
CERN

SIS tests in UCAP for TCDIL and TDIS, bunch length interlocks & SIS V2

J. Wenninger

Interlocks on TCDILs (and TDIS revision)

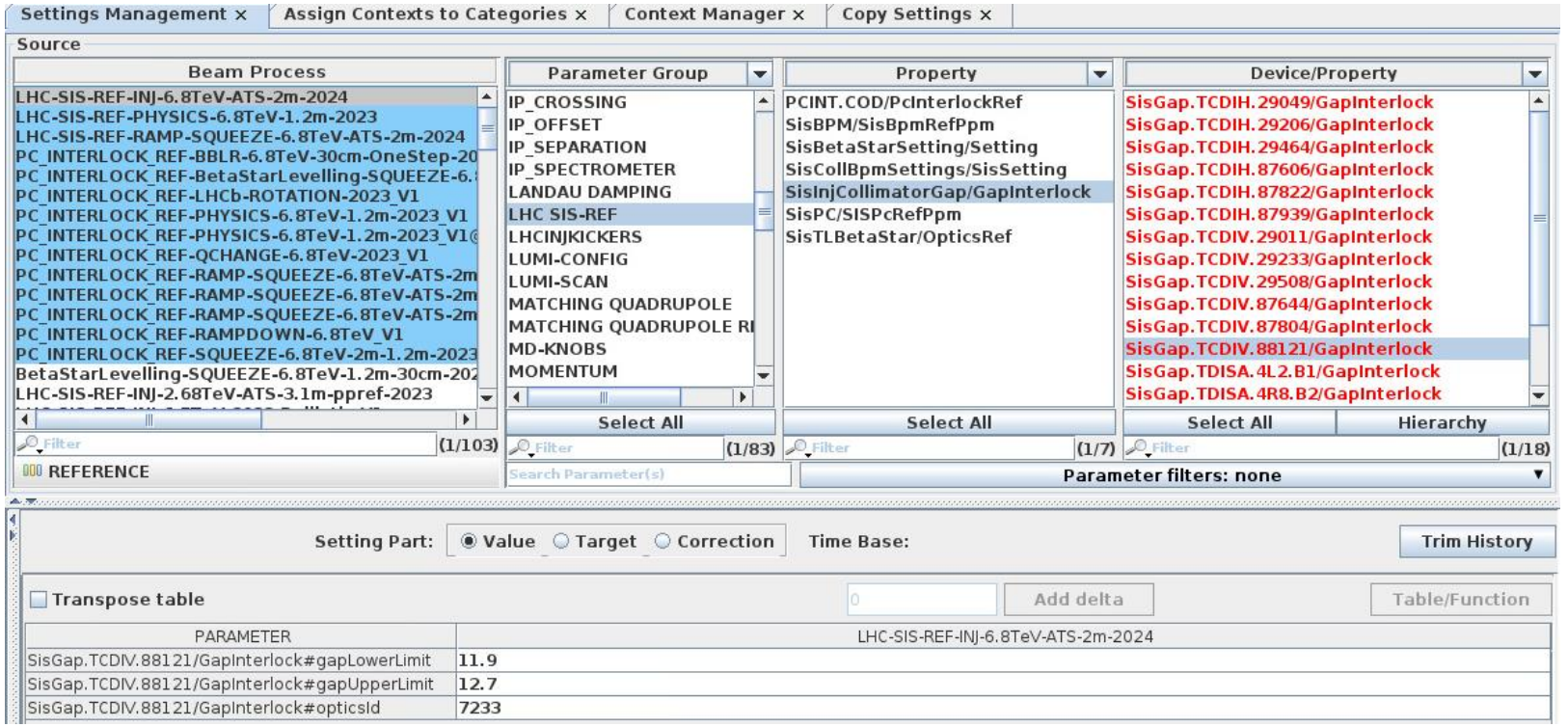
- Since Run1 we are discussing a **redundant interlock on TCDI collimators** to avoid mishaps as happened many years ago when the SPS optics was changed (Q20 vs Q26).
  - Gaps were left at Q26 settings while SPS ran with Q20.
- The idea was to **re-use the beta* gap interlock concept** of the ring collimators. **TL optics ID telegrams** were added (same role as beta*), the optics ID is filled by LHC SIS.
  - Optics ID checked after every SPS pulse using the TL quadrupole currents.

- **Nothing ever happened…**

- **Resurrection** of the idea (not at Easter) in software form, use as test **for SIS interlocks in UCAP**.
- Implement an **Injection** interlock in **SIS**.

Interlocks on TCDILs (and TDIS revision) (2)

- **Test of gap versus SIS reference in LSA**, optionally also **check of optics ID** validity.
  - The reference is in a BP attached to the hypercycle → can accommodate different settings for pp and ions.
- This concept happens to also **fit the needs for the TDIS** → use the opportunity to make TDIS gap interlock in SIS more flexible (no more fixed limit in project xml configuration).



If opticsID = 0 → not checked (for TDIS)

Interlocks on TCDILs (and TDIS revision) (3)

- UCAP transformation (1 per TCDIL and 3 per TDIS [A,B,C]) on node **UCAP-NODE-LHC-SIS**.
- Each transformation publishes:
  - All input values (measured gaps, gap interlock limits, optics IDs),
  - **Boolean result → used in SIS for test.**
  - Text message explaining interlock test decision.
- Once stable → can be **logged in nxcals** (not yet).
- During the **2023 ion run**:
  - All **TCDIL tests** were **active during part of the run** – all ok.
  - For the TDIS, the **new tests** based on UCAP **ran in parallel to the old versions** – ok.
- In **2024**:
  - **TCDIL tests** will become **operational**.
  - **New TDIS tests** will **replace old versions**.

And more UCAP

- The SIS tests on the **injection buckets** (based on BQMs, prevents over-injection is more than 1 bunch in a ring & checks bucket requests vs AGK limit) and **SPS beam intensity interlocks** (total I, intermediate I, etc) – both acting on **injection** – were converted to UCAP to evaluate the conversion time.
  - It is quick (< 2 hours for a test type).

- In 2024 both tests will run in UCAP transformations (node **UCAP-NODE-LHC-SIS**).
  - Same publication philosophy than for TCDIL/TDIS gap tests (all inputs, test result, text message).
  - SPS intensity threshold as setting in LSA.

```
UCAP-NODE-LHC-SIS> dl
LHCSIS-INJ.BQM.B1 -> BqmInterlock RUNNING (inputs: 335722, results: 335722, no issues)
LHCSIS-INJ.BQM.B2 -> BqmInterlock RUNNING (inputs: 335723, results: 335723, no issues)
LHCSIS-INJ.BQM.SPS -> BqmInterlock RUNNING (inputs: 305012, results: 305012, issues: 100)
LHCSIS-INJ.SPSBCT -> IntensityInterlock RUNNING (inputs: 3523735, results: 3523735, no issues)
LHCSIS-INJ.TCDIH.29049 -> CollimatorGapInterlock RUNNING (inputs: 3102944, results: 3102944, no issues)
LHCSIS-INJ.TCDIH.29206 -> CollimatorGapInterlock RUNNING (inputs: 3102892, results: 3102892, no issues)
LHCSIS-INJ.TCDIH.29464 -> CollimatorGapInterlock RUNNING (inputs: 3102381, results: 3102381, no issues)
LHCSIS-INJ.TCDIH.87606 -> CollimatorGapInterlock RUNNING (inputs: 3103277, results: 3103277, no issues)
LHCSIS-INJ.TCDIH.87822 -> CollimatorGapInterlock RUNNING (inputs: 3103253, results: 3103253, no issues)
LHCSIS-INJ.TCDIH.87939 -> CollimatorGapInterlock RUNNING (inputs: 3103301, results: 3103301, no issues)
LHCSIS-INJ.TCDIV.29011 -> CollimatorGapInterlock RUNNING (inputs: 3102940, results: 3102940, no issues)
LHCSIS-INJ.TCDIV.29233 -> CollimatorGapInterlock RUNNING (inputs: 3102886, results: 3102886, no issues)
LHCSIS-INJ.TCDIV.29508 -> CollimatorGapInterlock RUNNING (inputs: 3102385, results: 3102385, no issues)
LHCSIS-INJ.TCDIV.87644 -> CollimatorGapInterlock RUNNING (inputs: 3103275, results: 3103275, no issues)
LHCSIS-INJ.TCDIV.87804 -> CollimatorGapInterlock RUNNING (inputs: 3103248, results: 3103248, no issues)
LHCSIS-INJ.TCDIV.88121 -> CollimatorGapInterlock RUNNING (inputs: 3103296, results: 3103296, no issues)
LHCSIS-INJ.TDISA.A4L2.B1 -> CollimatorGapInterlock RUNNING (inputs: 3102403, results: 3102403, no issues)
LHCSIS-INJ.TDISA.A4R8.B2 -> CollimatorGapInterlock RUNNING (inputs: 3102818, results: 3102818, no issues)
LHCSIS-INJ.TDISB.A4L2.B1 -> CollimatorGapInterlock RUNNING (inputs: 3102407, results: 3102407, no issues)
LHCSIS-INJ.TDISB.A4R8.B2 -> CollimatorGapInterlock RUNNING (inputs: 3102811, results: 3102811, no issues)
LHCSIS-INJ.TDISC.A4L2.B1 -> CollimatorGapInterlock RUNNING (inputs: 3102393, results: 3102393, no issues)
LHCSIS-INJ.TDISC.A4R8.B2 -> CollimatorGapInterlock RUNNING (inputs: 3102805, results: 3102805, no issues)
Listed 22 devices (22 transformations, 0 actors)
```

SIS JAVA tests to UCAP?

PROs

- Smaller projects/entities can help simplify maintenance and tests.
- Detailed test data (input values, thresholds…) may be **published → NXCALS** for diagnostics.

CONs

- Not a single data buffer like in SIS, may imply adding data concentrator layers to avoid replication.
- **Lack of *user-friendly* diagnostics** (subscription state, transformation state, structure of nodes etc).
  - Only the OP Michi tool can help.
- **UCAP becomes mission critical**.
  - HW reliability, SW updates etc.
- **Changing names of devices, parameters, fields** is heavy and disruptive since It implies CCDE updates.
  - Must change transformation, SIS and CCDE at the same time… Testing??

Too early to engage in a "massive" conversion to UCAP !

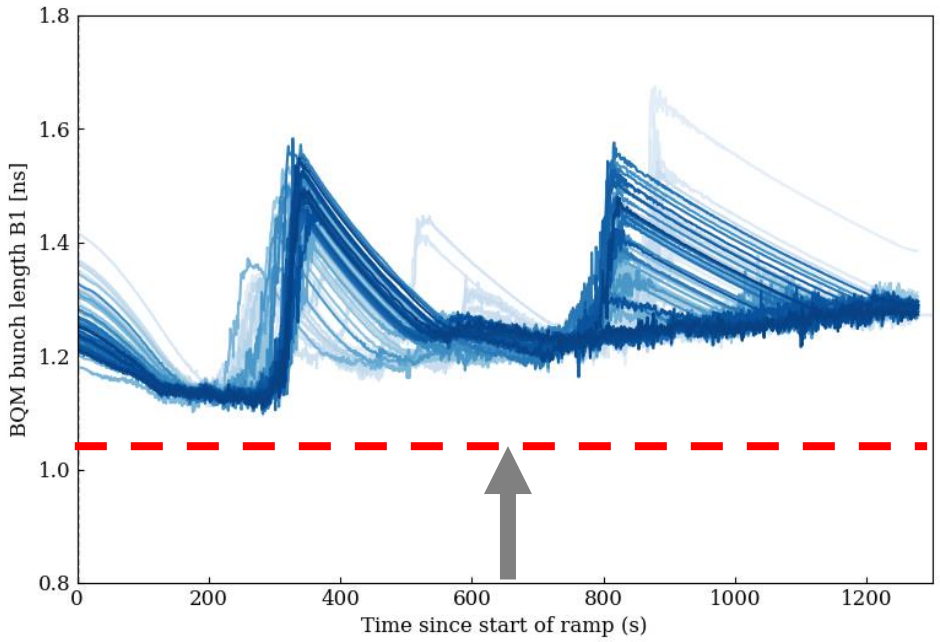
Bunch length interlocks in SIS

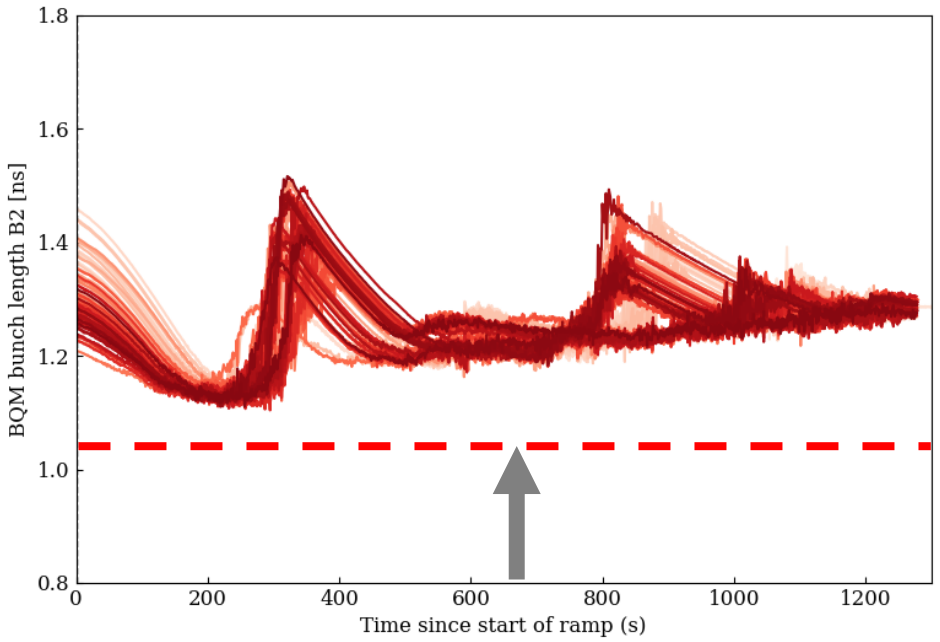
- Since a few years there is an **interlock on the bunch length in SIS** to cover a failure of the longitudinal blow up in the ramp.
  - The initial motivation was driven by **collimator temperatures**.
  - The threshold was set to **0.8 ns for >= 500 bunches** (on 15s averages).
  - **Only active in the ramp**.

- With the problems of the vacuum modules, one should consider **tightening the threshold** to avoid issues with more vacuum modules. Failures of the longitudinal blowup are rare but do happen !

Bunch lengths in 2023 - ramp

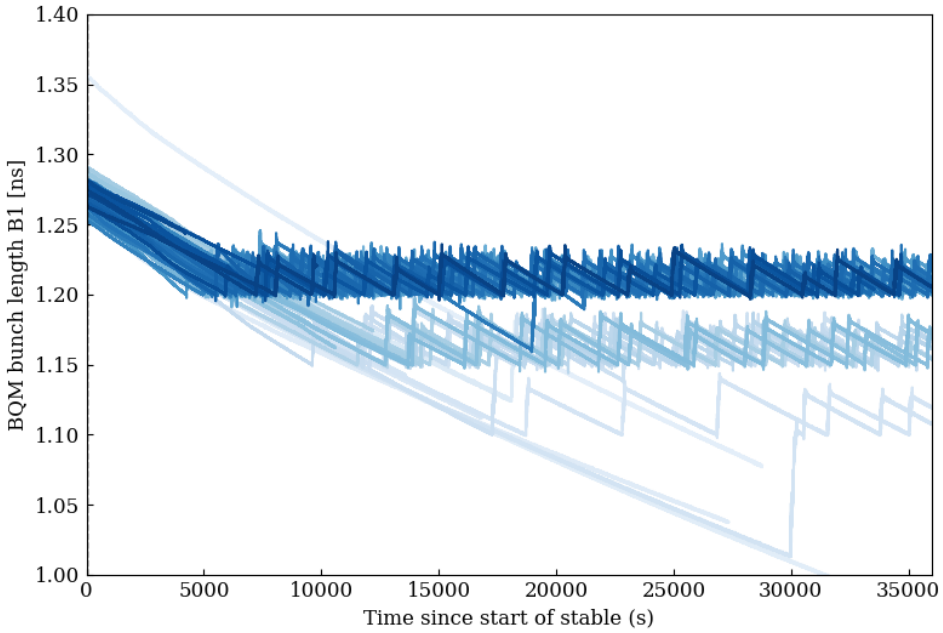
- Bunch length along the ramp in 2023 for fills with >= 200 bunches (and stable beams) – shown below.
- The lower bounds are quite reproducible.
- **With respect to 0.8 ns**, there is quite some **margin**: proposal **to raise the threshold to 1.05 ns** for 2024 (for >= 300 bunches). To be updated based on further experience, or improved blow up etc.

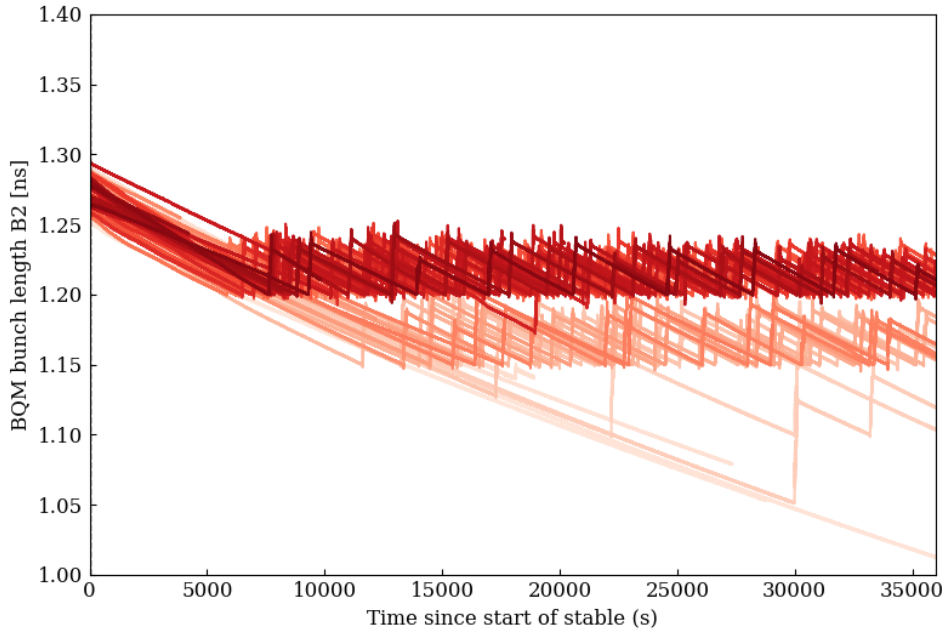




Bunch lengths in 2023 – stable beams

- Bunch length in 2023 during stable beams for fills with >= 200 bunches.
- One can see how we progressively raised the bunch length for which the blow up is triggered.
- **Automated blow-up trigger process** (UCAP) is **very effective**.
- One could consider **extending the SIS interlock to stable beams**, and/or add a **BigSister warning message** when the length drops below XX ns.
  - Interlock with some protection against BQM outages (changes are very very slow).





SIS V2

- BE-CSS has allocated resources (2 staffs at xx %) to prepare a **new version of SIS**, **SIS V2**.
- The aim is to clean-up old technology items (JMS, Groovy ….), simplify the code that grew over the years, change the way tests are declared etc.
- BE-CCS is working on a **first prototype for the injectors** to be testable in **2024** in parallel to the existing system, an activity that started without FB from OP.
- This subject was discussed in an **OP technical meeting** in January (with BE-CSS) with presentations by OP and CSS.
  - It was clear that the **OP and CCS ideas and concepts were not fully aligned**.
  - OP pointed out the need for good monitoring of input data (UCAP weaknesses), for tools to regenerate large lists of identical interlocks (for ex LHC PCs), etc
- As an **outcome of the meeting**:
  - CSS agreed that **UCAP** deserves **more development**, for ex. “administration/monitoring” tools.
  - A **CSS-OP team with representatives from all machines** will work on **requirements and implementation of SIS V2**.