Update on Micromegas in Marlin

Peter Hayman

Carleton

Table of Contents

- **Part I: ILCSoft v01-13-01 (carleton-dev)**
  - Overview of the Code
  - Max Amplitude Resolution Comparison
  - Preliminary Reintegration Resolution Comparison

- **Part II: ILCSoft v01-17-02 (carleton-dev-new)**
  - Overview of the Code
  - Preliminary Distortion Results
  (COMING SOON)

- **Part III: Future Direction**

Part I: Overview of Code

- Pulse Finding:
  - **MMPulseFinderProcessor**

- Hit Finding:
  - **MMHitFinderProcessor**

  (uses PadResponseFunction)

- Track Finding/Fitting
  - **TrackMakingKalmanFitterProcessor (custom)**

- PRF Calibration
  - **MMCalibrationProcessor**

- Bias Correction
  - **MMBiasUtilityProcessor**

Part I: Overview of Code (cont.)

- No z-resolution was calculated with this code, but in preparation for it (and for reintegration), the processor MMHitTimeCorrectionProcessor was introduced.

- Reintegration was introduced with the (very preliminary) processor ReintegrationProcessor (not on SVN).

Part I: Overview of Code (cont.)

- Simple Max-Bin Procedure (N.B., excluding steps to convert raw data and map channels):

  - **First Run:**
    - MMPulseFinderProcessor
    - MMHitFinderProcessor
    - TrackMakingKalmanFilterProcessor
    - MMCalibrationProcessor

  - **Extract PRF Parameters from Calibration Processor output with custom Root script**

  - **Second Run (Using Pulses saved to LCIO)**
    - MMHitFinderProcessor
    - TrackMakingKalmanFilterProcessor

  - **Third Run (Using Pulses and Hits saved to LCIO)**
    - MMBiasUtilityProcessor
    - TrackMakingKalmanFilterProcessor

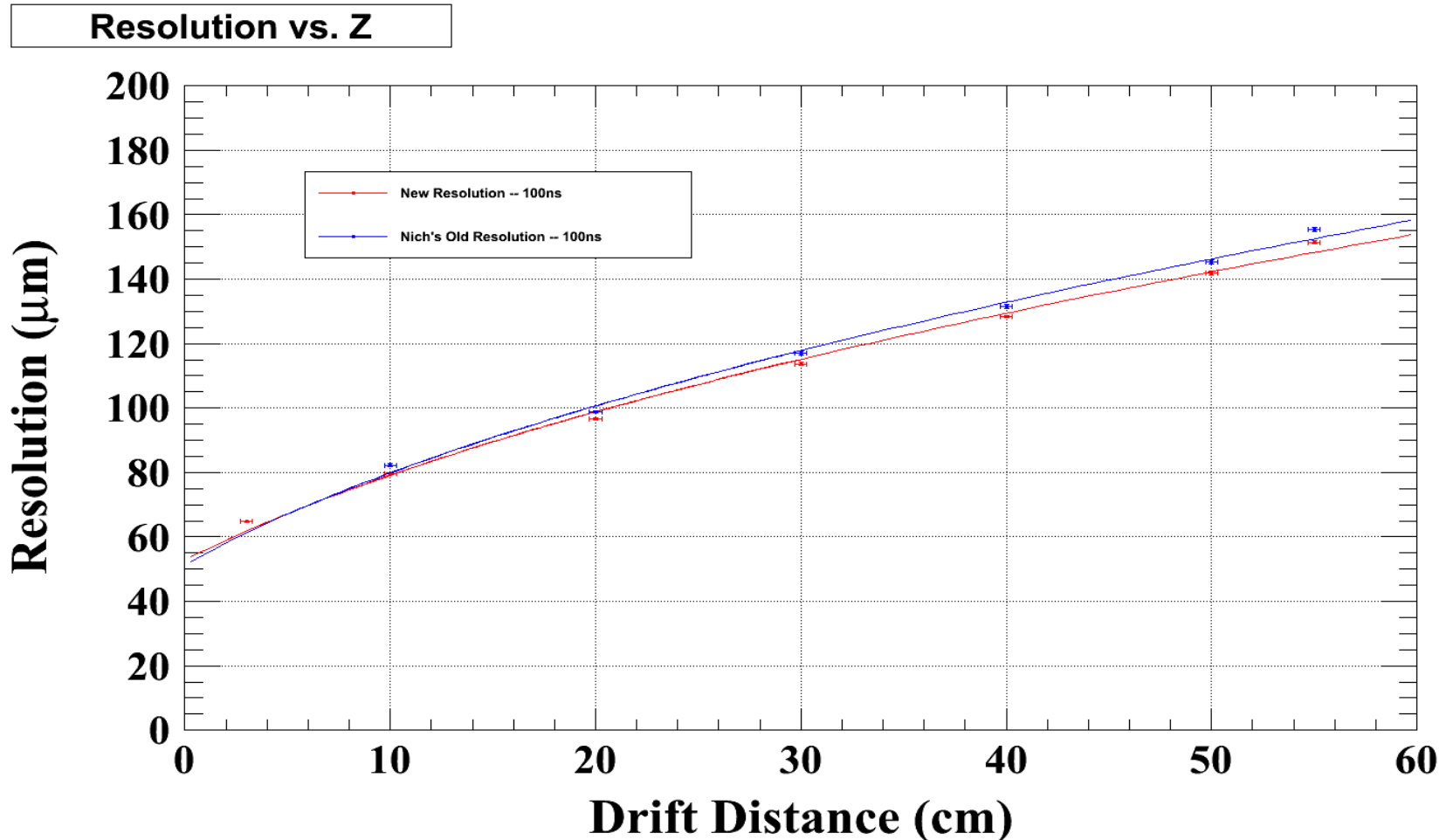 - **Resolution calculated with custom Root script**

Part I: Overview of Code (cont.)

- Reintegration Procedure (N.B., excluding steps to convert raw data and map channels):
    - **First Run:**
        - MMPulseFinderProcessor (no threshold cut; simple max bin time; channel-based pedestal subtraction)
        - MMHitFinderProcessor (no PRF fit, just used to group pulses)
        - MMHitTimeCorrectionProcessor
        - ReintegrationProcessor
        - MMHitFinderProcessor (with PRF fit)
    - **Then follow the same procedure as with the max bin method**

Part I: Max Amplitude Results
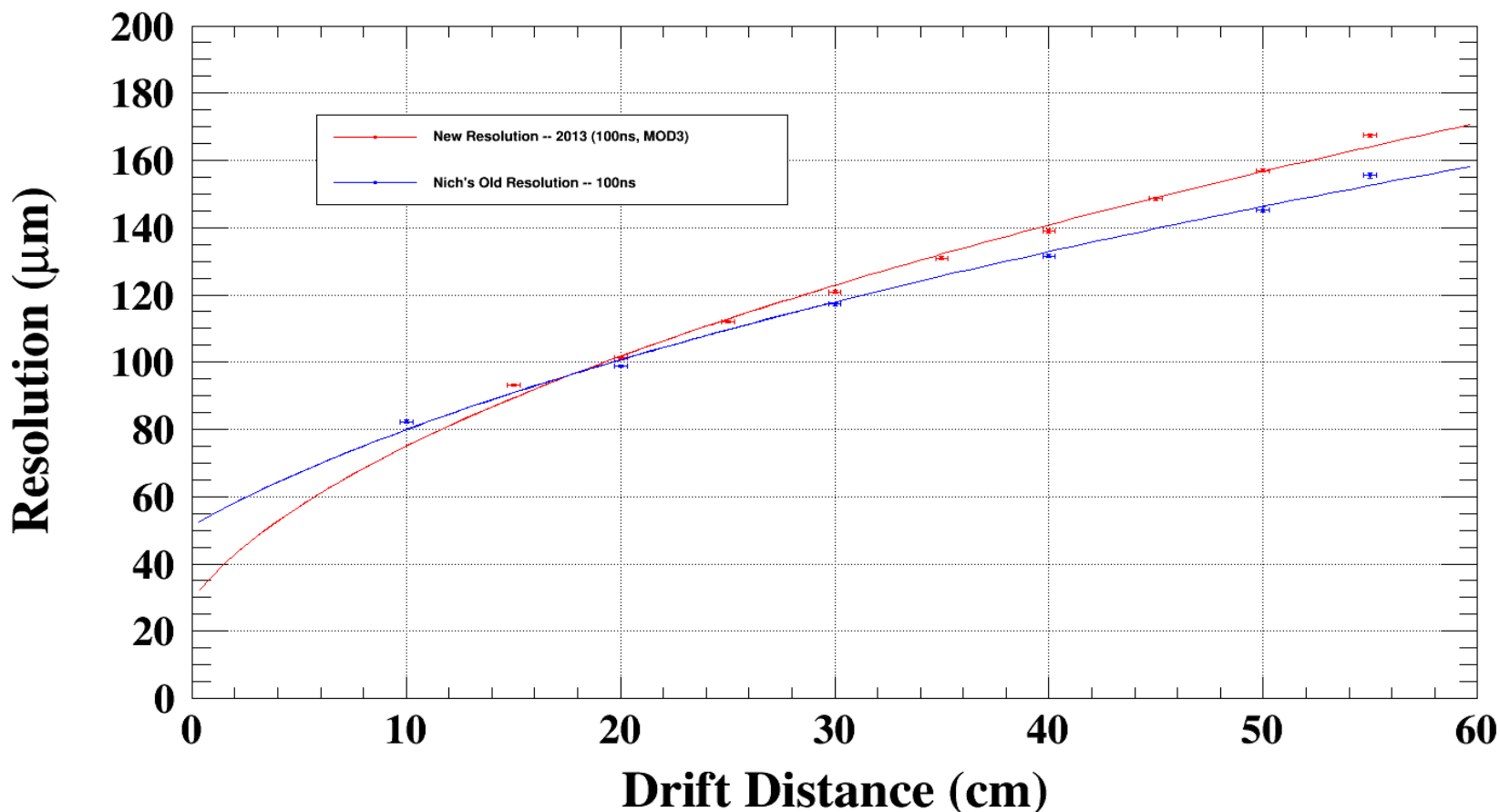
- Results 2011 (100ns Peaking Time):

Resolution vs. Z

- Results 2013 (100ns Peaking Time):

Resolution vs. Z
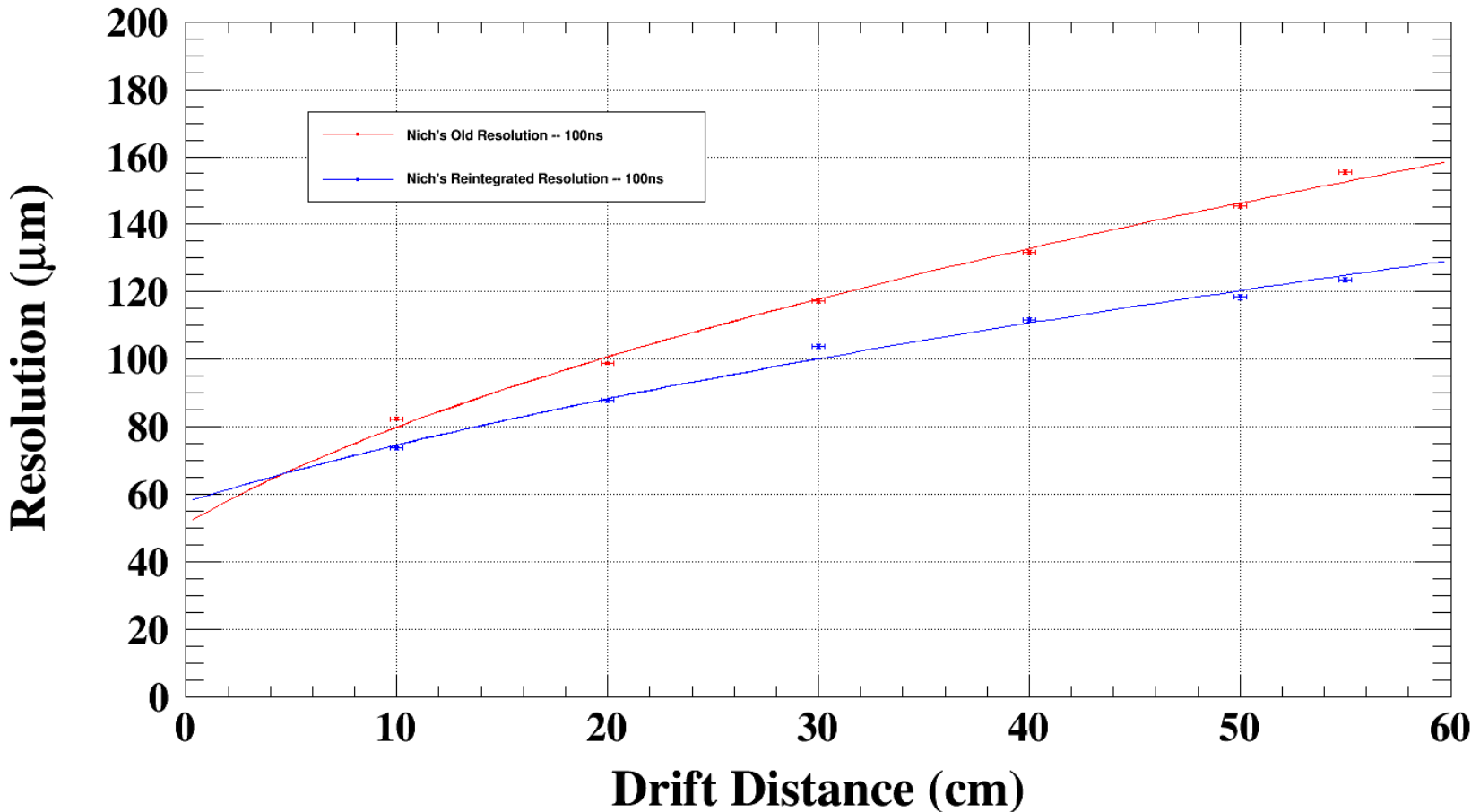
- For Reference:

Resolution vs. Z

- Results 2011 (100ns, Reintegration):

Resolution vs. Z

2011 Reintegrated Resolution (100ns)

Nich's Reintegrated Resolution -- 100ns
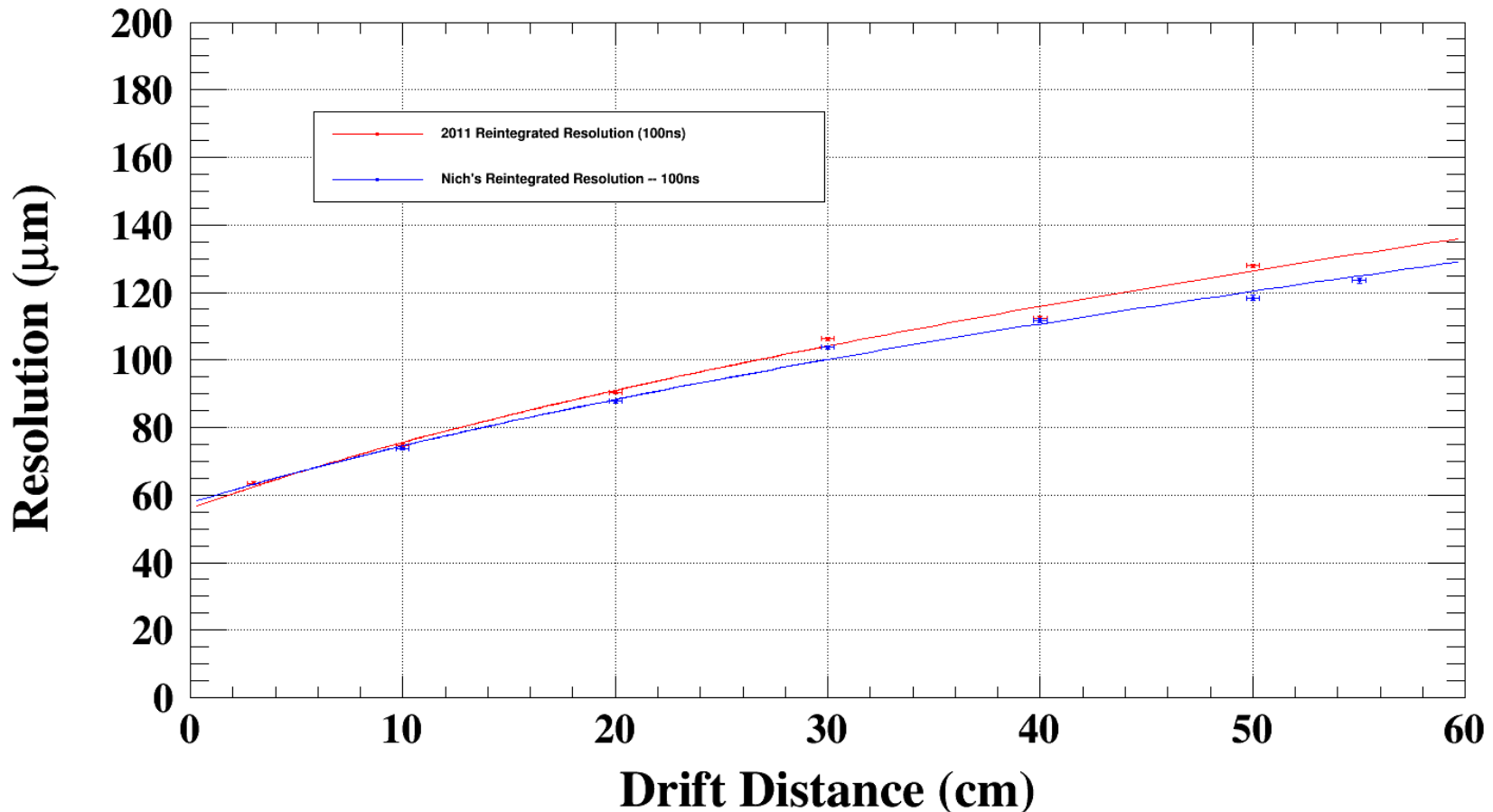
Part I: Reintegration Results

- Results 2013 (100ns, Reintegration):

Resolution vs. Z

Legend:
- 2013 Reintegrated Resolution (100ns)
- Nich's Reintegrated Resolution -- 100ns

Y-axis: Resolution (μm)
X-axis: Drift Distance (cm)
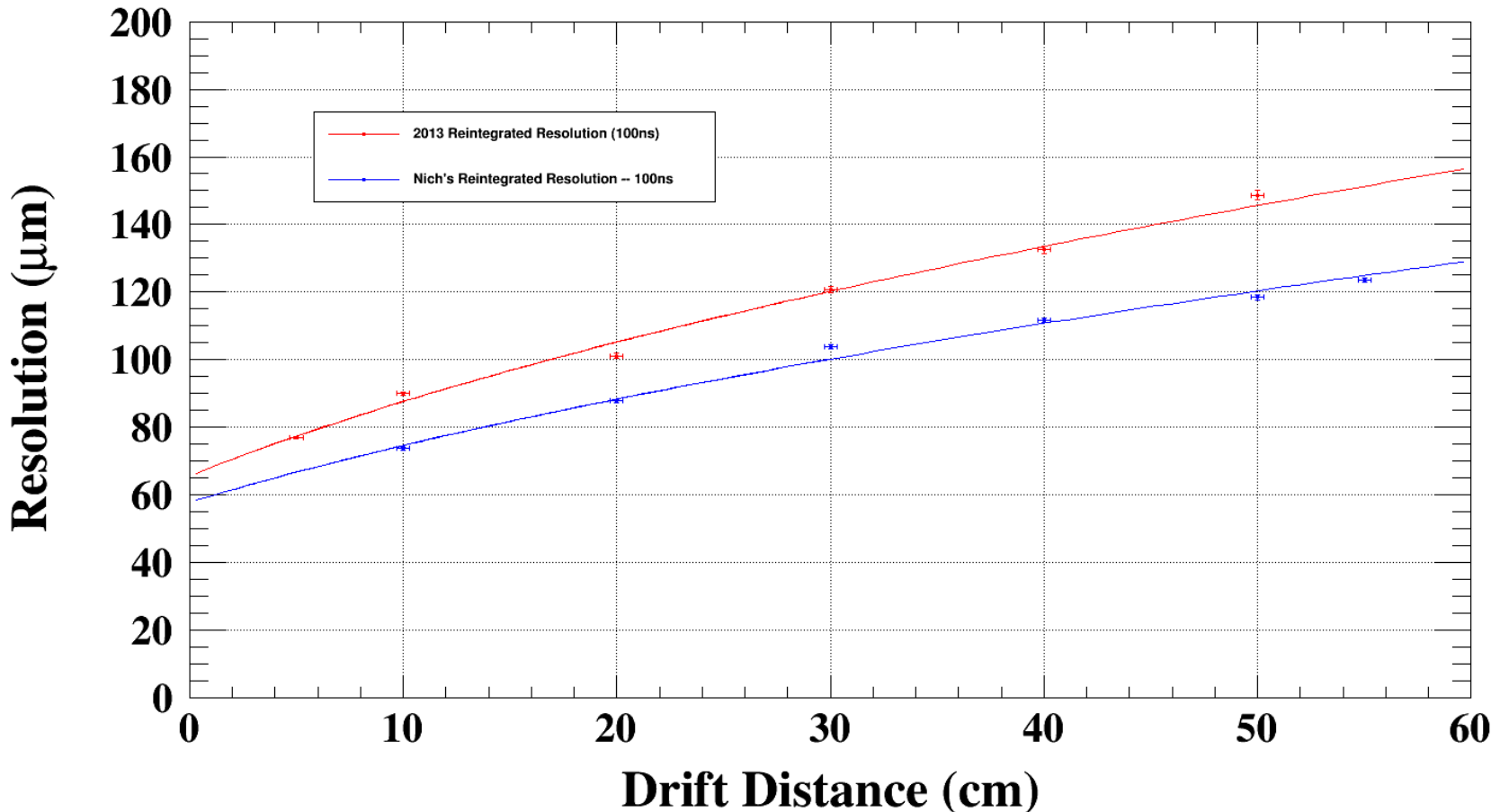
Part II: Overview of Code

- Code that has been successfully tested with ILCSoft v01-17-02
    - **MMPulseFinderProcessor**
    - **MMHitFinderProcessor (and PadResponseFunction)**
    - **MMHitTimeCorrectionProcessor**
    - **MMCalibrationProcessor**
    - **MMBiasCorrectionProcess**
- To-Do
    - **MMReintegrationProcessor**

13-08-27

- Code that has been added

  - **GeometricMeanResidualsTupleProcessor**

    - A simple processor that uses RAIDA to output inclusive and exclusive residuals to Root tuples, along with useful hit and track data (e.g., track parameters, hit position, etc.)

    - Uses the TrackFitterBase and a processor parameter to calculate residuals with user-specified track fitter

    - Useful for:

      - Bias calculation/correction

      - Quickly and easily calculating resolution with different cuts

- ## Code that has been added (cont)

  - ### DumpLCIO2Root

    - A compiled program to convert lcio data files to Root TTrees with useful data (e.g., hit Cell IDs, position, charge, etc). Currently supported data types: TrackerRawData, TrackerData, TrackerPulse, TrackerHit, and Track. Should be straightforward to add support for any other desired type.

    - Useful for:

      - Easily verifying output of (virtually) any processor

      - Finding out what's in that random LCIO file you can't remember making!

- ## To-Do

  - ### ResolutionCalculator

    - A tuples-based, Geometric Mean resolution calculation compiled tool for use with the output of the GeometricMeanResidualsTupleProcessor.

- Preliminary Procedure (Max Bin; neglecting channel mapping et al.)

  - **Basic Run**

    - MMPulseFinderProcessor

    - MMHitFinderProcessor

    - MMHitTimeCorrectionProcessor

    - TPCFix_FixCellIDs_forTracking

    - ClupatraProcessor

    - TrackFitterSimpleHelixProcessor (or some other TrackFitterBase-derived processor)

    - GeometricMeanResidualsTupleProcessor

- To-Do

  - **Reproduce previous analysis chain – i.e., use PRF Calibration, Bias calculation/correction, calculate resolution**

  - **Extensive Testing!**

Part II: Distortion Results

Coming Soon!

Part III: Future Direction

- Primary Coding Issue from our End == Solidifying Track Finding/Fitting procedure
  - Virtually everything after hits should be interchangeable between MM and GEM
  - Need a consistent, agreed-upon track finding and fitting procedure
  - Naive Suggestion: Clupatra (track finding), TrackFitterBase (track fitting)
  - TrackFitterBase seems to be a good framework to develop new track fitter tools while maintaining general compatability with the rest of the software

Part III: Future Direction

- **Short Term Goals:**
  - Use the new framework to calculate spatial and time resolution with curved and straight tracks
  - Produce module distortion plots

- **Longer Term Goals:**
  - Incorporate reintegration procedure into new ILCSoft
  - Investigate reintegration of zero-suppressed data
  - Investigate the use of LCCD

Part III: Future Direction

- Questions from us:

  – **What is the status of the GEM's track finding/fitting procedure, and resolution calculation?**

  – **Detector name is hardcoded into the Clupatra and MarlinTrk libraries as a check. The name it checks for is "LPTPC." However, our data all has the name "LP-TPC" (if I recall correctly). Is it possible in future versions for these libraries to include simple switches of some sort to avoid this confusion?**

- Questions for us?