MYA-only beam charge estimation

Sho Uemura
SLAC

(Dated: January 11, 2016)

This document explains how MYA is used to find SVT-good time ranges and integrated beam charge for the 2015 engineering run.

I. CODE AND FILES

Shell scripts, crawler output and MYA dumps (gzipped; gunzip before using) are in SVN at `/java/sandbox/biascrawling/crawler`.

A. Commands

Get start and end times (first and last TI timestamps) from an EVIO file:

```
java -cp hps-distribution-bin.jar org.hps.evio.BasicEvioFileReader -qcts in.evio
```

Display plots of run ranges:

```
java -cp hps-distribution-bin.jar org.hps.conditions.svt.SvtBiasConditionsLoader
-tc runcrawleroutput.csv -dm biasdump -p motordump -s
```

Load ranges to conditions DB (dev.prop is a connection properties file for the conditions DB):

```
java -Dorg.hps.conditions.connection.file=dev.prop -cp hps-distribution-bin.jar
org.hps.conditions.svt.SvtBiasConditionsLoader -tc runcrawleroutput.csv -dm biasdump
-p motordump -g
```

Calculate per-run charge integrals:

```
java -cp hps-distribution-bin.jar org.hps.users.meeg.SvtChargeIntegrator -rt runcrawleroutpu
fcupdump
```

Calculate per-file charge integrals:

```
java -cp hps-distribution-bin.jar org.hps.users.meeg.SvtChargeIntegrator -t filecrawleroutpu
fcupdump
```

(For charge integrals, `-t` uses TI timestamps instead of head bank timestamps for better precision, and `-c` uses the TI time offset estimates in the CSV file instead of the run DB so runs without run DB information will still work. The commands listed here duplicate what appears in the run spreadsheet.)

II. RELEVANT FEATURES OF MYA, DUMPING MYA

MYA archives EPICS variables (PVs). See MYA documentation at http://devweb.acc.jlab.org/controls_web/certified/mya/. MYA only archives PVs that have been requested to be archived (form at http://devweb.acc.jlab.org/controls_web/cjs/request/request.php). The request can include a deadband (minimum change in value before the new value is archived). The SVT bias channels are archived with a 1 V deadband. There is no deadband for the motor positions.

MYA requests for MPOD-supplied SVT voltages (including all bias channels) were only submitted on May 5, 2015. Runs prior to 5403 have no bias information in MYA or in EPICS events; bias periods will be recovered from the data.

`myData` dumps the history of a list of PVs over a specified time range. Each row of the output is a snapshot of the archived values at some point in time; a row is printed for every time an archived value changed.

III. SVT TIME RANGES

The bias and position time ranges are extracted from the MYA dumps and written to the conditions DB. Each run is assigned every bias and position time range it overlaps with. This means a single bias or position time range may be assigned to multiple runs.

A. Bias

The bias dump (`biasdump.gz`, generated by `bias_dump.sh`) contains voltages for all 36 bias channels.

The bias time ranges are calculated by `org.hps.conditions.svt.SvtBiasMyaDataReader.readMyaData()`. A bias time range starts at the first row of the bias dump where every channel is at 178 V or higher, and ends 2 seconds before the first row where any channel drops below that threshold. The 2-second margin allows for delays in the IOC and MYA. The value stored in the bias time range (which is never used) is the voltage of the lowest-voltage channel at the start of the bias range.

B. Position

The motor dump (`motordump.gz`, generated by `motor_dump.sh`) contains values for the two motor positions. These are the raw positions of the linear shifts.

The position time ranges are calculated by `org.hps.conditions.svt.SvtMotorMyaDataReader.readMyaData()`. A position time range is created whenever there is more than 10 seconds between two rows of the output (i.e. the motors were stationary for 10 seconds). The values stored in the position time range are the top and bottom SVT angles relative to nominal (0.5 mm), computed from the motor positions using the same formulas used in EPICS (`xpsMotorApp`).

IV. SVT EFFICIENCY

The SVT DAQ flags (`svt_burstmode_noise_good`, `svt_latency_good`) each affect a fraction of triggered events. Burst-mode noise affects 3.5% of events (at 17 kHz — this fraction scales linearly with trigger rate). Latency affects 2 out of 6 trigger phases, only in runs where the SVT latency was set incorrectly (all Engineering Run data before run 5722). The conditions DB is used to identify runs with bad latency.

V. CHARGE INTEGRALS

The Faraday cup dump (`fcupdump`, generated by `fcup_dump.sh`) contains the Faraday cup current (`scaler_calc1`) and pulser livetime (`HPSTRIGSC_LIVETIME`).

For each line in the Faraday cup dump, the current, livetime, and time interval from the previous line are multiplied to get an estimate of the gated charge in the time interval (for ungated charge the livetime is omitted). The SVT efficiency factor (burst-mode noise and latency effects) times the gated charge is called the "good" charge for the time interval. If the run start/end or an SVT-good start/end falls within the time interval, the partial interval is used. All such intervals are summed to get the total (or with bias, or with bias and at nominal position) good/gated/ungated charge for the run or file.

For per-run integrals, the run start and end times are the `GO` and `END` event timestamps if available, and first/last head bank timestamps otherwise. For per-file integrals, the file start and end times are the first and last TI timestamp (plus the "TI time offset," which is an estimate of the offset between TI timestamps and the head bank Unix timestamps).

VI. ERRORS AND SYSTEMATICS

The SVT flags (`svt_bias_good`, `svt_position_good`) are based on the time ranges, so a charge calculation based on the time ranges is exactly correct for an analysis that uses the flags to reject events. This assumes that the SVT is actually good for the full duration of the SVT-good time ranges. The SVT-good time ranges only start when a set of values is seen that indicate the SVT is good; assuming the timestamps are sync'ed up correctly between all systems, there should be no possibility for error here. Since the bias normally only starts ramping down after beam is lost (and there is a 2-second margin built in to the bias time range), and the SVT is only opened by the shift worker after the beam is lost due to FSD trip, the ends of the SVT-good time ranges should also be conservative.

As a test, the bias time ranges were checked for run 5785, which is unblinded (field-off run) and has a beam trip. We found that the EPICS events are consistent with the MYA values, and that noise hits in the SVT show the bias was off only outside of the SVT-good time range. Limitations of this test: EPICS events are only every 2 seconds, and noise hits only become significant at low bias (below 20 V or so).

The run start/end times (based on `GO` and `END` events, or head bank timestamps) have 1 second precision (we can use TI timestamps instead). The file start/end times (based on TI timestamps) have nanosecond precision for the file duration, and absolute time precision limited by the TI time offset (millisecond precision). There is some time between the last event of one file and the first event of the next, but at 17 kHz this gap is negligible, even if events are recorded out of order (which happens rarely and is not a large effect).

The accuracy of the pulser livetime is unknown.

The error introduced by hand-integrating the Faraday cup current (as opposed to using the Faraday cup scaler directly) is unknown. But since the Faraday cup current seems to vary by no more than a couple % during steady running, it is probably on the order of 1%.

SVT DAQ errors (marked by the `svt_event_header_good`) are not yet accounted for. Most of the errors marked by this flag are caused by a DAQ lockup that affects all events from a certain point in the run. An analysis is in progress to identify the lockup times and apply them to the charge integrals.