The DMC Preprocessing Package

Jorge D. Morales Dave Toback

DMC Meeting

03/18/16

Overview

I. Introduction

II. How Data is Scaled in CDMSBats Processing

   A. Turning DMC Pulses into Input for CDMSBats

III. Processing Events with Known Calibration Constants

   A. How Data is Scaled in CDMSBats

IV. Calculating Calibrations Using the DMC Preprocessing Package

   A. How the Calibration Constants are Determined
   B. Optimal Filter in the DMC Preprocessing Calibration Tools
   C. If CDMSBats has an OF, why not just use that instead of the OF in the DMC Preprocessing?

V. PulseSim and How We Add Noise to the DMC Events

   A. Looking towards the future: Take PulseSim out of CDMSBats and put it into the DMC Preprocessing Package?

Introduction

- The DMC Preprocessing Package consists of the two different sets of tools that we need to turn DMC output (pulses) into events that we can process in CDMSBats the same way we do for Soudan data.

- The two sets are:

  a. ***Calibration Tools***: Determines the calibration constants
  b. ***Scale and Format Tools***: Uses calibration constants to take DMC raw-output events and give them correct format so we can process them with CDMSBats in the same way as real data

- These two tools are independent from each other, almost as if they were completely independent tasks. Set 2 only needs a set of constants determined, whether they came from Set 1 or not.

How Data is Scaled in CDMSBats Processing

- CDMSBats receives raw data (in ADC units) and scales it to *measured energy* with calibration factors defined in processing configuration files

- Typically the calibration constants are studied and determined with Ba data γ peaks

- A few key elements about the scaling of data pulses:
  - All calibrations factors are linear scaling factors
  - There is ADC to Amps scaling: doesn’t change, given by electronics specs
  - There is relative phonon calibration: to normalize channel response of same detector, for position resolution
    currently ignored by DMC assuming no relative calibration is needed/simulated
  - Amps to energy calibration *constant:* temperature dependent[1], slightly non-linear, but linearized with T^2
    also ignored in DMC
  - There are DMC specific calibration parameters in CDMSBats that just ignore the temperature dependence

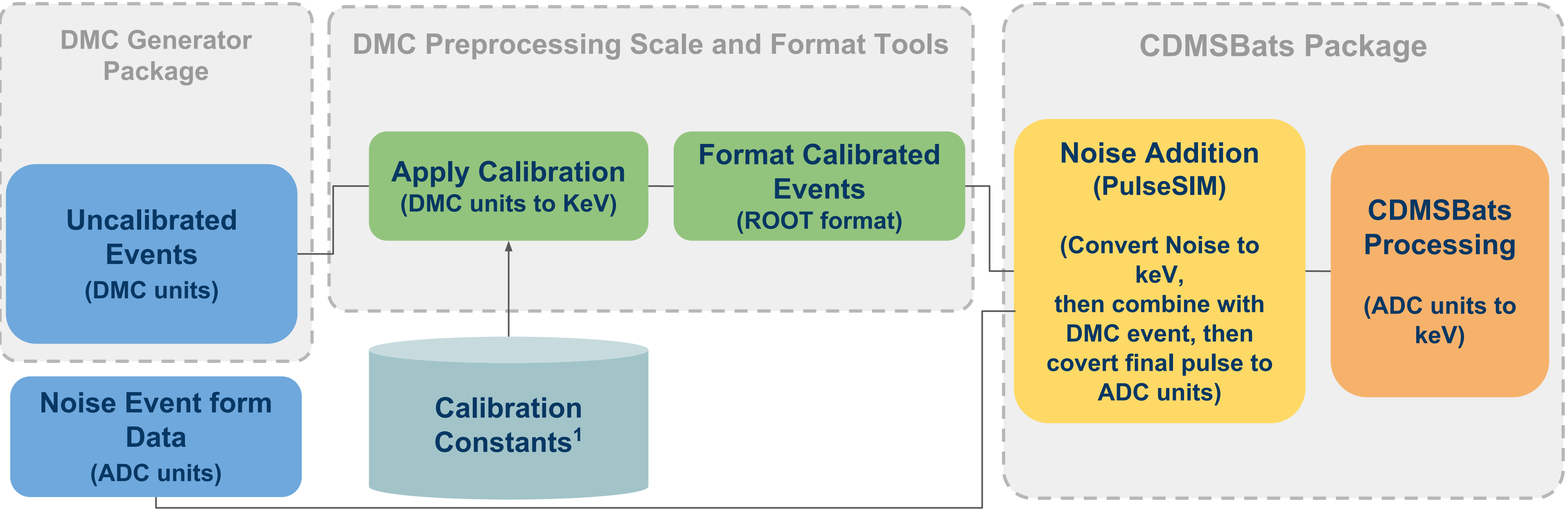
[1] http://titus.stanford.edu/cdms_restricted/Soudan/R133/ebook/130515/

Turning DMC Pulses into Input for CDMSBats

- CDMSBats takes input data files in Soudan ADC units
- Our DMC Generated Output is in *arbitrary* DMC units.
- We can’t just scale the DMC Pulses directly into ADC units because we need to add noise first.
- There is a patch in CDMSBats that adds noise (PulseSIM) but it requires the input pulses to be in energy units (keV)

The DMC Pulses are scaled to *truth energy* and then passed to PulseSIM which is part of CDMSBats

Processing DMC Events with Known Calibration Constants


DMC Generator Package
Uncalibrated Events (DMC units)
Noise Event form Data (ADC units)
DMC Preprocessing Scale and Format Tools
Apply Calibration (DMC units to KeV)
Format Calibrated Events (ROOT format)
Calibration Constants[1]
CDMSBats Package
Noise Addition (PulseSIM)
(Convert Noise to keV, then combine with DMC event, then covert final pulse to ADC units)
CDMSBats Processing
(ADC units to keV)

Note[1]:

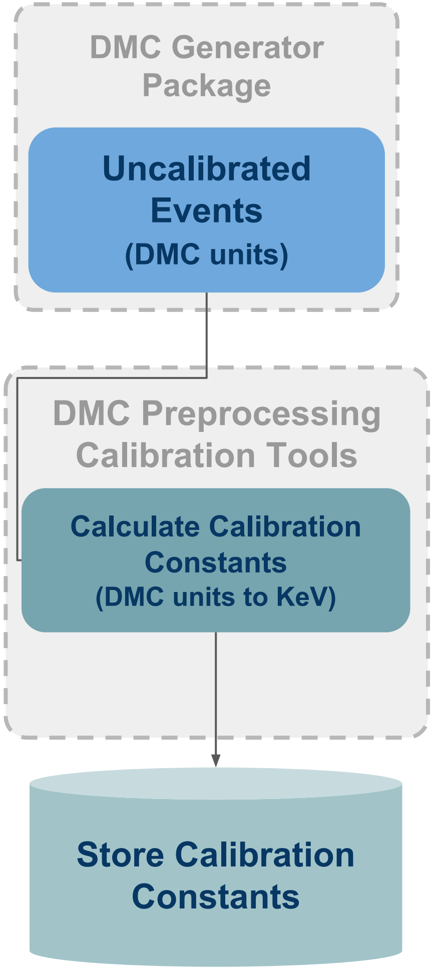
- The ***Calibration Constants*** can be determined by any means, not necessarily from the DMC Preprocessing Calibration Tools. Historically this is part of the DMC Preprocessing Package, it is useful that way to validate and understand, we'll get to *Calculating Calibration Constants* later

- The idea is to generate calibration constants with Ba and then use them for all other samples of the same version

Calculating Calibration Constants using the DMC Preprocessing Package

- The tools to create the calibration constants are in the DMC Preprocessing Package

- It is designed to scale the pulses from DMC output (DMC units) into energy units matching the *truth recoil energy*:
    - The calibration constants are ***linear/constant factors*** just like data
    - The tools for ***measuring amplitudes*** should be the same, and reliably give the same results as data, otherwise the calibrations will be skewed
    - Validation plots that show the algorithm results are obtained as part of the output, in this case a small website is created to display them


DMC Generator Package
Uncalibrated Events
(DMC units)
DMC Preprocessing Calibration Tools
Calculate Calibration Constants
(DMC units to KeV)
Store Calibration Constants

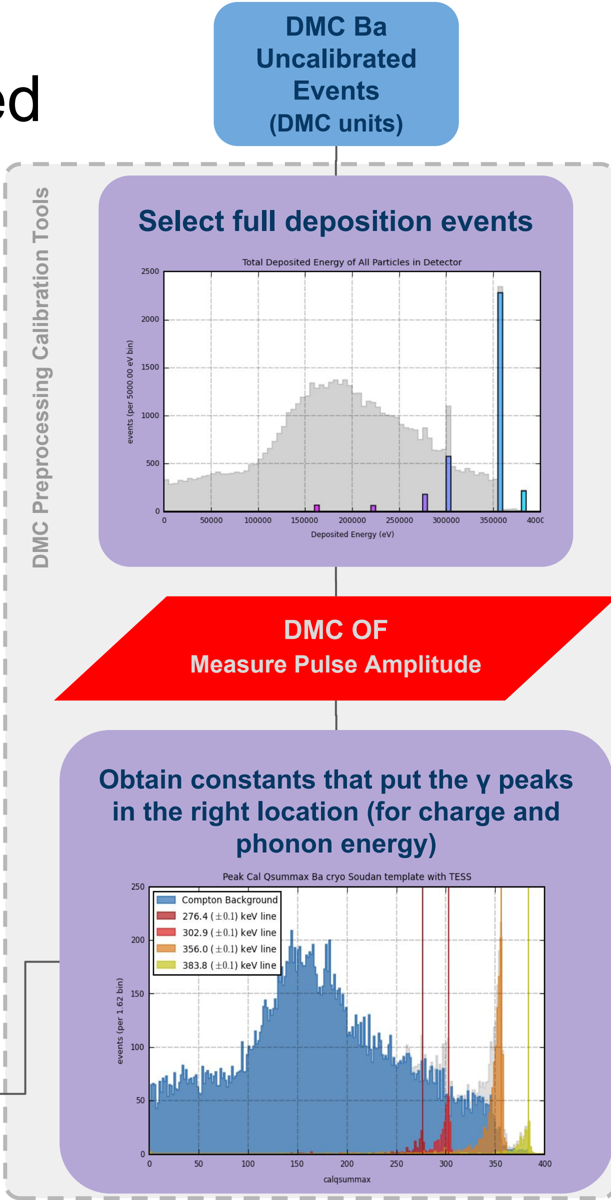
How the Calibration Constants are Determined

- To get Calibration Constants for DMC pulses we use Ba samples and do the following:

  a. From the *truth input energy* we select events which deposited all the energy in a nucleon recoil (i.e. without Compton scattering)

  b. We measure the amplitudes of the pulses uncalibrated pulses using the ***Optimal Filter***

  c. Then we obtain the constants that set the Ba γ peaks in the correct locations according to the *truth recoil energy* given by step a



DMC Ba Uncalibrated Events (DMC units)

DMC Preprocessing Calibration Tools

Select full deposition events

DMC OF
Measure Pulse Amplitude

Obtain constants that put the γ peaks in the right location (for charge and phonon energy)

Store Calibration Constants

Optimal Filter in the DMC Preprocessing Calibration Tools

- The ***Optimal Filter (DMC)*** is critical in the calibration process because later, when data is processed again an *Optimal Filter* (in CDMSBats) is used to scale from ADC units to *measured energy*.

- So, the *Optimal Filter* in the DMC Preprocessing Calibration Tools must achieve an equivalent measurement to the one in CDMSBats, otherwise the measured values will disagree and thus will be wrong.

- For that reason first we need to guarantee/validate that the ***OF*** in the DMC Preprocessing Package is the same as the one in CDMSBats[1].

[1] **This is complete now**, working on the validation talk but see implementation details here:
http://titus.stanford.edu/cdms_restricted/Soudan/R133/ebook/160318_jm/OFImplementation.pdf
http://titus.stanford.edu/cdms_restricted/Soudan/R133/ebook/160318_jm/ Python notebook with the code

If CDMSBats has an OF, why not just use that instead of the OF in the DMC Preprocessing?

- We are using the DMC Preprocessing Package as an external tool to scale, format, and obtain calibration constants for the DMC pulses before CDMSBats.

- It is not completely clear what we need to do to bring the DMC to its best, so we have been trying out ideas and need to understand every step in the process.

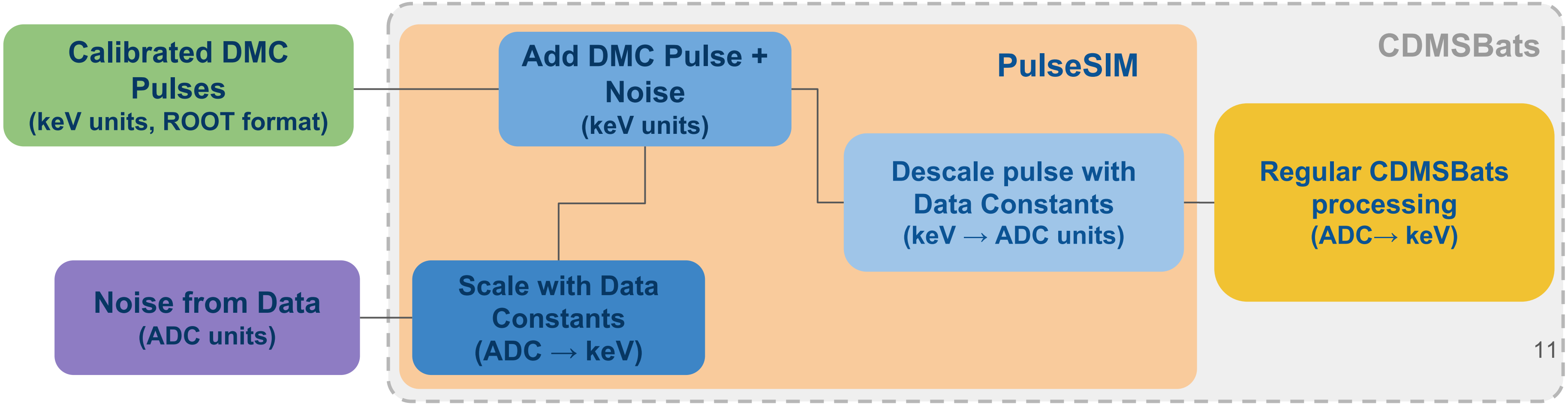
- While CDMSBats is great for processing large amounts of data, it is not ideal for debugging and testing → for now, the number one reason is ***versatility*** and ***transparency*** of the process.

- The second reason is that we understand how CDMSBats behaves, due to the multiple releases of data, so not modifying this layer and making the DMC match the input of CDMSBats seems like a better plan. At least until the whole calibration task is clear.

PulseSim and how we add Noise to the DMC events

- After DMC Preprocessing, PulseSIM is the CDMSBats module that turns ROOT formatted raw pulses, into processed quantities using the same tools as real data. It is the only tool to process DMC samples so far, and the one we are closest to get right, but the Binary module is under development (by KC) as well.

- PulseSIM expects the DMC Pulses calibrated to truth energy, both charge and phonon pulses are expected in energy units. Noise is scaled up to energy units, then it is added to the pulses, and scaled back down to raw trace amplitudes with inverted data calibration constants (ignoring Temperature dependence). Finally the raw trace is processed as any data in CDMSBats.


Calibrated DMC Pulses (keV units, ROOT format)
Noise from Data (ADC units)
CDMSBats
PulseSIM
Add DMC Pulse + Noise (keV units)
Scale with Data Constants (ADC → keV)
Descale pulse with Data Constants (keV → ADC units)
Regular CDMSBats processing (ADC→ keV)

Looking towards the future: Take PulseSim out of CDMSBats and put it into the DMC Preprocessing Package?

- We cannot just scale pulses in CDMSBats with different constants because of the intermediate step of Noise Addition, for now this is done in a patch inside CDMSBats: ***PulseSIM*** (we could really name it *NoiseSIM*).

- ***PulseSIM*** does the minimum to add noise to the DMC pulses, but requires the input in keV. Adding noise requires to match real data (noise in ADC units) with DMC pulses, which means that there will always be more than one ‘scaling’ in the process.

- In the future the Binary module (KC), will take the most appropriate scheme to add noise, but a proper ‘calibration’ is still needed (i.e. you don’t want to amplify noise with this calibration!).

- Should we still keep the DMC Preprocessed output in keV? The final result should be the same, but CDMSBats will have to be appropriately configured to do a single calibration without Temperature dependence.

Conclusions

- The ***DMC Preprocessing Package*** is capable of determining the calibration constants, scaling the pulses, and formatting them for CDMSBats, as well as handling the job submission and monitoring.

- The ***Calibration Tools*** determine scaling factors to scale DMC output into energy units using a *special* implementation of the *Optimal Filter* that replicates the same results as the one in CDMSBats

- The ***Scale and Format Tools*** use the calibration constants to scale the pulses to the appropriate energies, and give them proper format to process with PulseSIM and CDMSBats.

- In PulseSIM noise is added and the pulse is scaled down to raw trace units (ADC counts), finally CDMSBats turns the OF amplitudes into *measured* energy.

- The current scheme: DMC → DMC Preprocessing (Root) → PulseSIM → CDMSBats seems the way to go for now. Due to the ***leverage and flexibility*** of keeping ‘intermediate’ steps before the CDMSBats package, but a future version which doesn’t use PulseSIM to add noise and makes files raw binaries is under development (KC)