DVCS software and analysis tutorial

Carlos Muñoz Camacho

Institut de Physique Nucléaire, Orsay, IN2P3/CNRS

DVCS Collaboration Meeting
January 16–17, 2017

Outline

- Online software (decode raw data $\rightarrow$ online ROOT files)
- Offline software
  (calorimeter analysis: ARS waveform analysis, clustering, etc)
- Time-dependent information: the databases
- In practice:
  where to find input files, how to setup code, where/how to run it...
- "Useful" scripts available: waveform analysis, π^0 subtraction, etc

1st step: decode CODA raw data

Uses the standard Hall A analyzer plus one DVCS library (libTDVCSCalo.so) to output calorimeter information to the ROOT file as a C++ object (not just simple leaves into an ntuple).

- This steps requires simple to run a very similar script as the one ran during the shifts
- Need to be ran again making sure all HRS database parameters are up-to-date (BPM+BCM calibrations, HRS angle+momentum, beam energy and optics matrix)
- The script is slightly different because of additional fine corrections added wrt online analysis: target-mass corrections, energy loss in the target, raster corrections, etc)
- Check w/ Frédéric for the best and latest script (+corresponding DB)

1st output file

The output ROOT file after decoding CODA raw files contains:

- Final reconstructed scattered particle (as a TLorentzVector assuming a massless particle)
- Vertex of interaction
- Raw calorimeter data (i.e. ARS data) encapsulated in a TCaloEvent C++ object
- Lots of HRS detector variables as defined in the LHRS.odef file

Warning: scattered particle and vertex are output several times with/without different corrections (target-mass, energy-loss, etc). Make sure you choose the one you need for your further analysis.

Runtime: ~30 min for 1h run
Filesize: similar to the original CODA files (maybe ~10-20% smaller)

2nd step

A ROOT C++ script to perform calorimeter ARS waveform analysis

- Amplitude(s) and arrival time(s) for each calorimeter block are computed and written into the TCaloEvent dedicated branches
- ARS raw data (amplitude of each sample/block/event) are dropped to reduce file size
- No further analysis is done
- Time consuming...

Runtime: $\sim$30-50h for a 1h run
depending on waveform analysis parameters
Filesize: ~ 5 times less than raw files (i.e. 400Mb for a 2Gb split)

3rd (and final) step

Calorimeter analysis

- 2×2 threshold applied to blocks
- Clustering
- Position and energy computation for each cluster
- Photon "physics" variables computed: ϕ, t, M_X^2, etc
- Outputs a simple and small ntuple

Runtime: ∼1h for a 1h run
Filesize: Very small (~ 30 Mb for an initial ∼50 Gb run)

Note: Additional scripts (similar to this one) will run on the waveform-analyzed file for specific analysis tasks (accidental subtraction, π^0 contamination computation, etc).

Time-dependent parameters

- **HRS analysis** (1st step) uses a text-based database, located in the $DB directory
  - Information is sorted by date tags in the files
  - One text file per HRS detector (S2m, Cerenkov, etc) or system (raster, bpm, etc)

- **Calorimeter analysis** (2nd+3rd steps) uses a MySQL database (jmysql server at JLab)
  - Data is sorted by run number
  - Access is more simple and more robust than a collection of text files
  - Typical information contain is: time/energy calibration coefficients, calorimeter positions, beam energy, etc
  - ROOT interface for reading from and writing into the DB

MySQL DB use

- Run number should be initialize (in the appropiate way - see examples) at the beginning of your script
- If you analyze runs with different calorimeter parameters in the same script, you should chain them using the DVCS class TDVCSChain (inherits from the ROOT class TChain).
- TDVCSChain updates the run parameters when the loop switches to a different run

Warning:

- Using TDVCSChain is tricky and the code may crash. Analyzing several runs in a script is rare and TDVCSChain is not maintained very well.
- A combined analysis of several run is (usually) only performed after calorimeter analysis, and no DB access is needed (simple TChain of ntuples is enough).

Raw file location

- All data is now (only) in MSS:
  /mss/halla/dvcs/raw/dvcs14_(Run).dat.(Split)
- Can be writte to disk from the ifarm computers, eg:
  jcache submit /mss/halla/dvcs/raw/dvcs14_15000.dat.0
  (file will appear as /cache /mss/halla/dvcs/raw/dvcs14_15000.dat.0)
- DVCS environement can be setup in the ifarm computer by sourcing the file: **/work/halla/dvcs/disk1/carlos/64bits/setup.sh**
  Note: the $DB may not be the latest one (check with experts)
- Preliminary runlist maintained in GoogleDocs at:
  https://docs.google.com/spreadsheets/d/1XJdPbu4MyWeruE-Bpdiz4glHHhMK35VEmPbtUlSBG2g/edit?ts=58169f79#gid=0
- Final runlist available for Spring16 (Frédéric) and Fall'14 (Mongi - report at this meeting)

Disk space available for DVCS analysis at JLab

- Work disks: **/work/halla/dvcs/disk*** (warning: not backup-ed)
- **/volatile/halla/dvcs**: "virtually" unlimited in the short-term, but files will disappear after a few weeks/months
- **/mss/home/username/...**: unlimited, but not a disk - need to *jput* to and *jcache* from it

Sample scripts available for common analysis tasks

- Decoding raw CODA data (1st step)
- Waveform analysis (2nd step)
- Calorimeter analysis (3rd step)
- Accidentals calculation (same as 3rd step, but w/ a shifted time cut)
- π^0 subtraction

To get latest versions and explanations, contact Mongi, Frédéric or myself, preferably this week, or by email later.

EPICS and scaler data

- Output to the ROOT file as simple leaves
- Kept up to step 3 (reduced ntuple),
  but can be added (copied) if necessary to any file

Monte-Carlo simulation

Not covered here:

- A tutorial and write-up by Rafo is available in the ELOG
- Contact me if you need more details or have questions