Update: Waveform Unfolding for the PDOM

John Kelley and Jim Braun

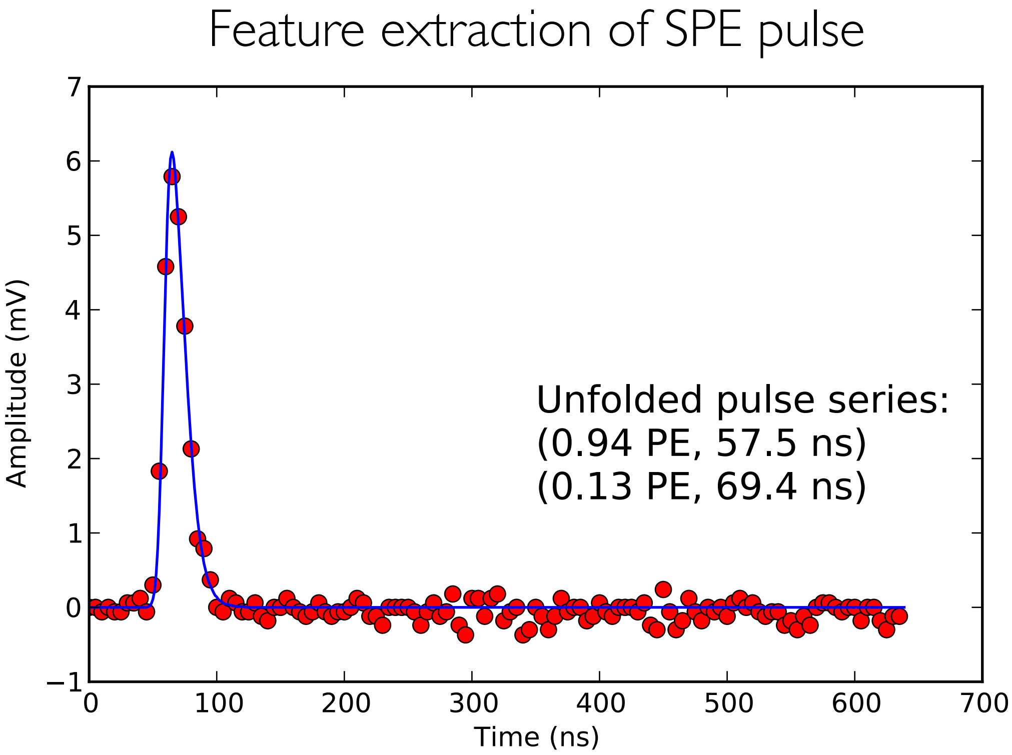
PINGU Hardware Call

October 16, 2014

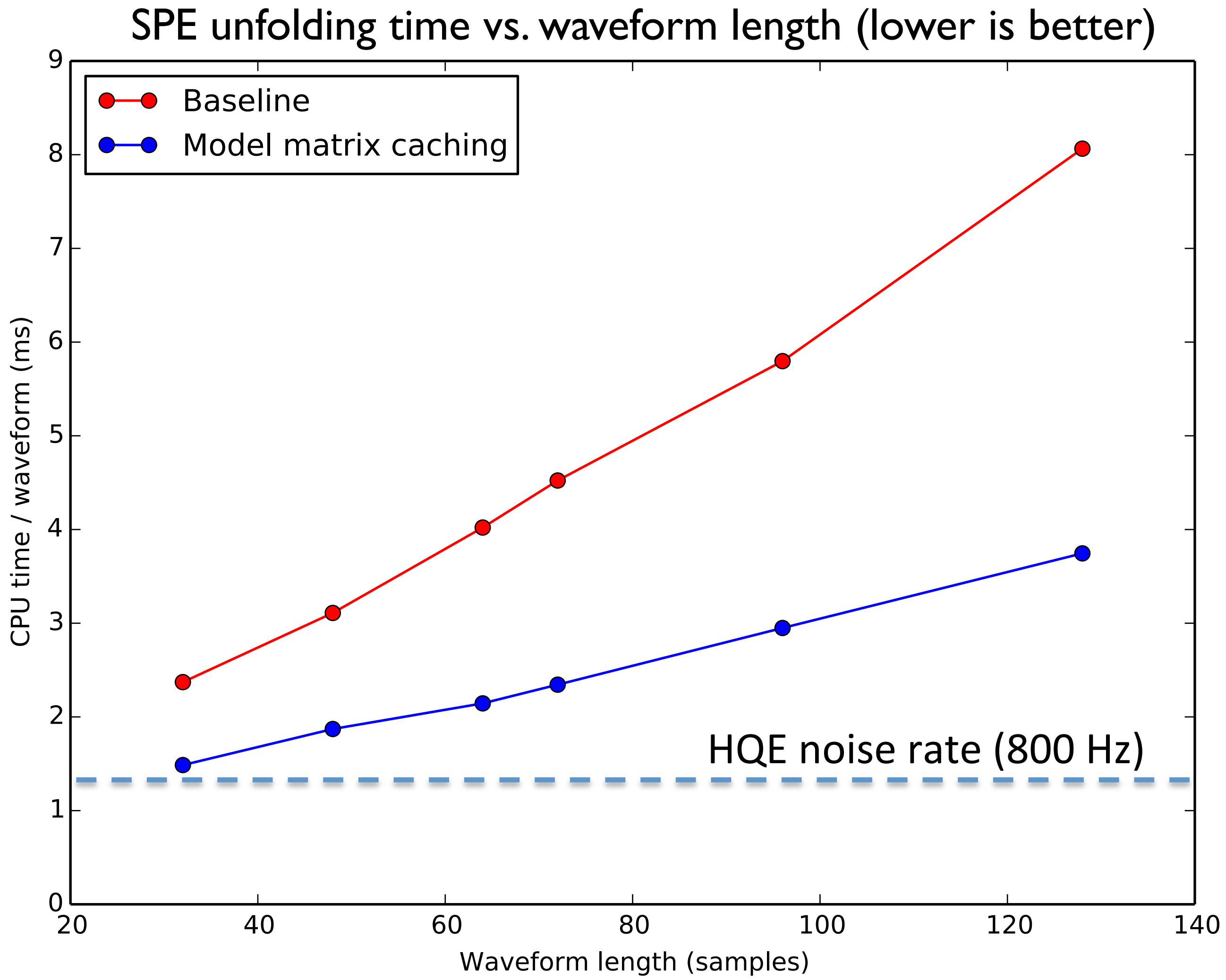
Reminder

- IceCube runs feature extraction to convert raw waveforms to a pulse series
  - pairs of (charge, time) with given pulse templates
  - filtering clients calibrate data and run on all events

- Goal: pull this code out of IceTray and get it running on a SOCkit dev kit
  - benchmarking... can we unfold SPEs at > 1 kHz?
  - power consumption



Initial benchmarking (ARM Cortex A9)



Not quite fast enough... but in the ballpark after 1 day of tuning
Preliminary tests suggest a multithreaded version will be > 1 kHz

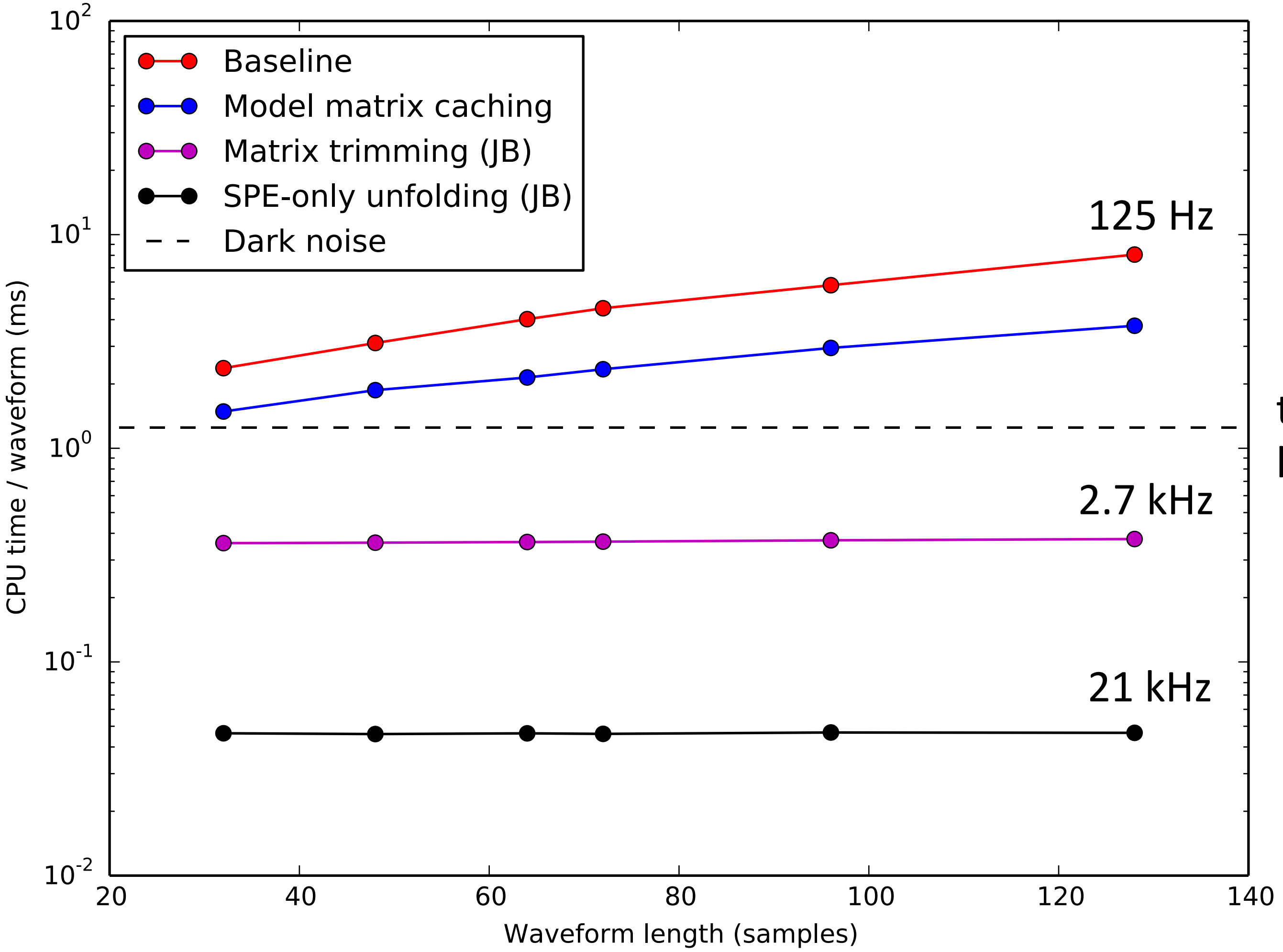
Updates since Geneva

- J. Braun taking over WaveDeform project from J. van Santen

- New optimizations (by J. B.):
  - automatic selection of waveform length, model matrix size (pick interesting portion of waveform)
  - simplify first iterations through the unfolding algorithm
  - added SPE-only "unfolding"
    - not a fundamentally different algorithm, just an optimized first iteration of WaveDeform
    - no time-consuming matrix operations!

- General speed improvements can also be back-ported to IceCube

Latest Benchmarks: SPE unfolding rate



target: 800 Hz
lower is better

Both new methods are fast enough to run on the dark noise

Arm Core Necessary?

- Demonstrated that our best feature extraction can run on a Gen2 DOM with ARM core

- If we only feature-extract SPEs, though, do we need the ARM?
  - SPE algorithm can be adapted to firmware or *maybe* soft core
  - don’t expect to save 800 mW, though (ARM power at full bore)
  - power consumption TBD

- Still should use a processor for DAQ slow control, managing data requests etc.
  - NIOS soft-core power depends on config (200 mW for ii/f?)
  - ARM power (good clock management) might still be OK if load is moderate

- Still open issues besides the algorithm
  - data buffering, movement
  - droop correction (!)

Backup

domwave

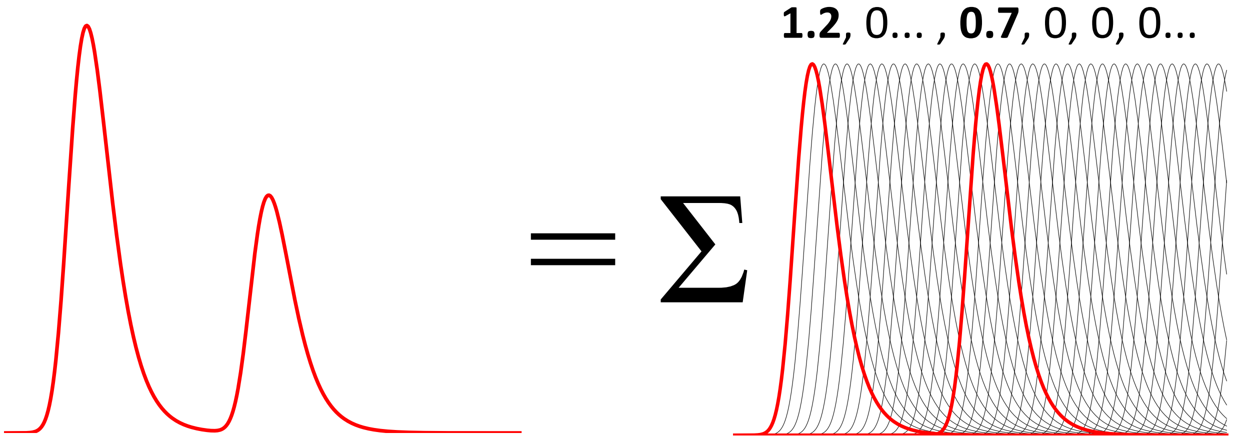
- http://github.com/WIPACrepo/domwave

- Standalone C99 port of WaveDeform
  - pull out all IceTray hooks
  - PDOM simplifications (single-channel front end)
  - add prototype pulse shape (courtesy K. Jero)
  - add command-line interface for testing

- SPE-only algorithm: see `getPulsesSPE()` in `domwave.c`

Waveform Unfolding

- Use time-shifted SPE pulse templates as a set of basis functions for a PMT waveform



- Non-negative least squares solution to matrix equation **y = Ax**
  - **y** is the binned waveform as a column vector
  - A is the basis matrix (pulse templates as columns)
  - **x** is the best-fit unfolding (pulse amplitudes as column vector)

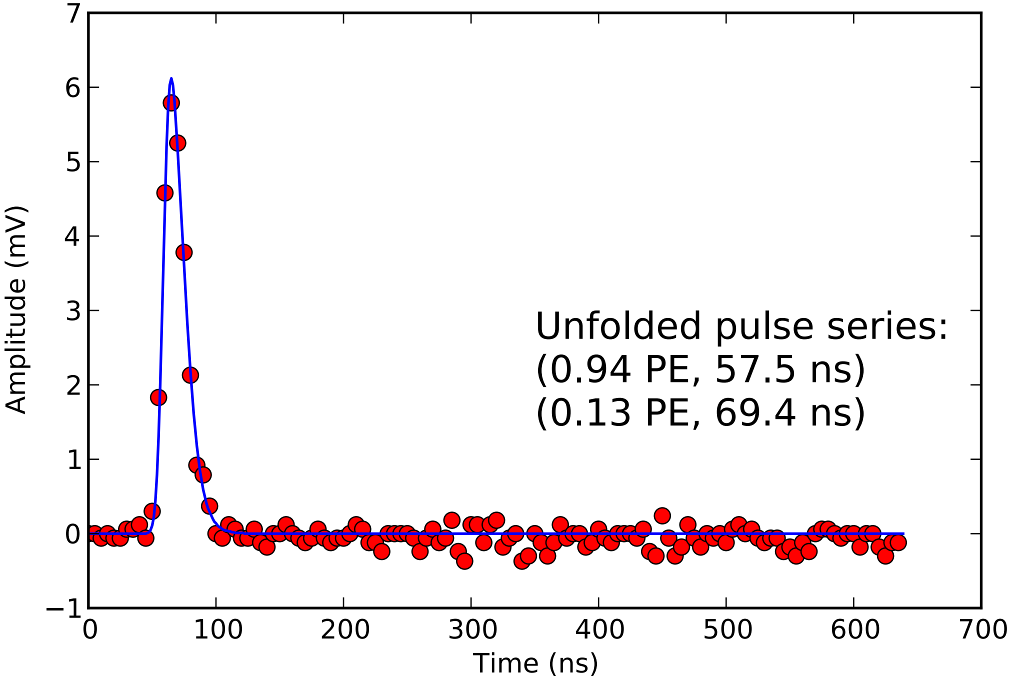
- Minimization code (Lawson-Hanson) can be used as-is from IceCube

- Some other tweaks
  - input waveforms are "road-graded" (bins below noise threshold set to zero)
  - saturated bins are not included

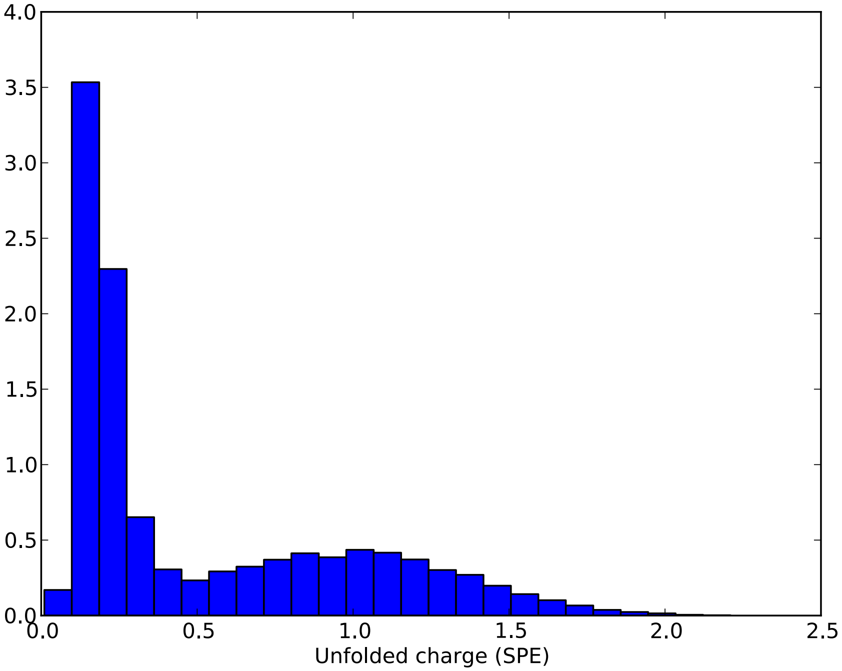
Sample Results

SPE data from early K. Jero lab setup (PMT + TI ADC + prototype firmware)

Pulse series and reconstructed waveform



Charge histogram (SPE tuned to 1.0)



SoCKit power consumption

SoCKit state	Power (W)
bootloader	5.95
Linux (idle)	5.77
Linux (unfolding, CPU at 100%)	6.14

delta: 370 mW

N.B.: SoCKit contains a lot of subsystems not on a mainboard (serial, Ethernet, backlit LCD)

Unknown idle draw of CPU alone... estimate is 250-500 mW (update from K. Hanson: estimate is 800 mW from spreadsheet)