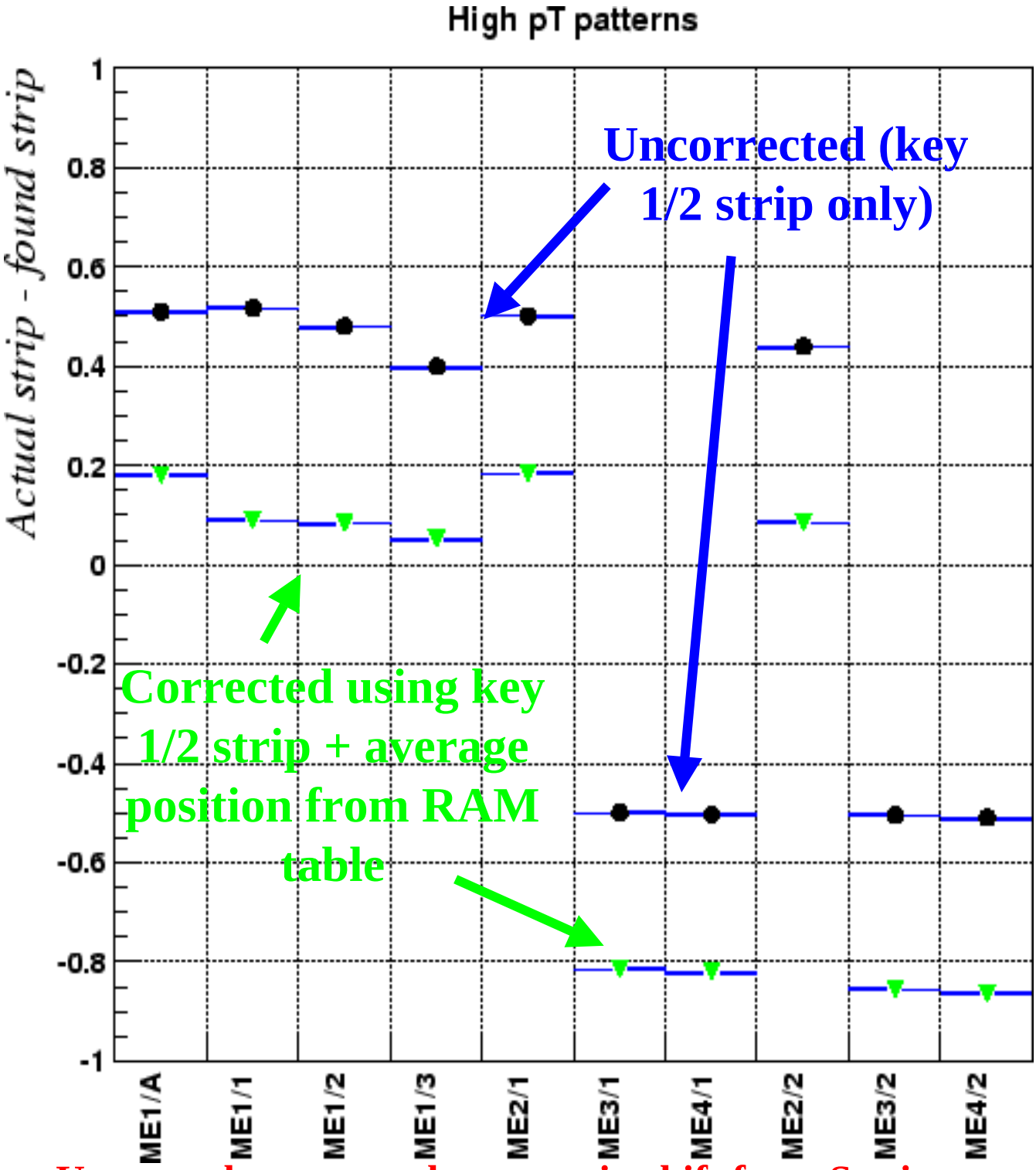


Calculated by taking the position of the muon in the chamber on layer 3 minus the key strip returned by the LCT simulation.

Use 1/2 strips for high p_T, di-strips for low p_T.

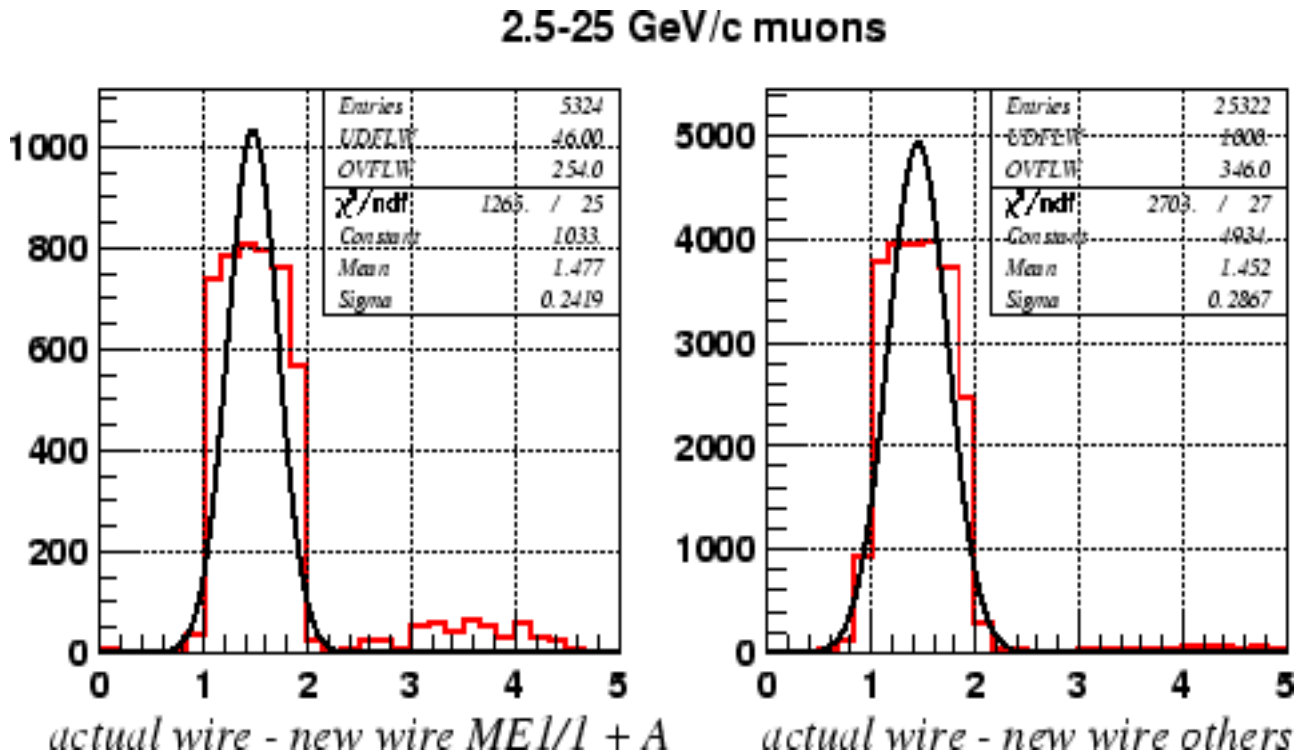
Naively expect $\sigma = 1/\sqrt{12} \cong 0.29$

Strip Resolution



Unsure what causes the one strip shift from Stations ME1 & 2 and ME3 & 4. Probably caused by switched orientation of the chambers.

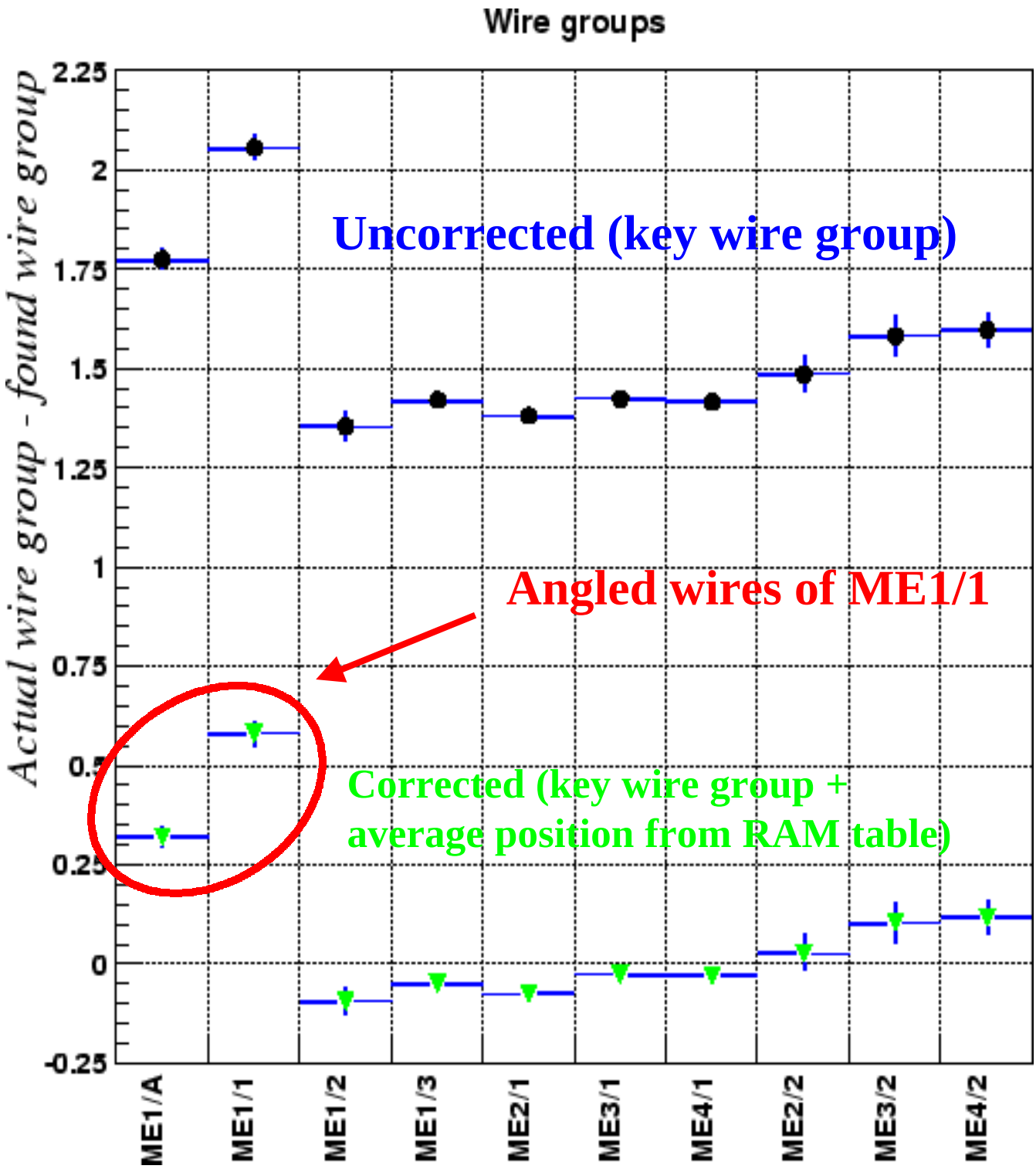
Wire Resolution



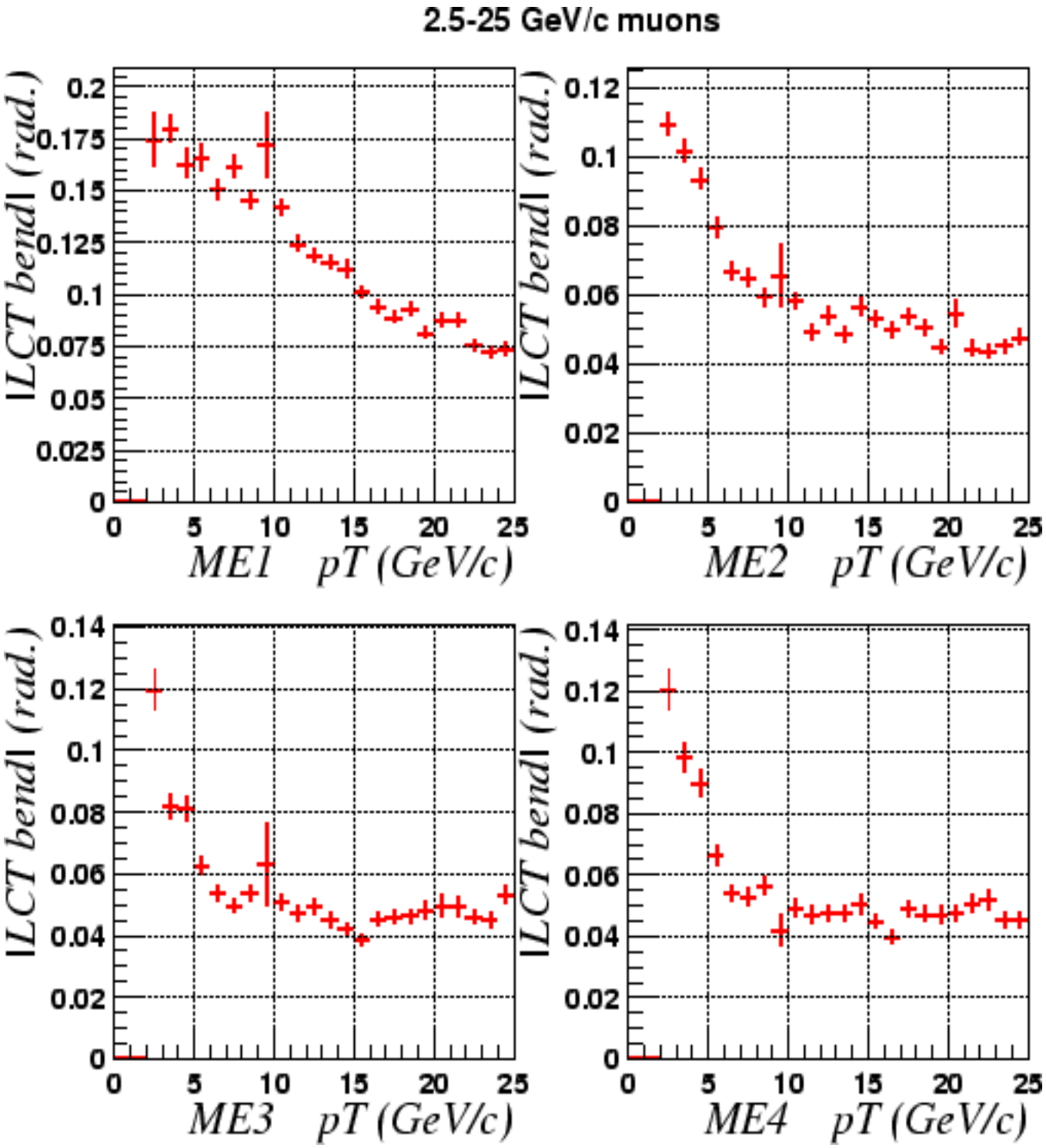
Calculated by subtracting the key wire group returned by the LCT from the actual wire struck by the muon.

Naively expect $\sigma = 1/\sqrt{12} \cong 0.29$

Wire Resolution



Local bend of track



Remaining issues

- **Efficiency is about 10% low.**
- **Station ME1 & 2 vs Station ME3 & 4 shift**
- **Angled wires of ME1/1 (not sure what can do, but not really an issue?)**

Plans

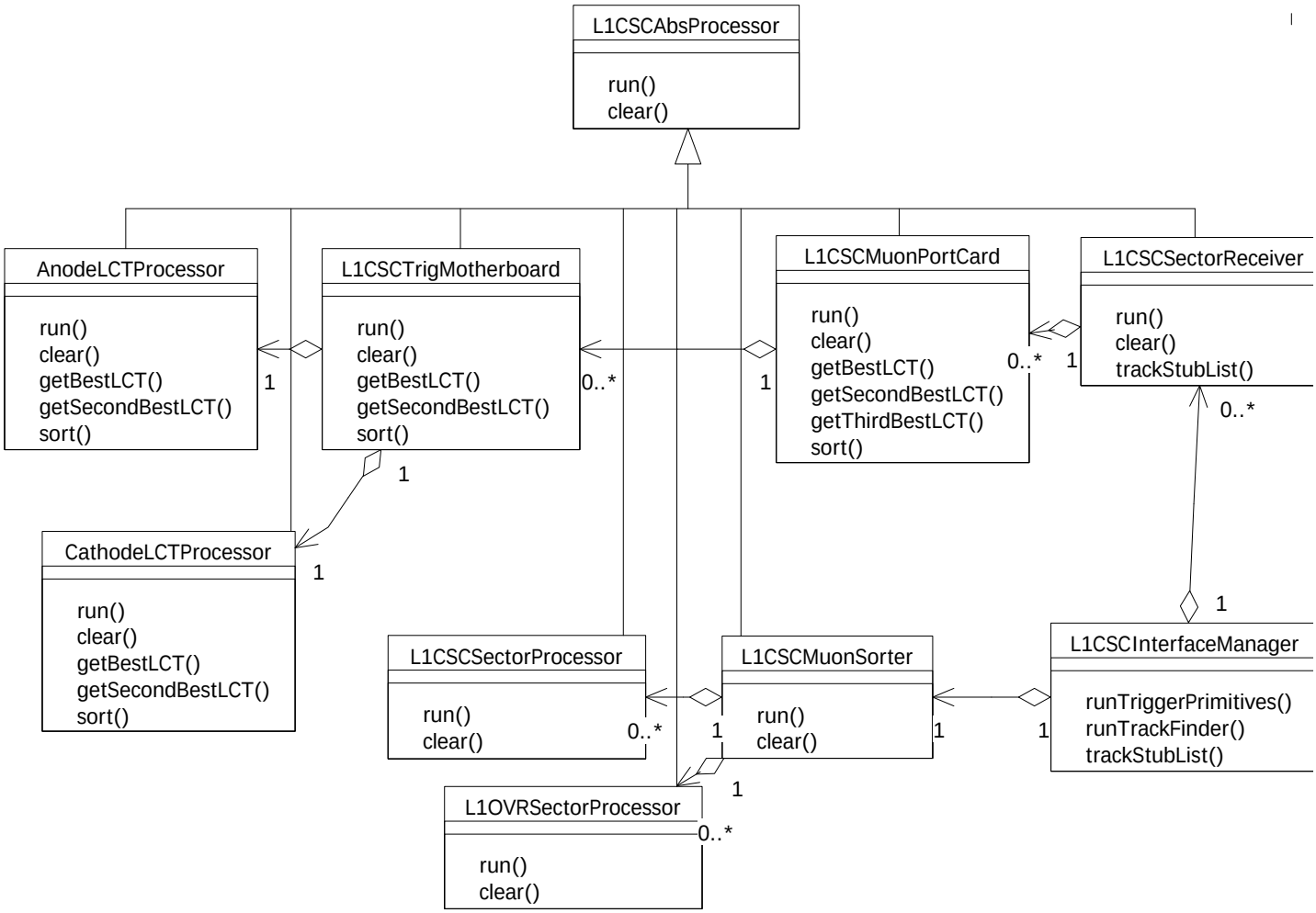
- **Release this software to repository as soon as possible**
- **Move to ORCA world**
  - **Translate existing code to C++**
  - **Add in sector receiver simulation (being written by T. Truong).**
  - **Intro to Objects/Intro to UML/Intro to C++ classes at FNAL this summer. Please join me so they don't get cancelled again!**
- **Simulate, simulate, simulate!**
- **High level trigger milestone in late Fall.**

EMU Trigger Code Development for ORCA

- Work started by Acosta, Avery, Wilkinson
- Creation of C++ objects from trigger block diagram
- Skeleton classes with minimal functionality
- Sophistication to be added after framework created
  - Incorporate existing C++ code for LCT pattern recognition
  - Write Track-Finder algorithms
- Need to interface with CSC geometry and digitization code being written

CSC and DT interface

- Code for DT trigger primitives recently installed into ORCA by Claudio Grandi
  - Hooks exist for CSC trigger
  - DT class organization seems unduly complicated
  - Start from scratch with CSC classes which mimic hardware functionality and hierarchy
- Must take care to run both CSC and DT trigger primitive processors before Track-Finders
  - Must split trigger path in software at the Sector Receiver


L1CSCAbsProcessor
run()
clear()
AnodeLCTProcessor
run()
clear()
getBestLCT()
getSecondBestLCT()
sort()
L1CSCTrigMotherboard
run()
clear()
getBestLCT()
getSecondBestLCT()
sort()
L1CSCMuonPortCard
run()
clear()
getBestLCT()
getSecondBestLCT()
getThirdBestLCT()
sort()
L1CSCSectorReceiver
run()
clear()
trackStubList()
CathodeLCTProcessor
run()
clear()
getBestLCT()
getSecondBestLCT()
sort()
L1CSCSectorProcessor
run()
clear()
L1CSCMuonSorter
run()
clear()
L1CSCInterfaceManager
runTriggerPrimitives()
runTrackFinder()
trackStubList()
L1OVRSectorProcessor
run()
clear()
1
0..*
1
0..*
1
0..*
1
1
0..*
1
1
1
1
0..*