
CERN

OP – Shutdown Lectures

Headtail Monitor

Kevin Li, Tom Levens

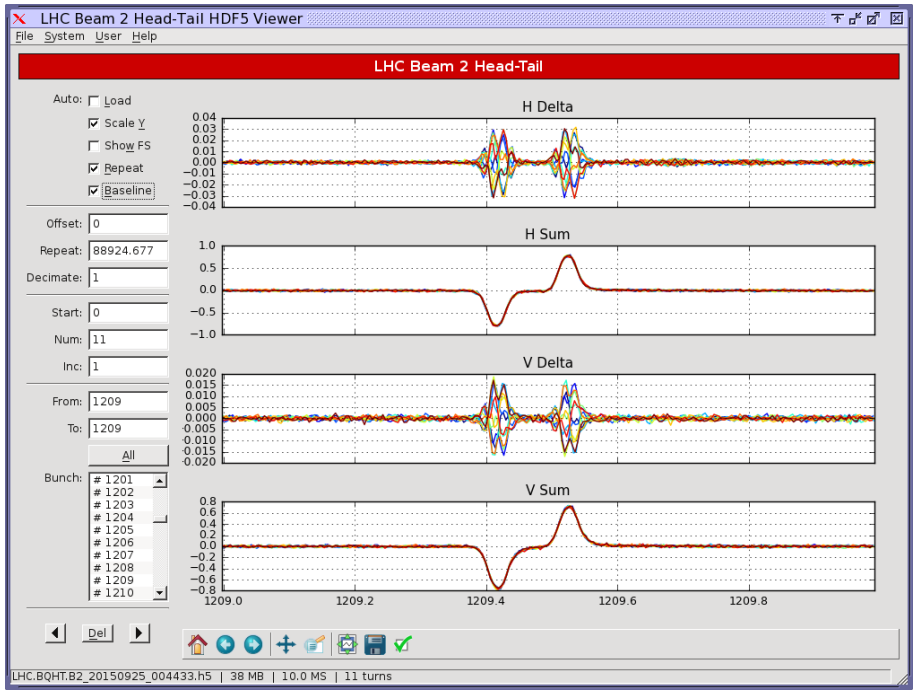
Overview

In both the SPS as well as the LHC the **Head-Tail monitor** is an **important tool** especially for **instability detection and characterization**.

It is not routinely used during operation, finding utilization mainly during MDs or by experts (e.g., the LHC instability team). The operational interface can be intimidating and the data model is slightly more complex.

The **goal of this lecture** is to give some motivation for as well as insight into the usage and the analysis of the Head-Tail monitor and its data for the example of the SPS.

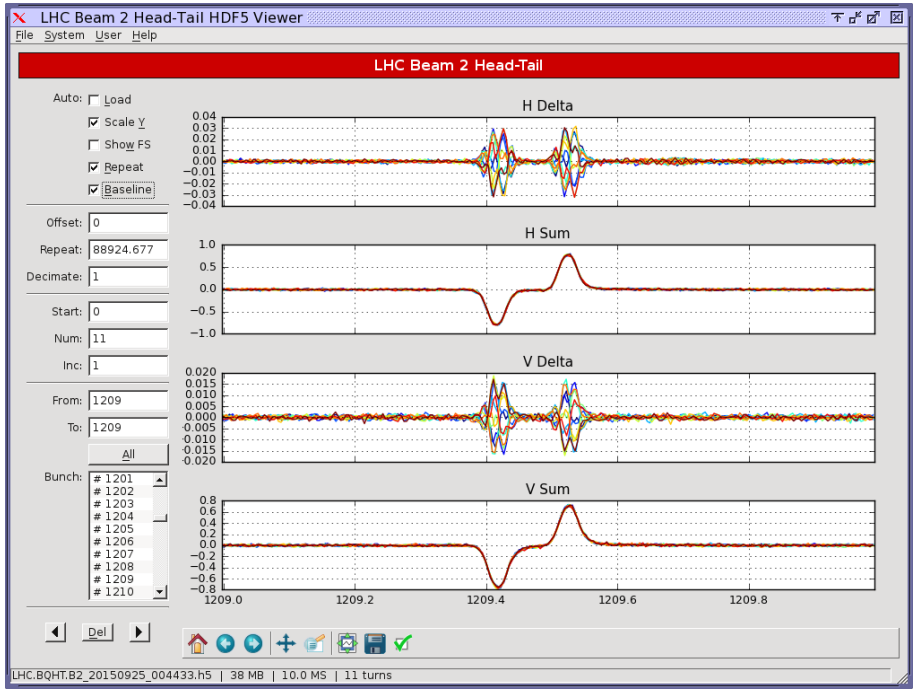
- Content
  - Background
  - Instrument / Usage
  - Post-processing (HDF5)
  - Examples / Gallery



Overview

We will first get some **background information on the Head-Tail monitor** and try and **answer some basic questions such as**, what is the Head-Tail monitor? Why do we need a Head-Tail monitor; what is it used for?

- Content
  - Background
  - Instrument / Usage
  - Post-processing (HDF5)
  - Examples / Gallery



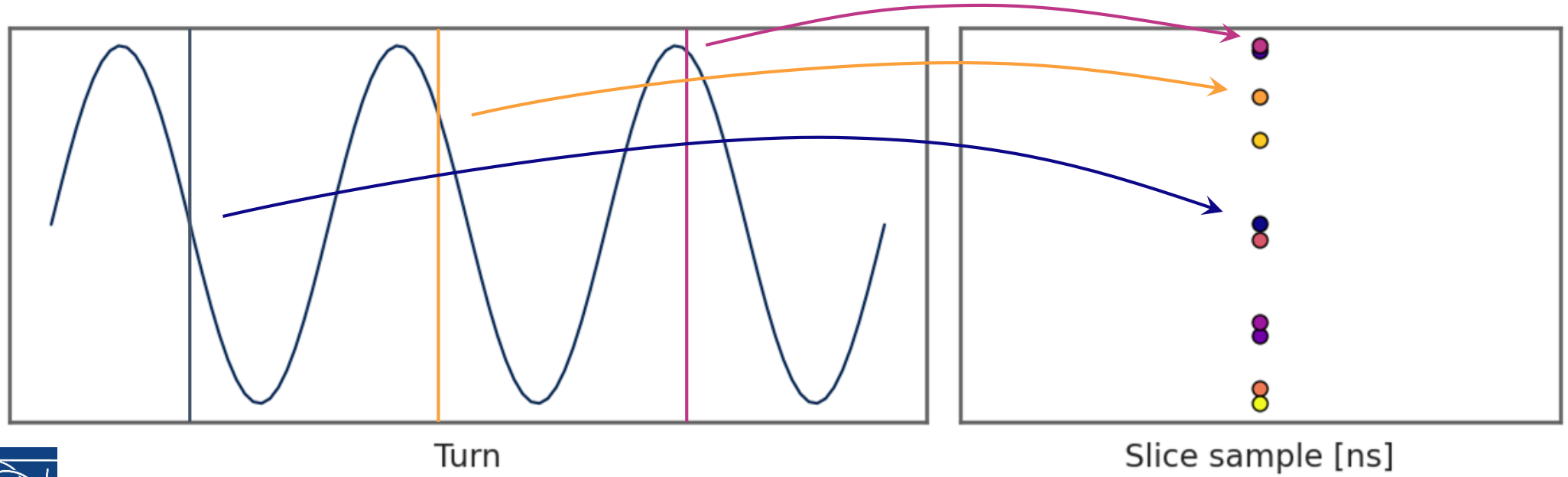
Background – definition and usage

So, what is a Head-Tail monitor anyway?

Background – definition and usage

Well Ernie, you know about what **a standard BPM** is supposed to do, right?

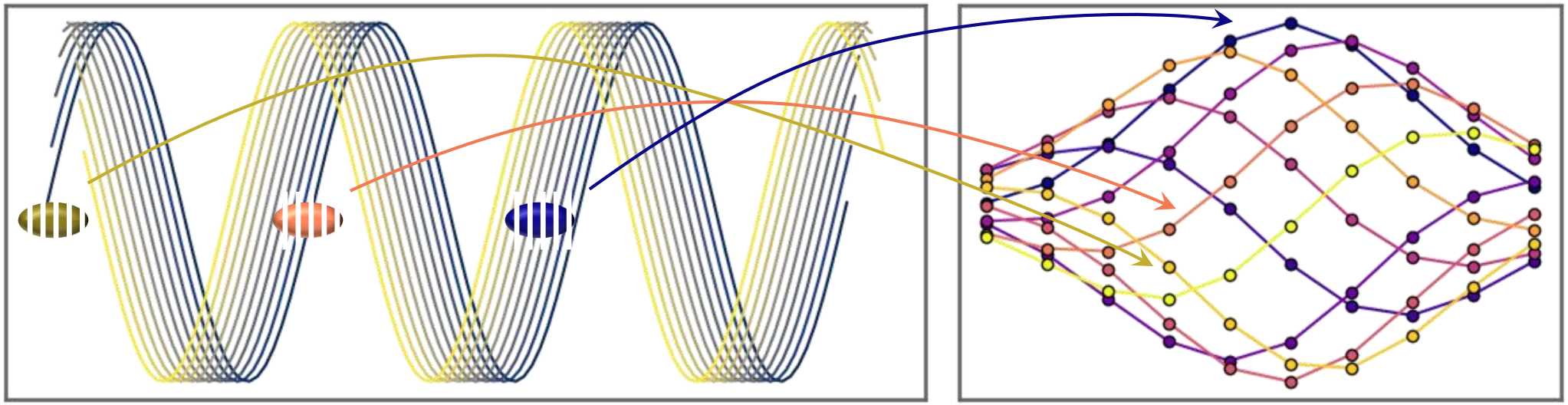
It just measures **the bunch centroid position** in the horizontal and the vertical plane. We can plot this over several turns and it would look something like:



Background – definition and usage

You see, the **Head-Tail monitor** in the SPS is a high bandwidth pickup with a fast digitizer and can be regarded essentially as a **high bandwidth version of a standard BPM** resulting in an **intra-bunch BPM**:

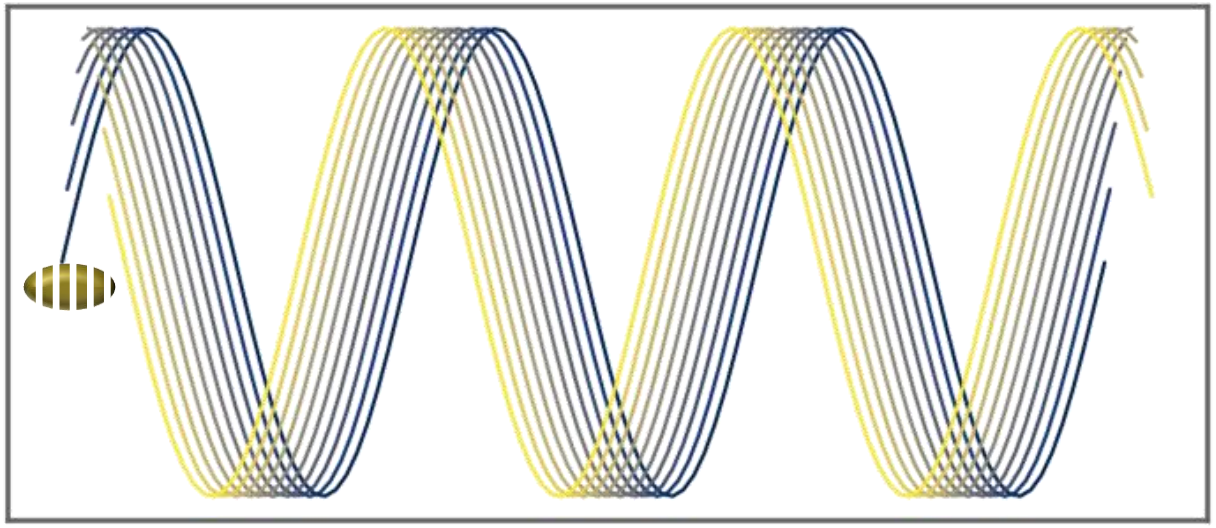
If we now plot this over several turns, this would rather look something like:



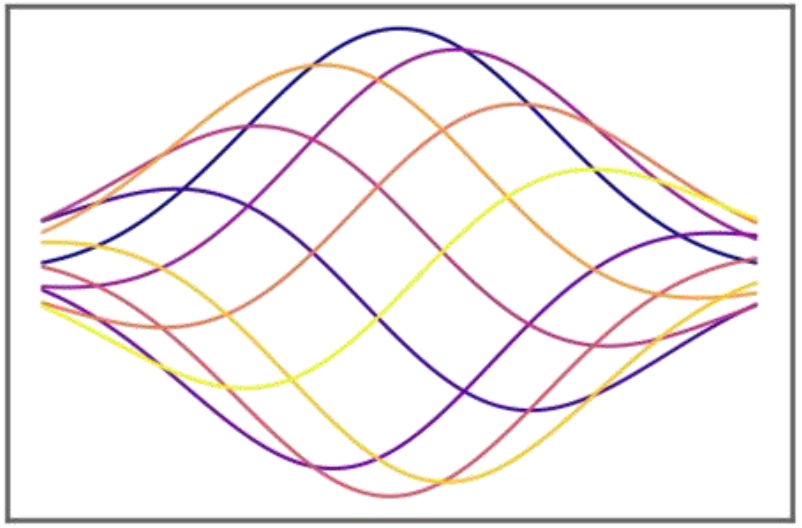
Background – definition and usage

You see, the **Head-Tail monitor** in the SPS is a high bandwidth pickup with a fast digitizer and can be regarded essentially as a **high bandwidth version of a standard BPM** resulting in an **intra-bunch BPM**:

If we now plot this over several turns, this would rather look something like:



Turn



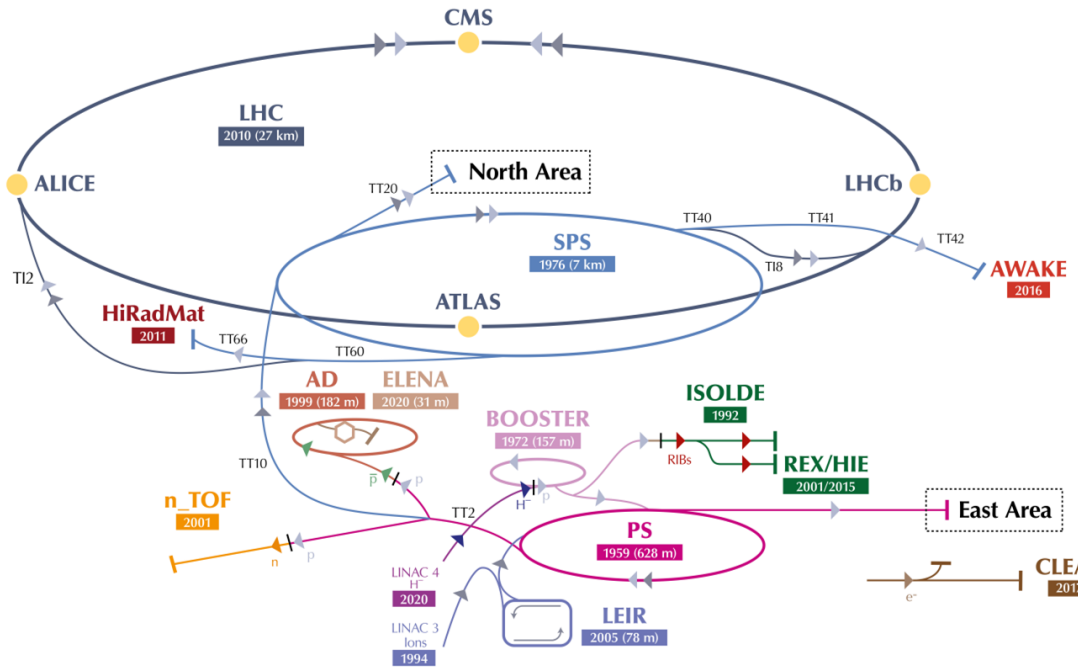
Slice sample [ns]

Background – definition and usage

Oh, I see… but apart
looking admittedly pr
would I need such sig
the centroid position

Ah Ernie… you are young and have yet much to learn.

I will tell you something about an evil force, that latently reigns in every high intensity, high brightness accelerator – it manifests itself in what we see as **coherent instabilities**!



Harharhaaar!

Time to try out my new performance limitor!

Background – instabilities

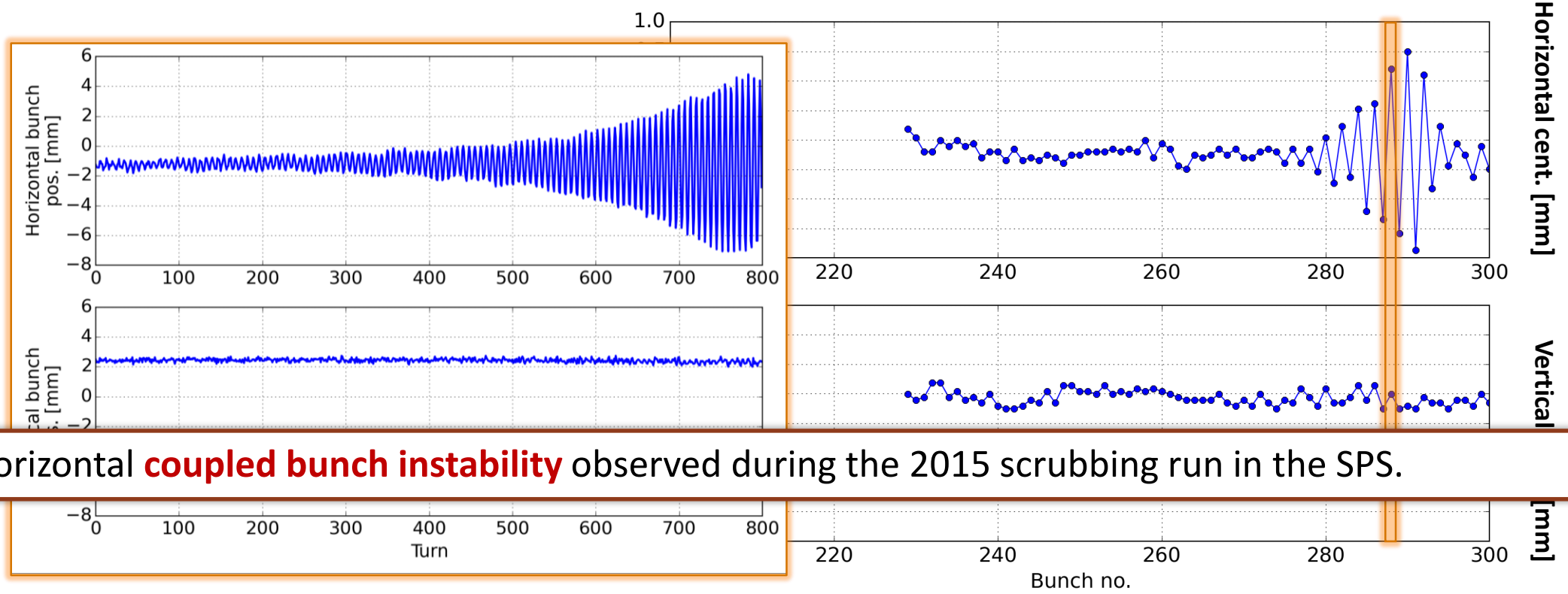
- Coherent instabilities are one of the **fundamental limitations in particle accelerators** – they limit the maximum achievable intensity

- There are **different sources for coherent instabilities**, i.e., impedances, electron clouds, beam-beam interaction,…
- There are also **mitigations for coherent instabilities**, i.e., feedback systems, machine optics, chromaticity, Landau octupoles,…

- Understanding the coherent instabilities of a given machine **is essential for understanding its limits** and **devising possible mitigation measures**
- Being able to **monitor and characterize coherent instabilities is a key asset** – in the SPS, we have the **Head-Tail monitor** for this

Background – coupled bunch instabilities

- Coherent instabilities **are characterized by a time constant pattern** developing along and across bunches which tends to grow exponentially in amplitude
- In the transverse plane, this can lead to emittance growth or even to beam loss, as particles start hitting aperture limitations
- Coupled bunch instabilities **feature a coherent pattern developing across a full bunch train** – these can be nicely observed using standard BPMs



Horizontal **coupled bunch instability** observed during the 2015 scrubbing run in the SPS.

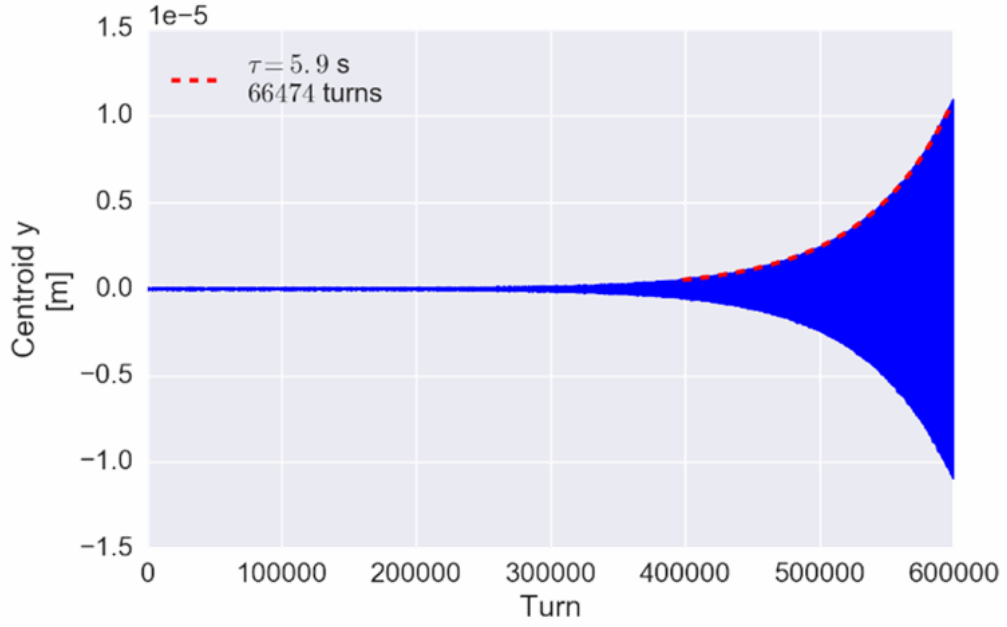
Background – single bunch instabilities

- Single bunch instabilities feature a coherent pattern developing across a single bunch
  - These patterns can be rather feature rich and can allow to characterize the respective instability
  - A standard BPM will **only display the centroid motion of this bunch** and hide these features
  - A headtail **monitor is needed to be able to "look into the bunch"**

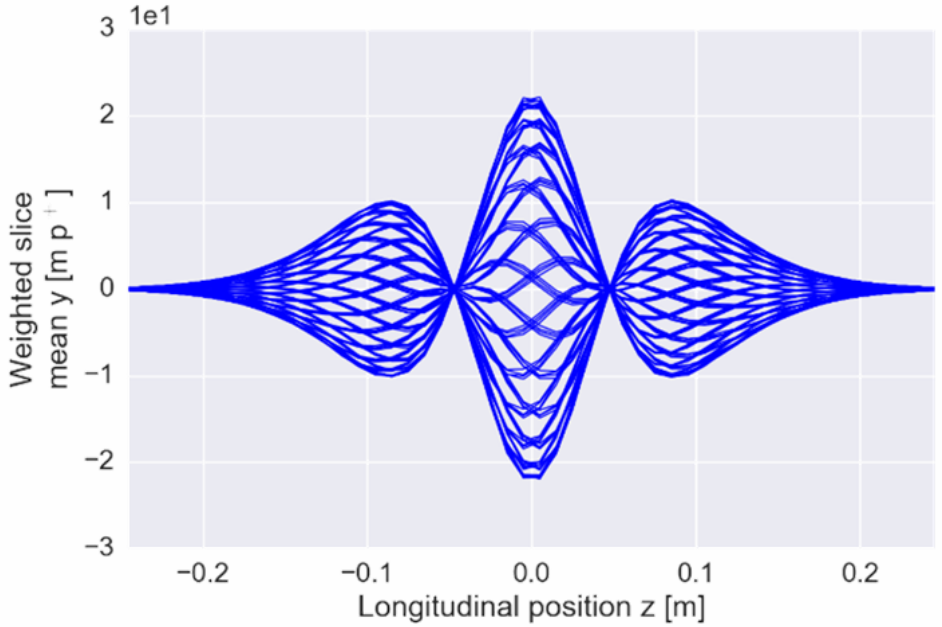
Okay… it's more or less clear to me, how to imagine a coupled bunch instabilities – so for single bunch instabilities, **what are these "features"**?

Background – single bunch instabilities

- LHC simulated headtail instability:



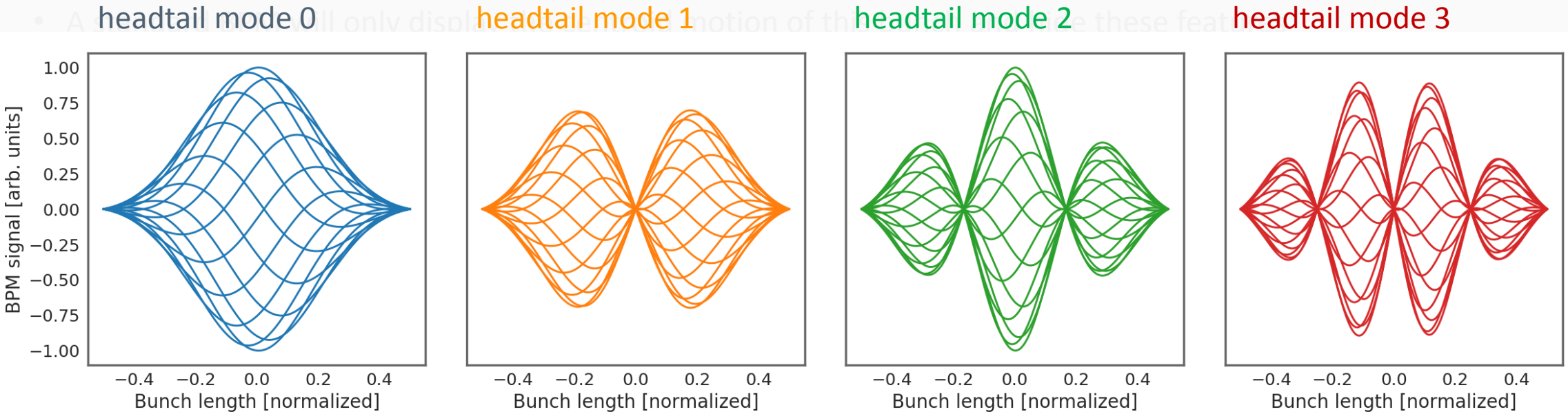
Centroid motion amplitude growth as detected by a standard BPM



Turn-by-turn traces along a single bunch of an LHC simulated headtail instability

Background – single bunch instabilities

- Different headtail modes can develop under different conditions (impedance, chromaticity,…)



- Knowing the headtail instability modes can help **to benchmark our understanding of the machine** (impedance models, instability sources) and find adequate mitigation measures

- The **only tool we have today** that can reliably measure and visualize these instabilities is **the Head-Tail monitor**

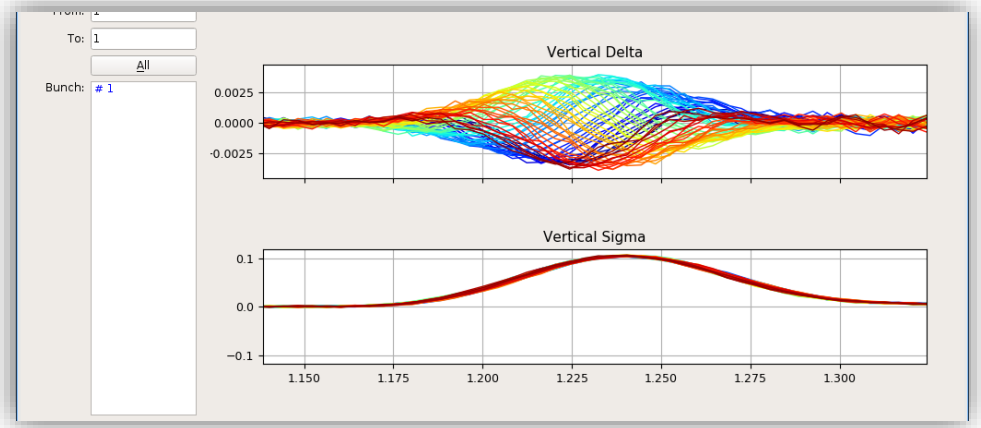
Background – instabilities

- The Head-Tail monitor, due to its ability to “look into the bunch” **is an important tool for instability characterization (especially used for this purpose also in the LHC)**

- We have spent some time highlighting this use case – we want to stress, however, that this ability of the Head-Tail monitor **to “look into the bunch” leads to many other (maybe less obvious) use cases**. Unfortunately, we can not detail all of these further here, but we would like to mention some other 2018 use cases:

  - Head-Tail monitor used during **crab cavity MDs** to detect “crabbing”

  - Head-Tail monitor used during **chromaticity measurement MDs** to extract chromaticity from head-tail phase shift

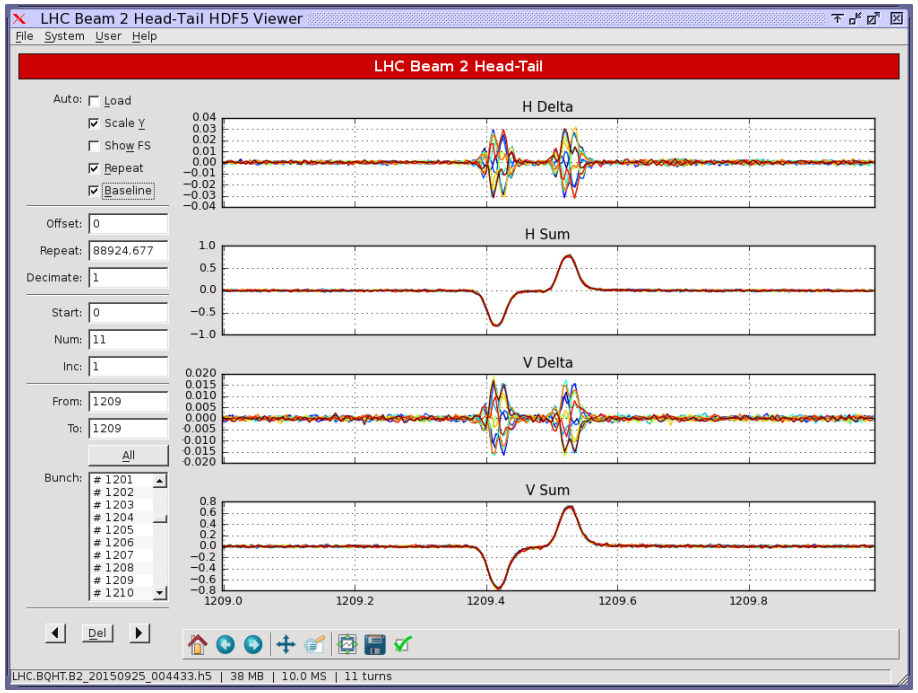


Signpost

We have seen that we can view the Head-Tail monitor as high bandwidth BPM **displaying the intra-bunch motion turn after turn**. We have also seen that this is **essential for the characterization and the understanding of coherent instabilities**, which is one of the important performance limitations in our accelerators. We have also seen it's usefulness for other use cases.

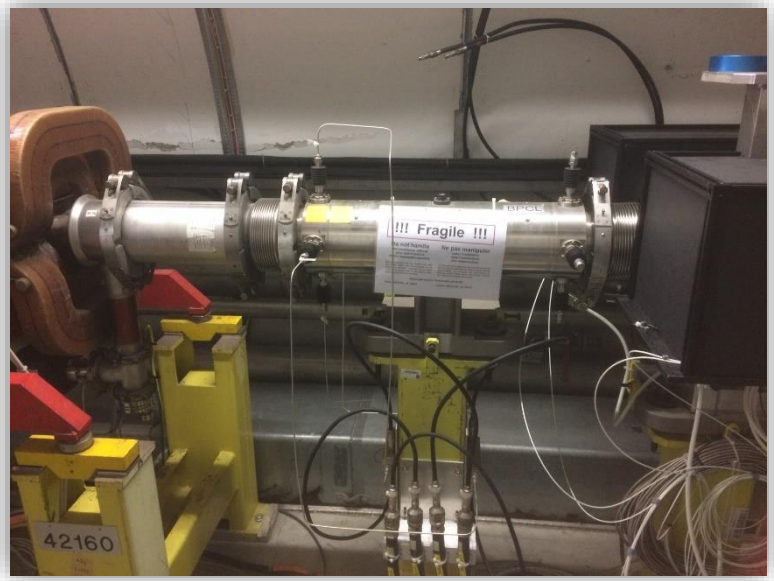
We will now look into a few more details on the Head-Tail monitor, in particular also it's **operational interface and usage**.

- Content
  - Background
  - Instrument / Usage
  - Post-processing (HDF5)
  - Examples / Gallery



Headtail monitor – the instrument

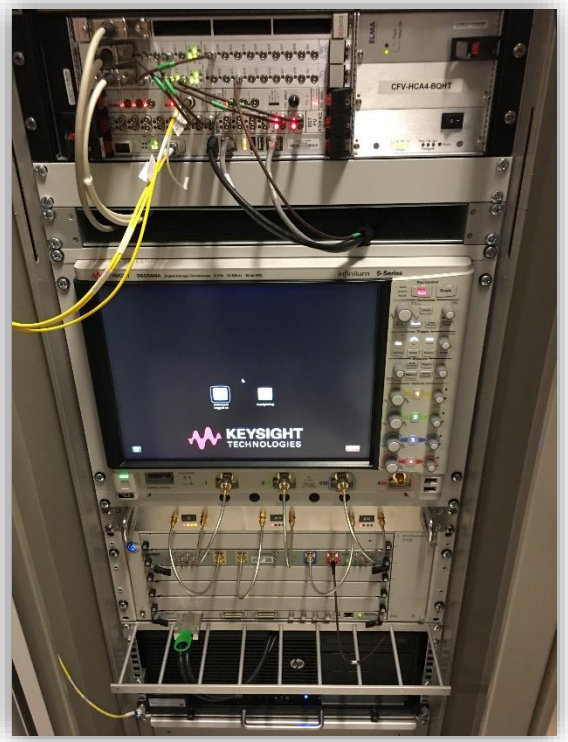
- The headtail monitor is located in BA4 (BPCL.42171); it delivers bunch-by-bunch (actually slice-by-slice), turn-by-turn data at a high sampling rate – to obtain bunch- and turn-synchroneous data it uses an elaborate trigger system.



Device

- The Head-Tail digitizers are triggered by a number of CTRV channels which are used to generate a beam synchronous trigger.

- The triggering is **one of the key items that need to be correctly set up** from the Head-Tail control.

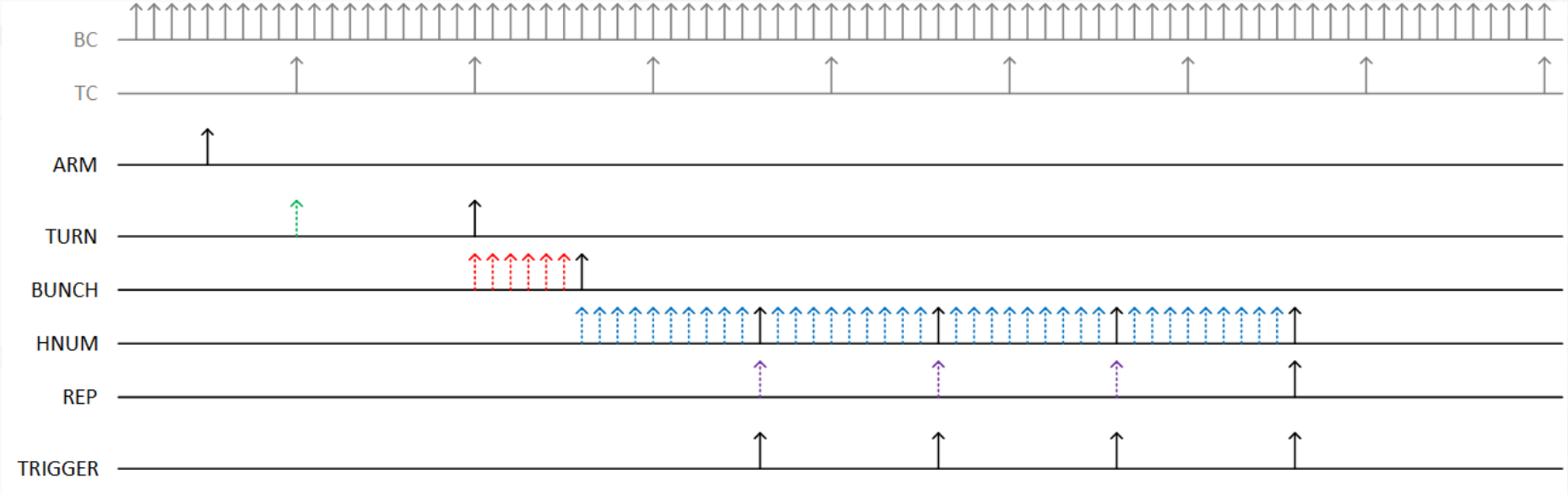


Digitizers / Trigger

Headtail monitor – the instrument

- The trigger is generated in a number of steps:
  - Each possible source (central timing, instability trigger, pre-pulse) which can trigger the scope has an individual CTRV channel. The output of these channels is ORed using the CTRV OMask to generate a single ARM pulse.
  - **A TURN counter, clocked by the turn-clock (TC) , synchronises the ARM to a turn and (optionally) delays the trigger by a number of turns.**
  - **A BUNCH counter, clocked by the bunch-clock (BC), delays the trigger to a specific bunch within the turn.**
  - **A Harmonic-NUMber counter, clocked by BC, generates a pulse every turn (or multiple turns).**
  - **A REPetition counter stops the HNUM counter after a certain number of pulses.**

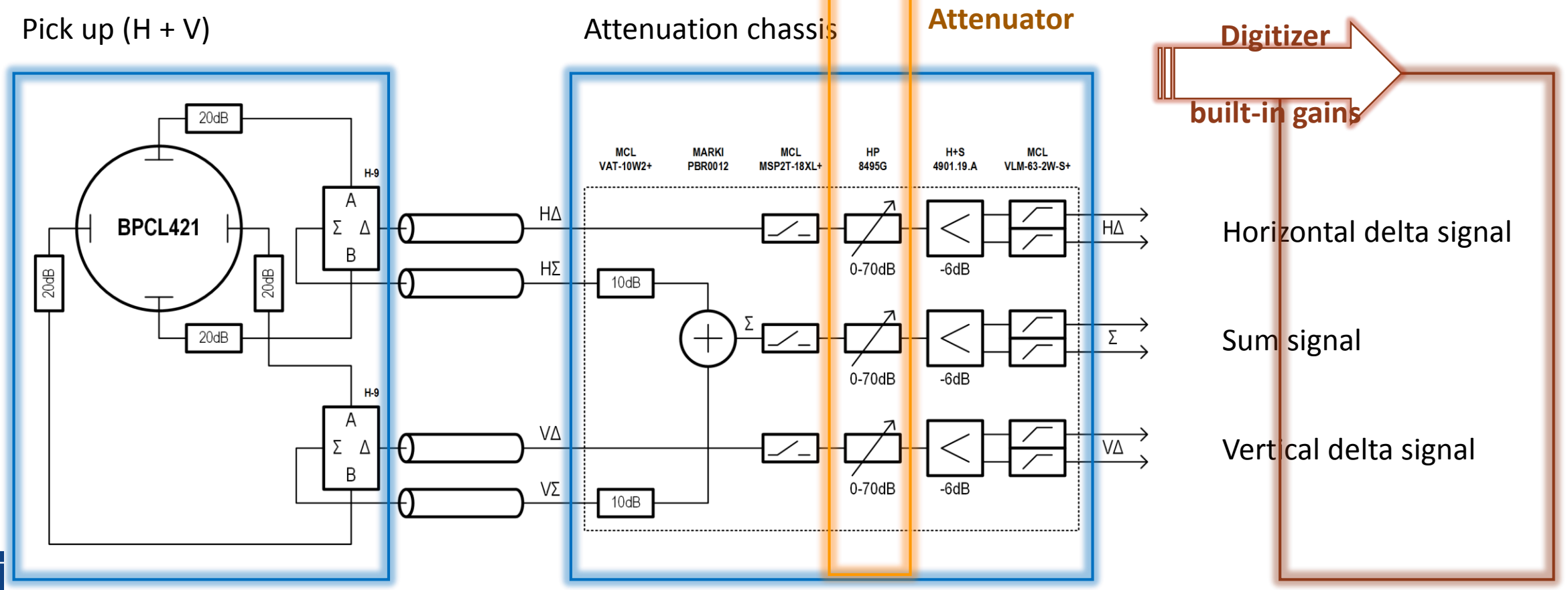
- The scope is triggered by the HNUM counter. The sequence is shown in the following diagram:



Headtail monitor – the instrument

- High bandwidth BPM and acquisition system using a high speed digitizer

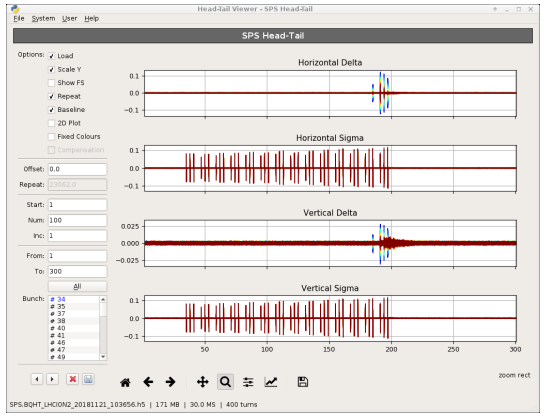
Knobs accessible via GUI → other key item that needs to be correctly set up from the Head-Tail control

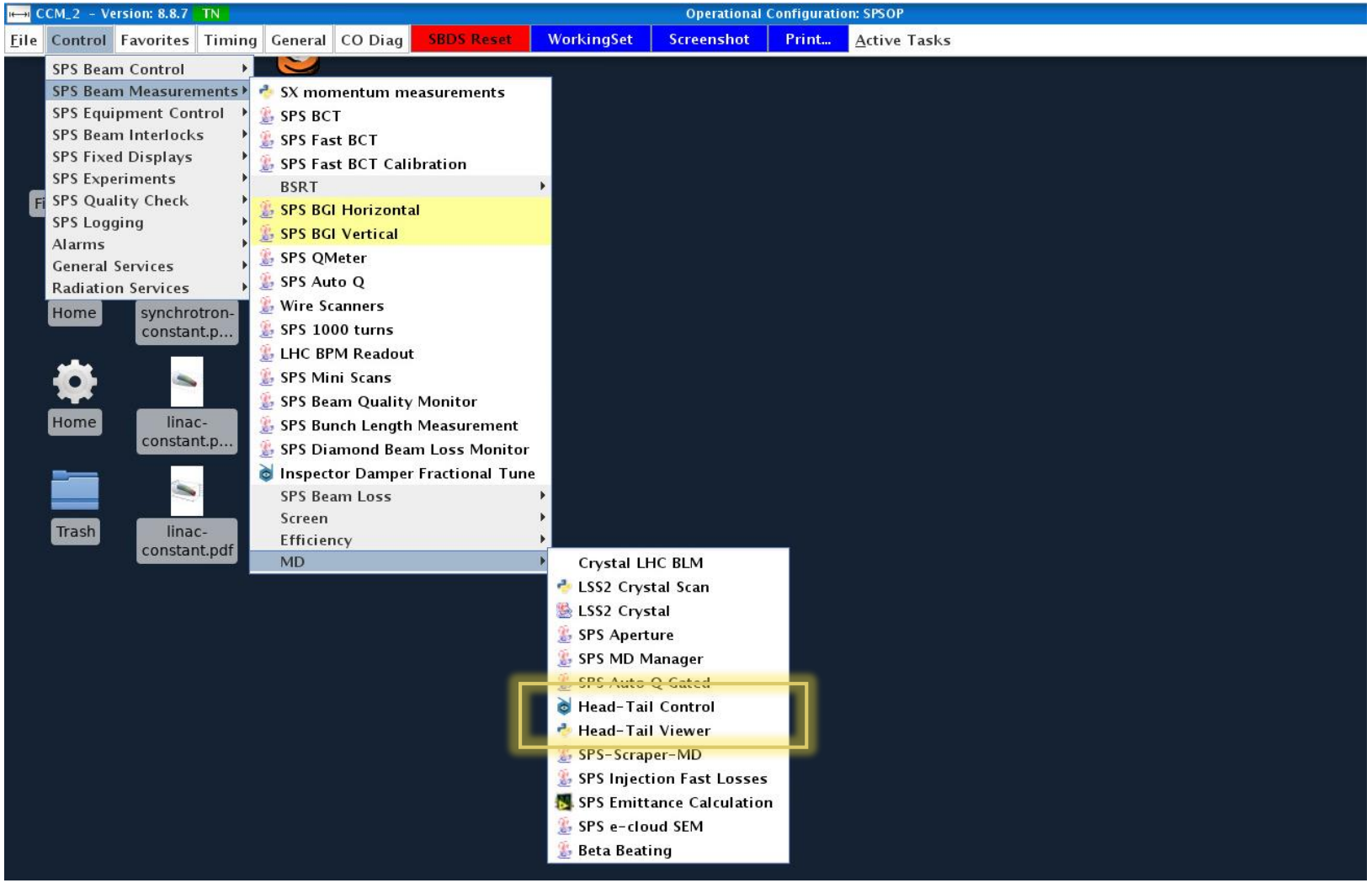


Headtail monitor – the application

- The Head-Tail operational GUI is split into separate control and viewer applications:
  - Head-Tail Control (Inspector)

  - Head-Tail Viewer (PyQt)

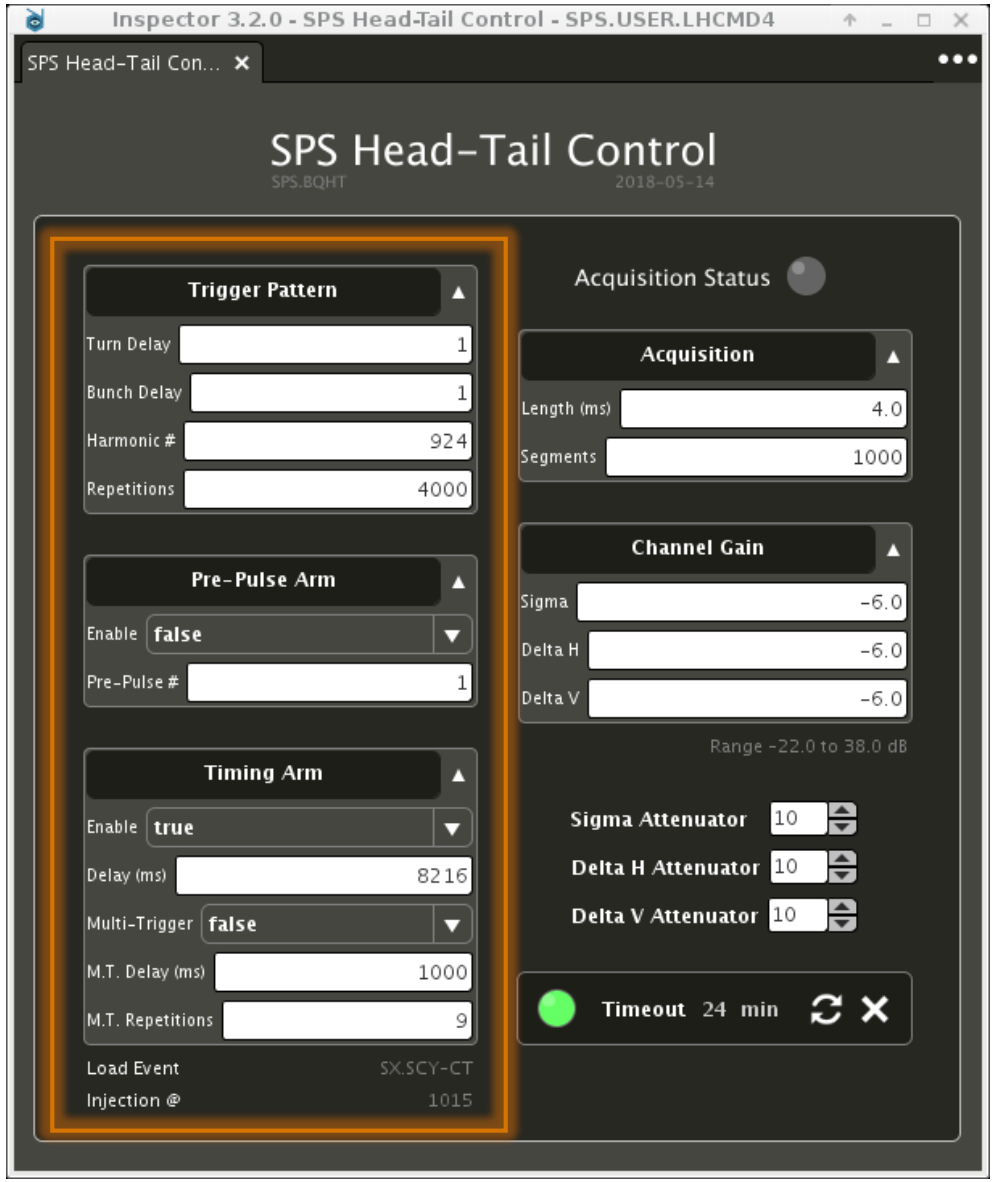




Head-Tail monitor – control panel

- **NOTE: to write a setting you must click on the darker "title" bar of each section.**

Section	Description	Sections	Description
Trigger Pattern	Controls the pattern of generated triggers	LED	Shows acquisition status (grey = idle; orange = triggered; green = data readout; red = error)
Pre-Pulse Arm	Enable arm on a specific injection/extraction pre-pulse	Acquisition	Controls the acquisition length and segmentation
Timing Arm	Enable arm on a SX.SCY-CT timing event (plus a delay in ms)	Channel Gain	Controls the gain (in dB) for each channel (0dB = 0dBm full scale)
		Timeout	Shows the timeout duration remaining, pressing the "refresh" button resets the timeout to 60 minutes



Head-Tail control – timing control

Trigger control – timing arm:

i. **Delay (ms):** the time in the cycle where the Head-Tail monitor will first trigger. (Note that injection occurs in this case at 1015 ms which is written at the bottom).

ii. **Multi-trigger (M.T.) Delay (ms):** the delay in time (ms) between each acquisition.

iii. **M.T. Repetitions:** how many times you want to repeat the acquisition in the cycle. If MT Repetitions = 9 then that means 10 total acquisitions in the cycle.

Head-Tail control – acquisition control

Trigger control – batch pattern:

i. **Trigger Pattern – Repetitions:** number of segment acquisition triggers

→ should be at least = number of acquisition segments

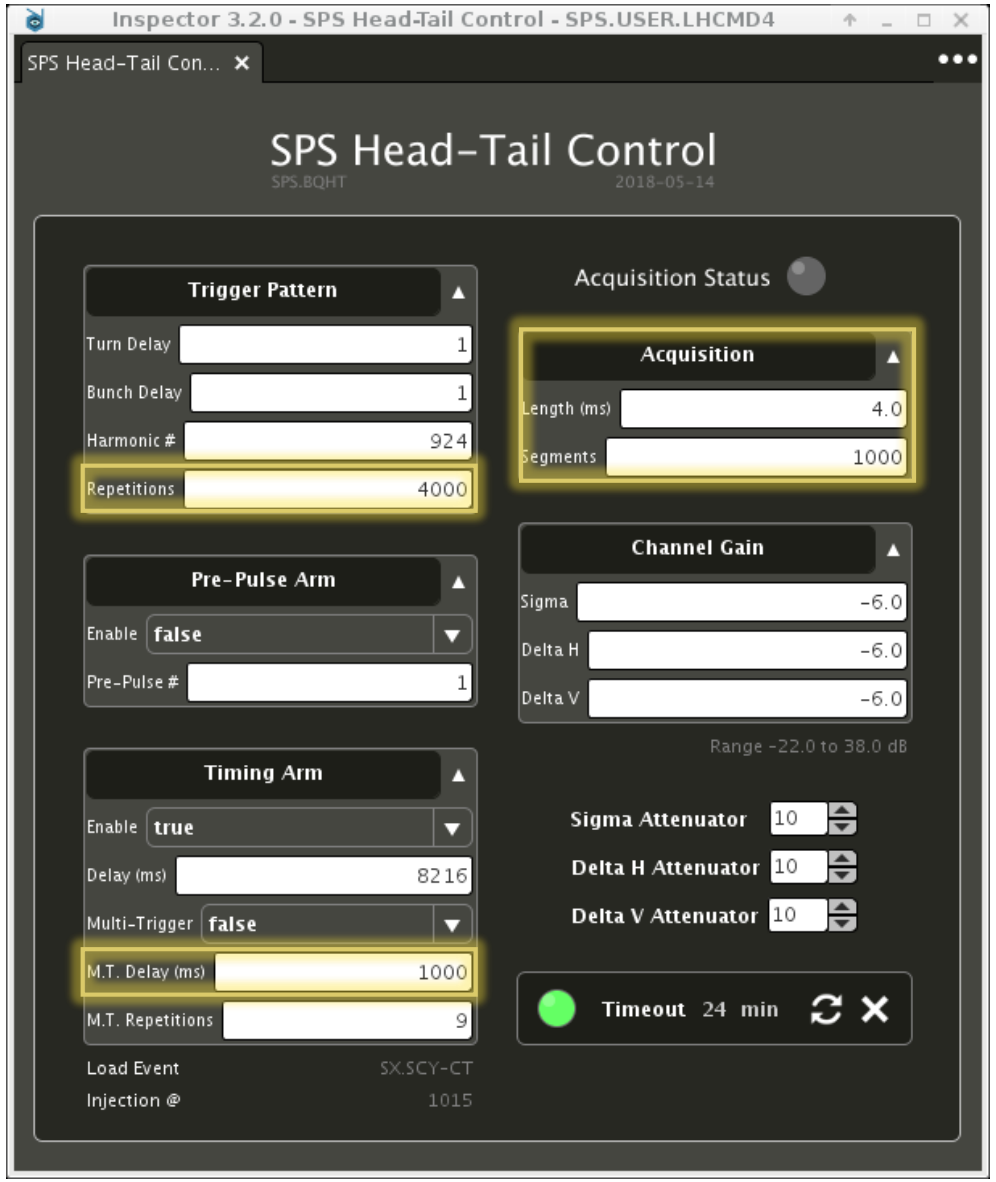
ii. **Timing Arm – M.T. Repetitions:** multiplier for trigger pattern repetitions

→ Total number of turn triggers =
Trigger Pattern Repetitions x (M.T. Repetitions + 1)

iii. **Acquisition – Length [ms]:** total acquisition buffer length

iv. **Acquisition – Segments:** number of segments the acquisition buffer gets split into

→ length of acquisition buffer segment determines number of displayed bunches



Head-Tail control – acquisition control

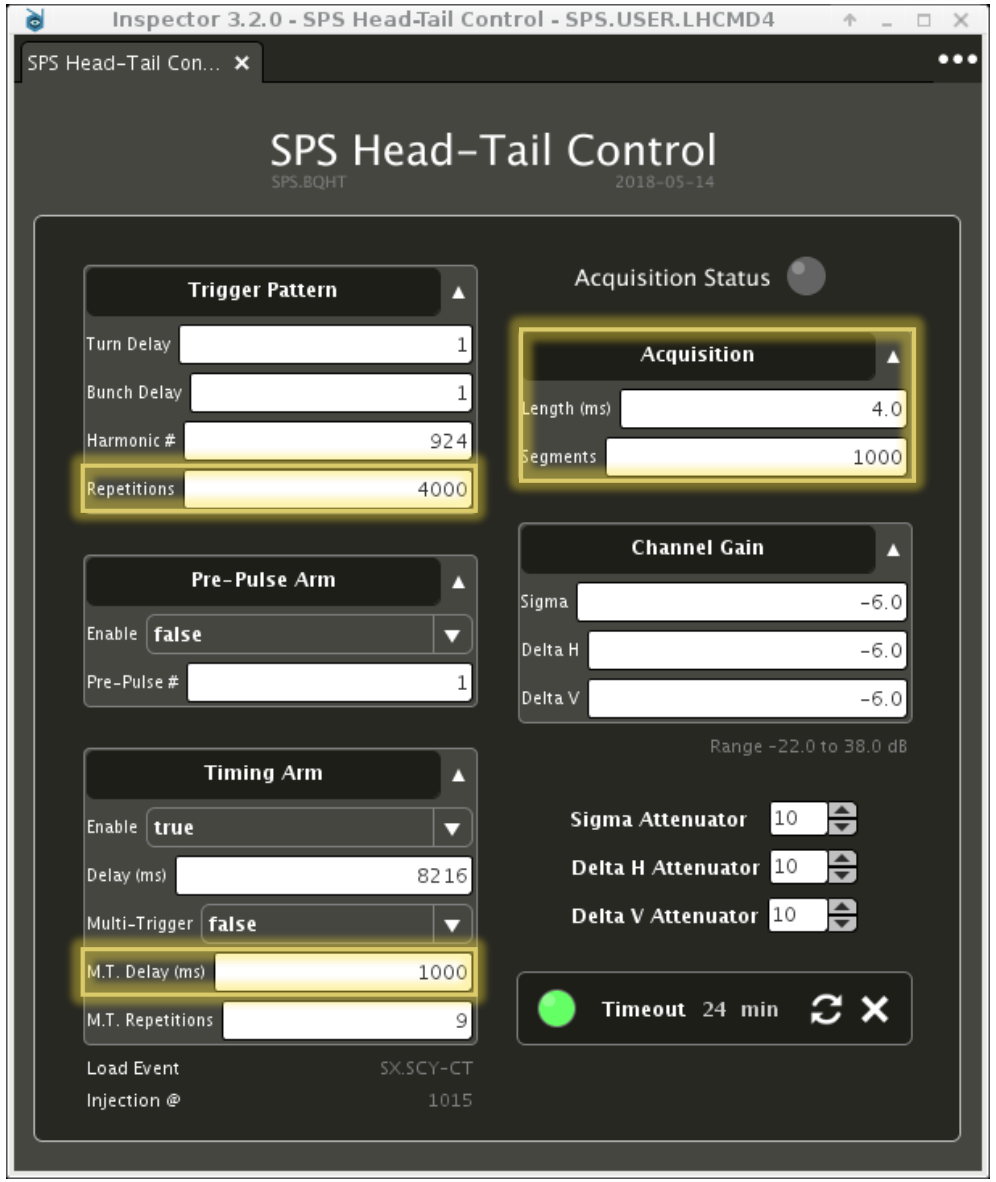
Trigger control – batch pattern:

This translates to:

- **n_turns = Segments**
- **n_bunch slots = Length [ns] / Segments / 25**
- **Trigger Pattern Repetitions = Segments / (M.T. Repetitions + 1)**

or

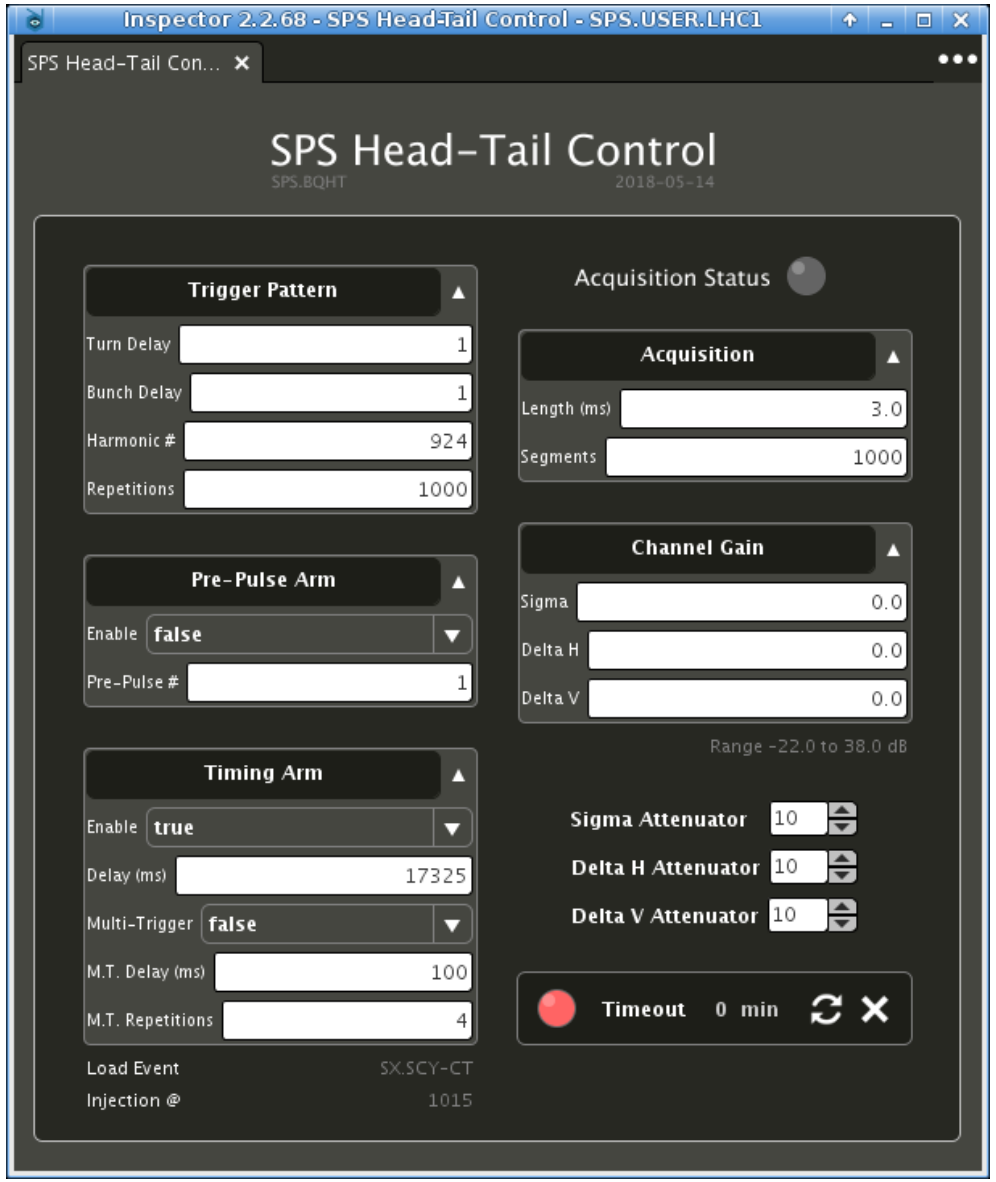
- **Segments = n_turns**
- **Length [ms] = Segments * n_bunch_slots * 25 / 1e6**
- **Trigger Pattern Repetitions = Segments / (M.T. Repetitions + 1)**



Head-Tail monitor – example acquisitions

- LHC1 instability tests:
  - Dump around 16330
  - Head-Tail monitor set up to capture last turns before dump.

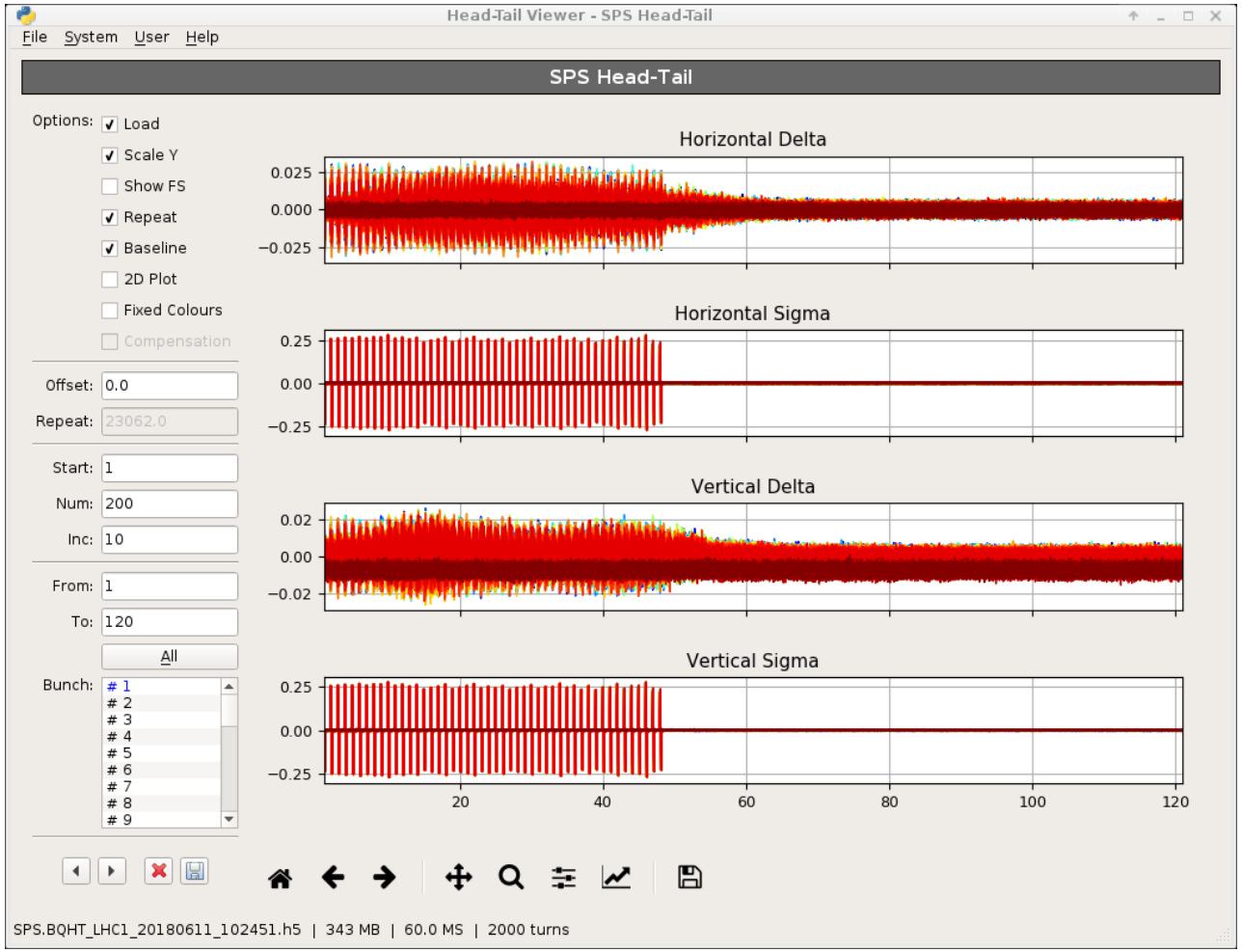
Acquisition @ 17325 – 1015 = 16310 ms

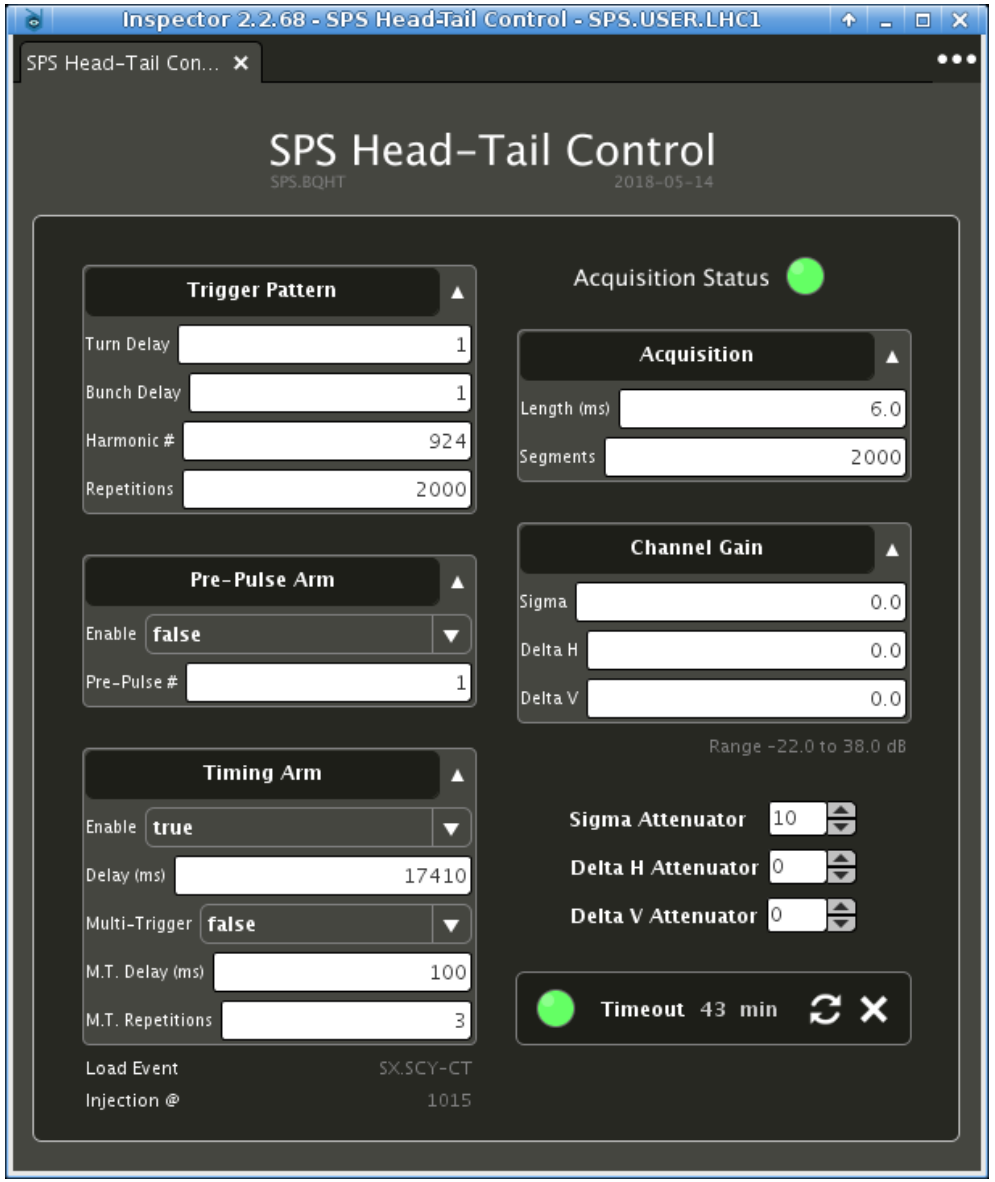


Head-Tail monitor – example acquisitions

- LHC1 instability tests:

slots = 6e6 / (2000 * 25) = 120

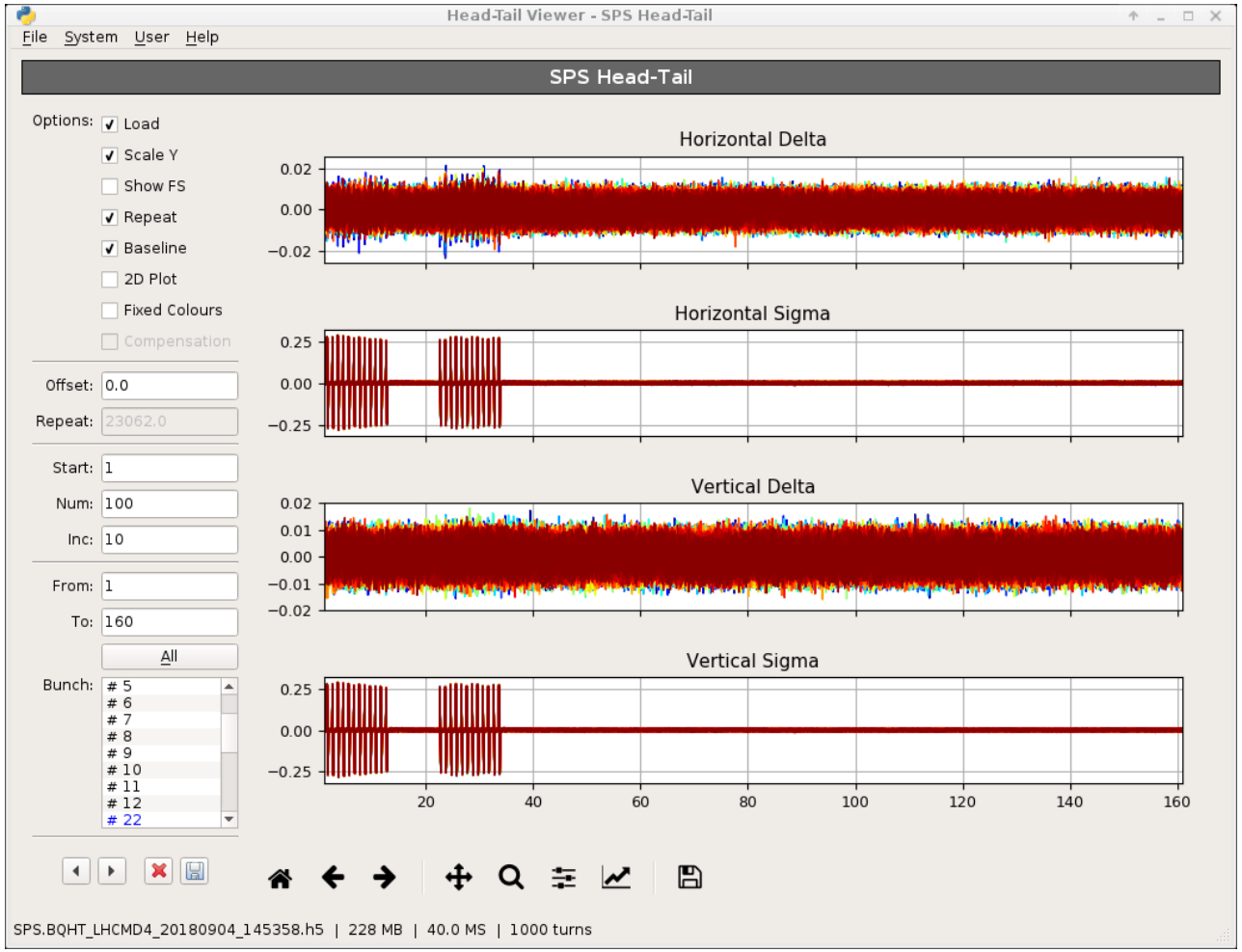


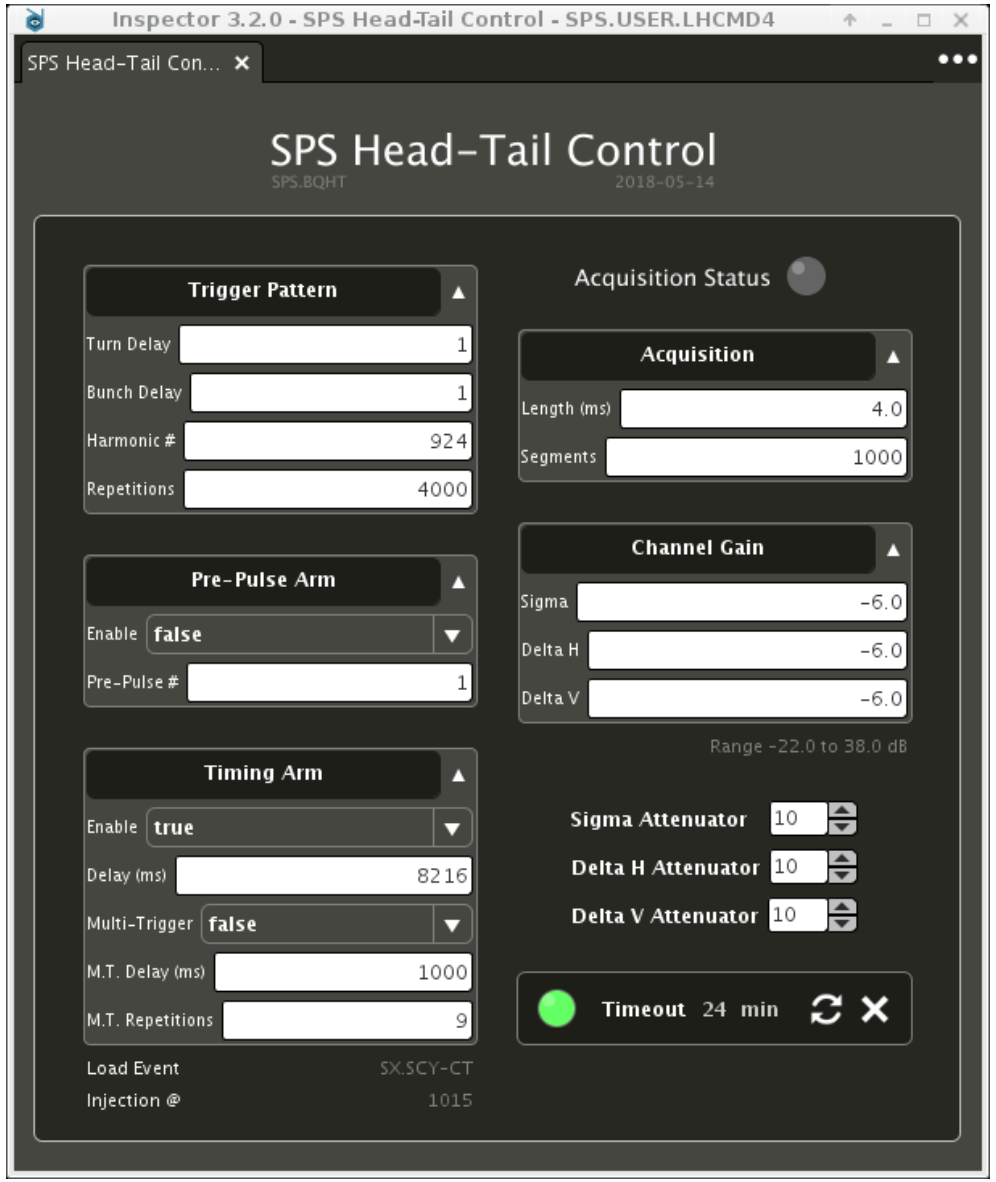


Head-Tail monitor – example acquisitions

- LHCMD4: Setting up the Head-Tail monitor for single acquisition after xth injection:

slots = 4e6 / (1000 * 25) = 160

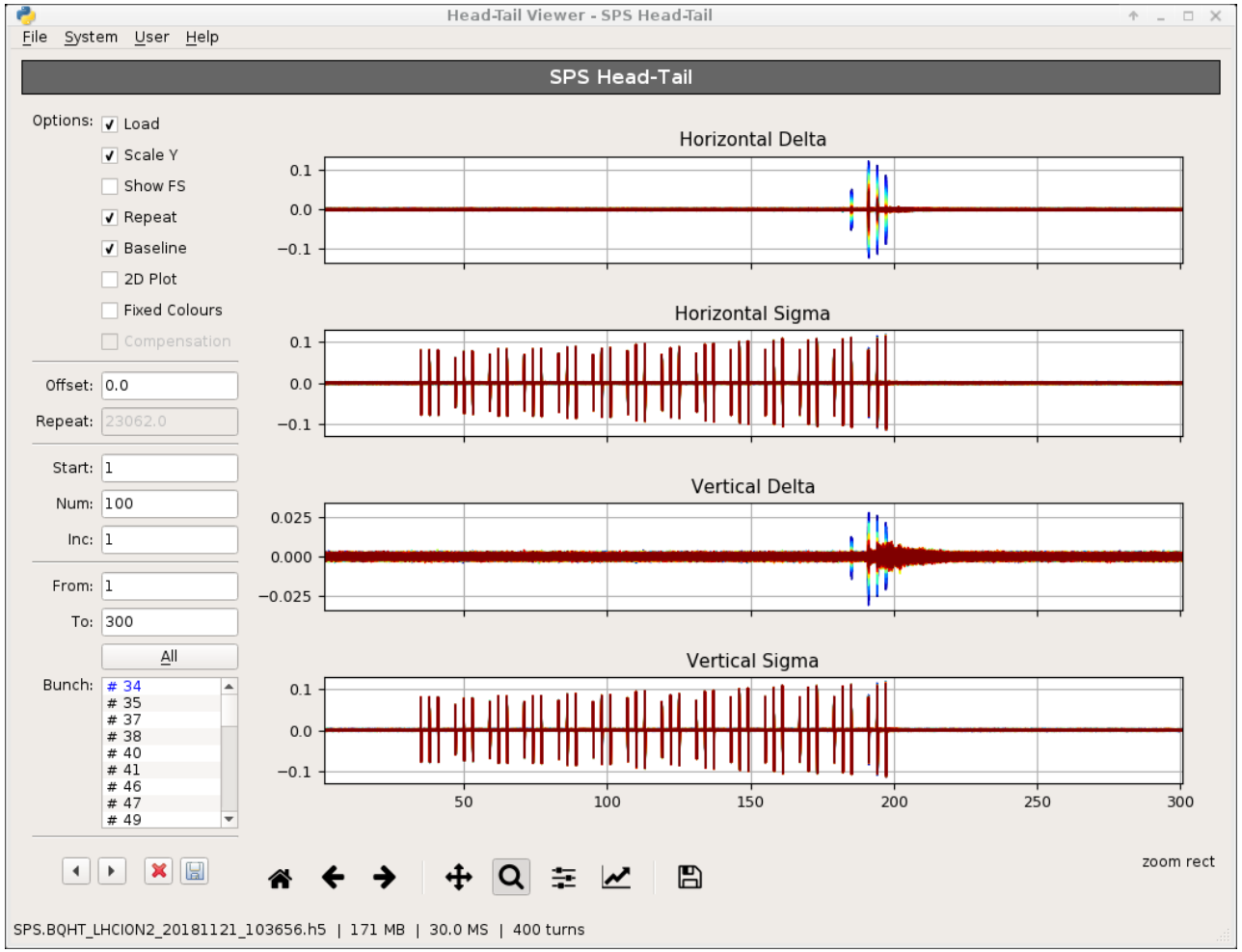


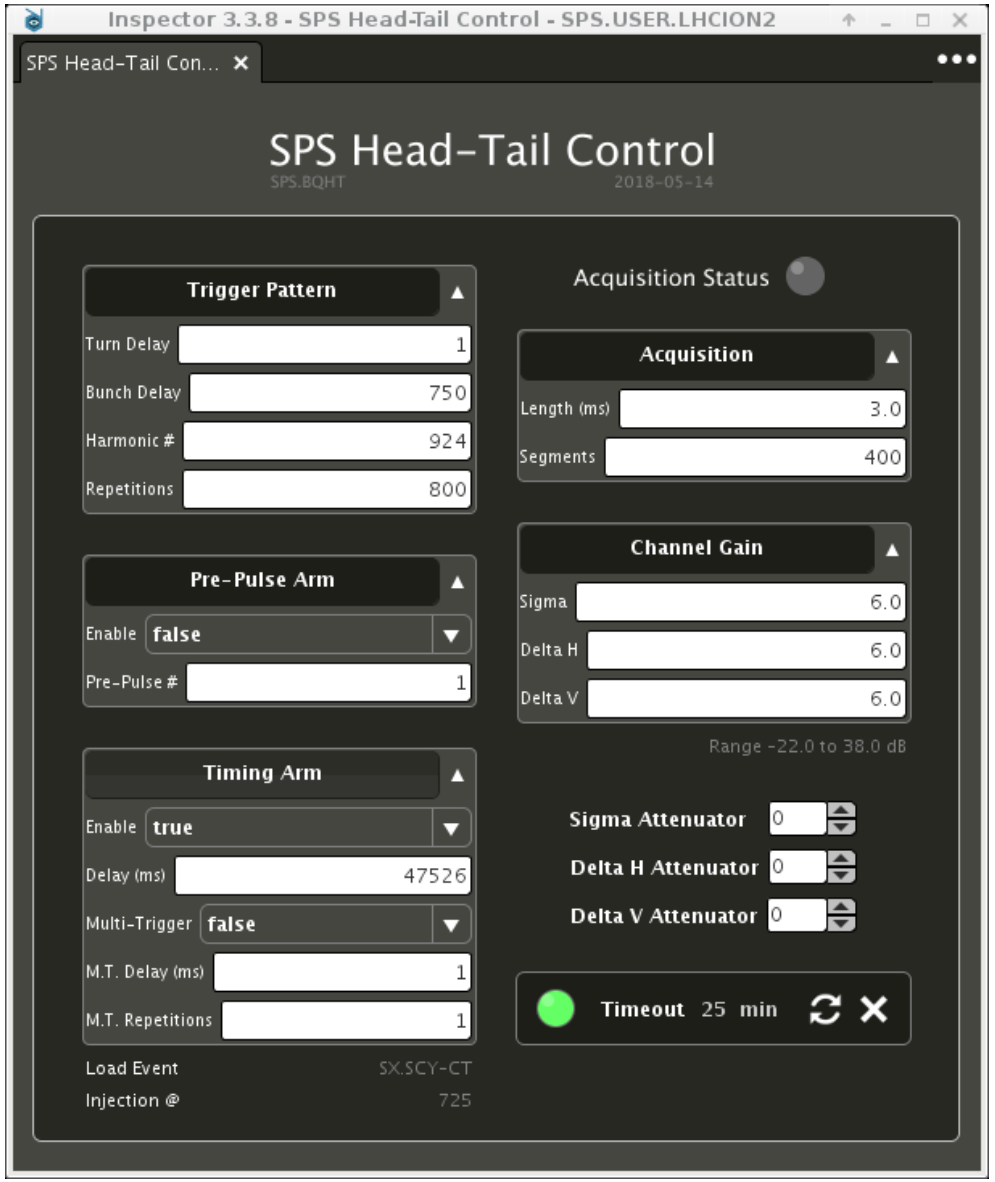


Head-Tail monitor – example acquisitions

- Head-Tail Monitor settings for ion acquisition:

slots = 3e6 / (400 * 25) = 300





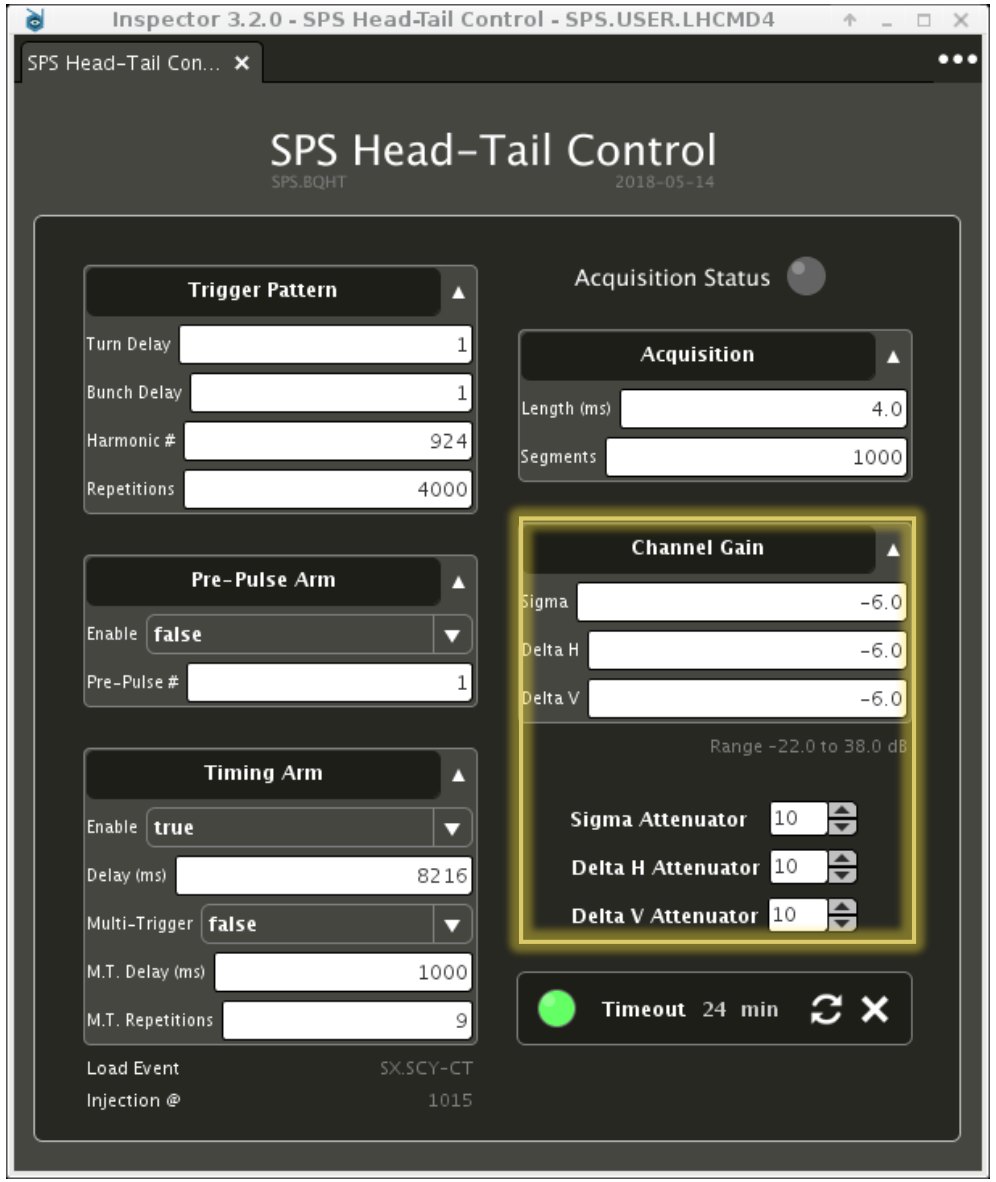
Head-Tail control – gain control

Gain control:

i. **Channel Gain:** gain per channel (Delta H / V & Sigma) from -22 dB to 38 dB.

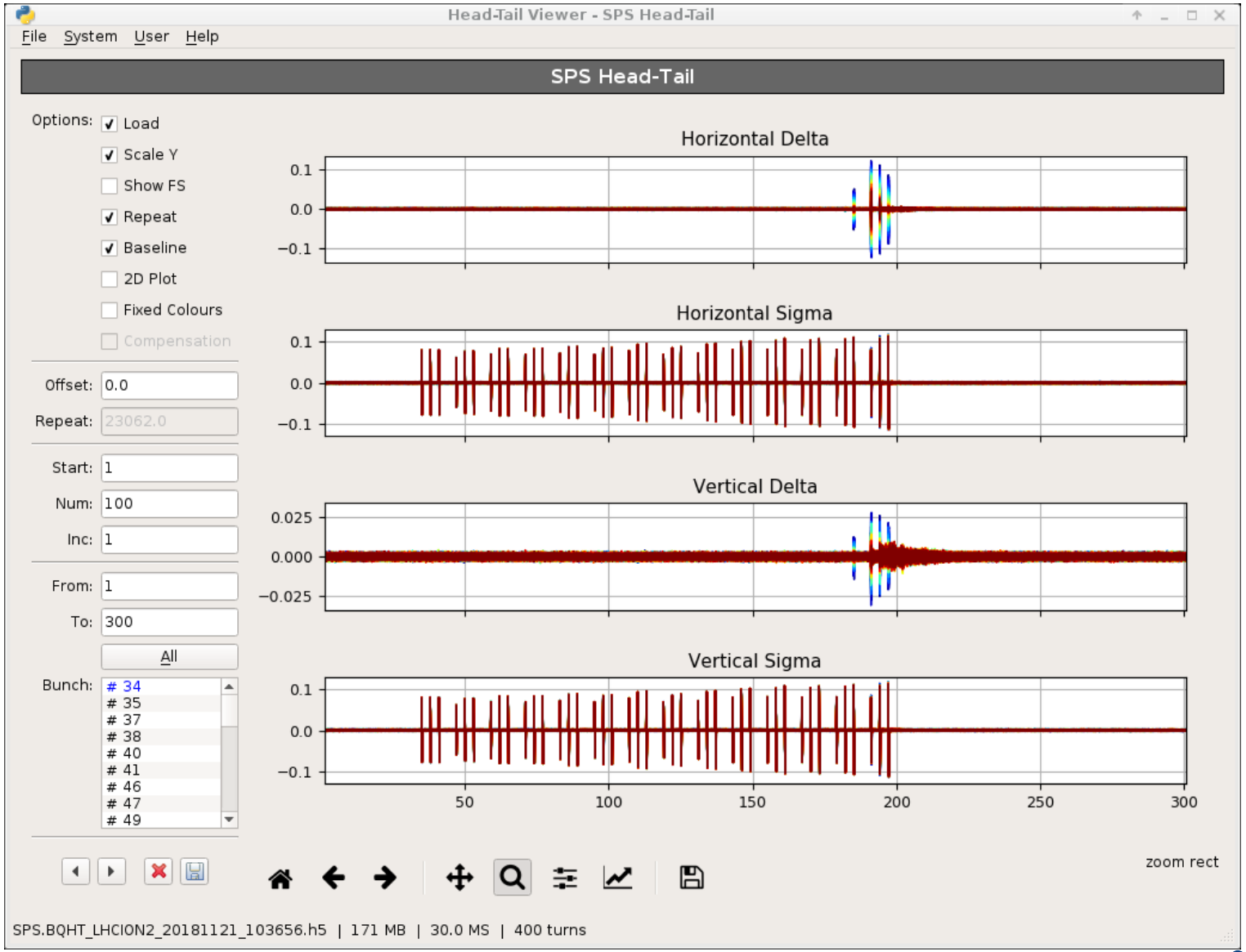
ii. **Attenuators (Delta H / V & Sigma):** available attenuation 5, 10, 20

Use gains and attenuations to increase signal and cure saturation effects.



Head-Tail viewer

- The Head-Tail HDF5 Viewer is a simple tool for visualising the files produced by the Head-Tail system.



Head-Tail viewer

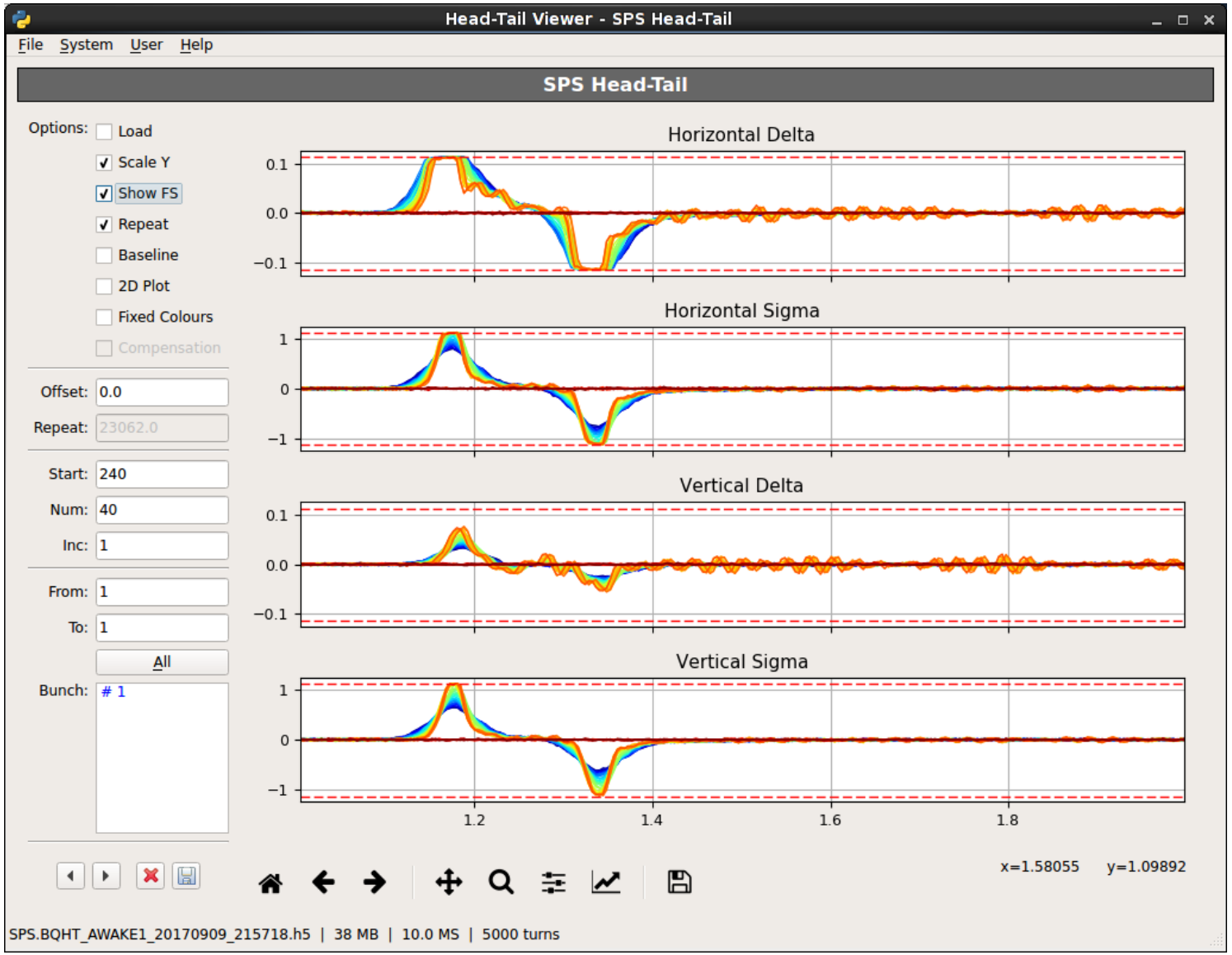
Section	Description	Section	Description
Load	Automatically load new files as they are generated.	Offset	Adjusts the trace offset to centre the bunch in the display, in units of nanoseconds.
Scale Y	Automatically adjust the Y scale to fit the data.	Repeat	Controls the turn length to overlap the traces. Is approximately equal to 1/Frev (i.e. changes with energy) in units of nanoseconds.
Show FS	Show the digitizer full-scale values (for adjusting the gain of the Head-Tail).	Start	First turn to display.
Repeat	Automatically adjust repeat value based on the sum signals.	Num	Total number of turns to display.
Baseline	Process the delta signals to remove effects from the hybrid. The mean of all traces is calculated and subtracted from each trace.	Inc	Number of turns to increment between the displayed traces.
2D Plot	Show the data as a 2D colour plot.	From	First 25ns bunch slot to display (1...3564 or 1...924).
Fixed Colours	Fix the colours of the currently displayed traces, allowing them to be stepped through without the colours changing.	To	Last 25ns bunch slot to display (1...3564 or 1...924).
Compensation	Compensate cable response (not yet implemented)	Bunch	Filled bunch slots are automatically detected based on the sum signal.

Head-Tail control – gain control

- "Unphysical" delta signal: the **centre of the bunch is zero** on the delta signal, like in the following example:

Head-Tail control – gain control

- "Unphysical" delta signal: the **centre of the bunch is zero** on the delta signal, like in the following example:
  - If you **switch off "Baseline" checkbox and turn on the "Show FS" checkbox**, the situation is much clearer:
  - The cause is **saturation of the digitizer** due to too high input signal. The baseline removal algorithm then sees no change between turns, so the saturated region becomes zero.



Head-Tail control – gain control

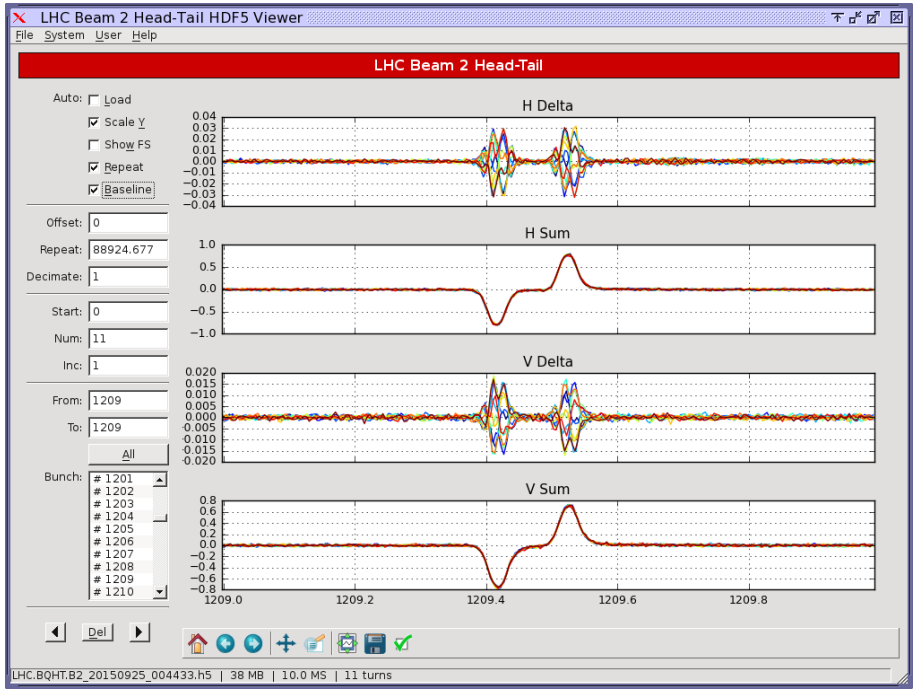
- "Unphysical" delta signal: the **centre of the bunch is zero** on the delta signal, like in the following example:
  - In this example, the Horizontal Delta, Horizontal Sigma and Vertical Sigma channels are all saturated. In this case, **the gain of these channels should be reduced in the control GUI**. After adjusting the gain, the signal looks more realistic

Signpost

We have seen how the **operational interface of the Head-Tail monitor** is composed of two different parts: an Inspector panel and a PyQt5 application. We should know now **how to set up the Head-Tail monitor** for certain set of measurements and how to its the visualization tool.

We will now look into the **offline analysis of the output files of the Head-Tail monitor**. Due to the complexity and the size of the output signals, this is not entirely straightforward – but thanks to readily available BI-libraries also not too difficult.

- Content
  - Background
  - Instrument / Usage
  - Post-processing (HDF5)
  - Examples / Gallery



Head-Tail monitor – output format and post-processing

- Head-Tail monitor usually produces large output data with a rather complex file structure
- For this reason, it is stored in the **hierarchical data format (HDF5)**
- The Head-Tail monitor comes with a ready-to-use post-processing library, found under: **https://gitlab.cern.ch/bqht/headtail**

Post-processing – get files

Headtail

In [2]:
```
# git clone https://gitlab.cern.ch/bi/headtail.git
```

In [3]:
```
from headtail.modules import bqht

# Create a new instance and set the system to SPS
ht = bqht.BQHT(system='SPS', user='LHC50NS')
```

In [ ]:
```
# The `files` parameter gives a list of the data files available for this system, so select the latest one
filename = ht.files[-1]
print(filename)
# /nfs/cs-ccr-bqhtnfs/lhc_data/LHC_BQHT/LHC.BQHT.B1/2016_04_16/LHC.BQHT.B1_20160416_071723.h5

# You can also use find a file by name:
filename = ht.file('LHC.BQHT.B1_20160416_071723.h5')
print(filename)
# /nfs/cs-ccr-bqhtnfs/lhc_data/LHC_BQHT/LHC.BQHT.B1/2016_04_16/LHC.BQHT.B1_20160416_071723.h5
```

In [4]:
```
# In our case, we have the headtail output file stored locally - we readfrom there
path = '/home/kli/cernbox/Data/OP_DATA/SPS/HEADTAIL/2017_10_09_sub'
outputpath = path + '/pngs/'
files = sorted(glob.glob(path+'/*.h5'))
```

Import and initialize BI provided BQHT module (here, for SPS)

Possibility to query last recorded files integrated into module

Alternatively, load files from local storage

Post-processing – prepare data

In [5]:

```python
# Prepare figure for turn-by-turn delta and sum signals
fig, axes = plt.subplots(2, 1, figsize=(11, 7), sharex=True, facecolor='w', constrained_layout=True)
smooth = partial(gaussian_filter1d, sigma=2, mode='nearest', axis=0)
fl = files[0]

axes[0].set_ylabel("$\Delta$ [arb. units]")
axes[1].set_ylabel("$\Sigma$ [arb. units]")
axes[1].set_xlabel("25 ns slot")

# Open the file using a context manager
with ht.open_file(fl) as htf:
    print('\n--> Open file', fl)

    # First we need to locate which bunch slots contain bunches
    htf.locate_bunches()

    # The `bunches` parameter should now contain a list of occupied bunch slots
    print(htf.bunches)
    # [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25,
    # 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 57, 58,
    # 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82,
    # 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 114,
    # 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133,
    # 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152,
    # 153, 154, 155, 156, 157, 158, 159, 160, 161, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180,
    # 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199,
    # 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218]

```

Open headtail monitor file object and discover the many implemented features such as:

- Automatically locating bunches using the sum signals
- Provide bunches list (filled buckets of the SPS / LHC)

Post-processing – prepare data

```
28
29      # And the `populated_turns` gives the range of turns for which the beam is present
30      print(htf.populated_turns)
31      # range(0, 100)
32
33      # Initially the `frev` parameter contains a default value which probably does not cause the
34      # traces to overlap properly
35      print(htf.frev)
36      # 2.3062e-05
37
38      # Running `optimize_overlap()` performs a fitting to find a better value
39      htf.optimise_overlap()
40
41      # Not much change... was already pretty good
42      print(htf.frev)
43      # 2.3062e-05
44
45      # Enable `align` if you want the output of all turns aligned to common X points.
46      # The default (False) returns a separate X axis per turn with an offset to make the traces al
47      htf.align = False
48
49      # Iterate over the planes (here only horizontal) and signals (sigma/delta)
50      for ax, (plane, signal) in zip(axes, htf.planes_signals[:2]):
51          # Get the data for this plane/signal
52          # You can also use the syntax htf.data.horizontal.delta (for example)
53          # but it is less convenient if you have the plane/signal as strings
```

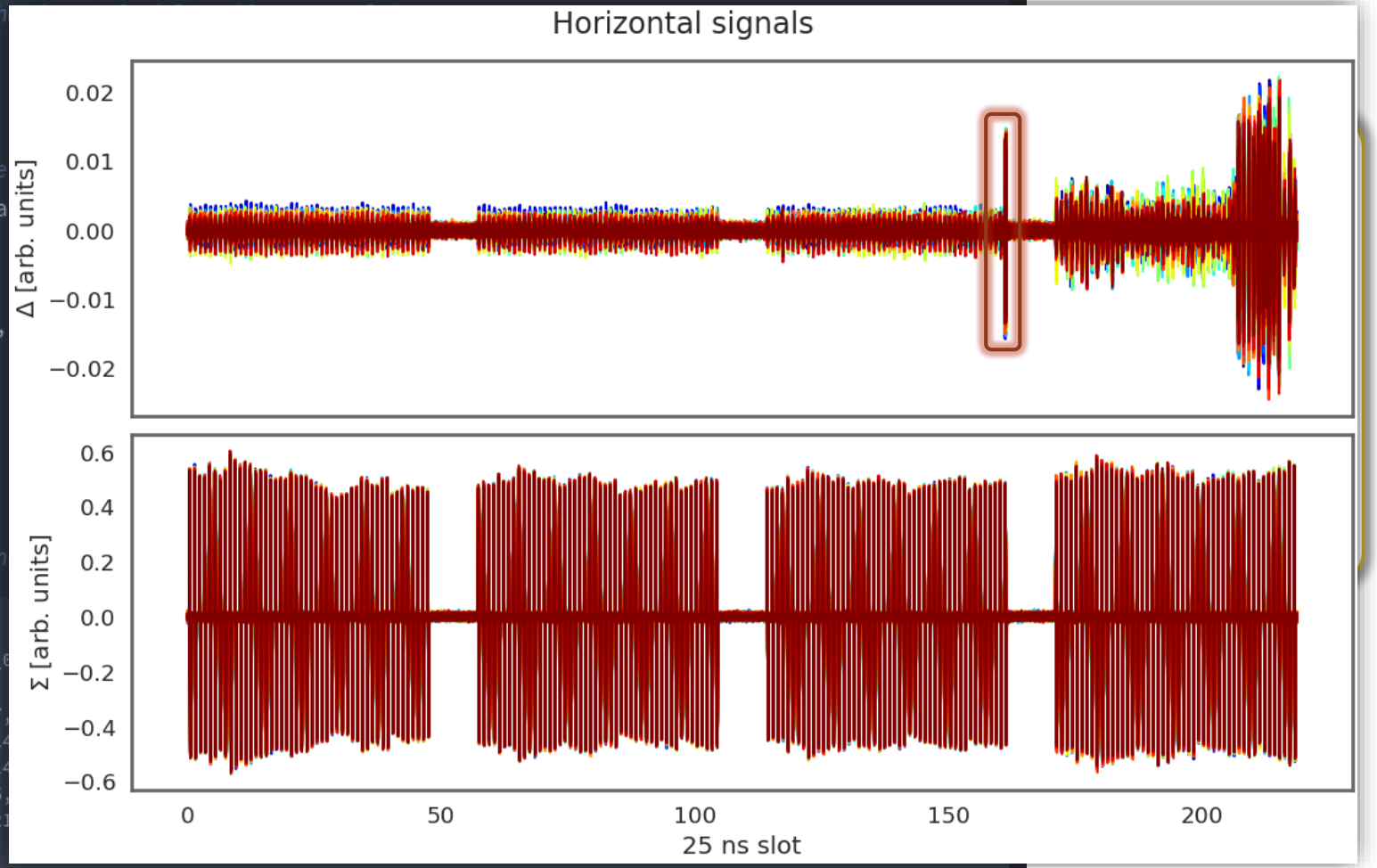
Open headtail monitor file object and discover the many implemented features such as:

- Automatically locating bunches using the sum signals
- Provide bunches list (filled buckets of the SPS / LHC)
- Evaluate turns which contain meaningful signals
- Align turn-by-turn signals using the sum signals

Post-processing – plot data

```
49      # Iterate over the planes (here only horizontal) and signals (sigma/delta)
50      for ax, (plane, signal) in zip(axes, htf.planes_signals[:2]):
51          # Get the data for this plane/signal
52          # You can also use the syntax htf.data.h
53          # but it is less convenient if you have
54          data = htf.data[plane][signal]
55
56          # Well, I actually really do not like Je
57          colors = plt.cm.jet(np.linspace(0, 1, da
58
59          for k in htf.populated_turns:
60              tt, yy = data.get(k, htf.bunches[0],
61              tt = tt / 25e-9
62              yy = smooth(yy)
63              ax.plot(tt, yy, c=colors[k])
64
65      fig .suptitle("Horizontal signals")
66  #       fig.savefig(htf.filename.replace("h5", "pn
```

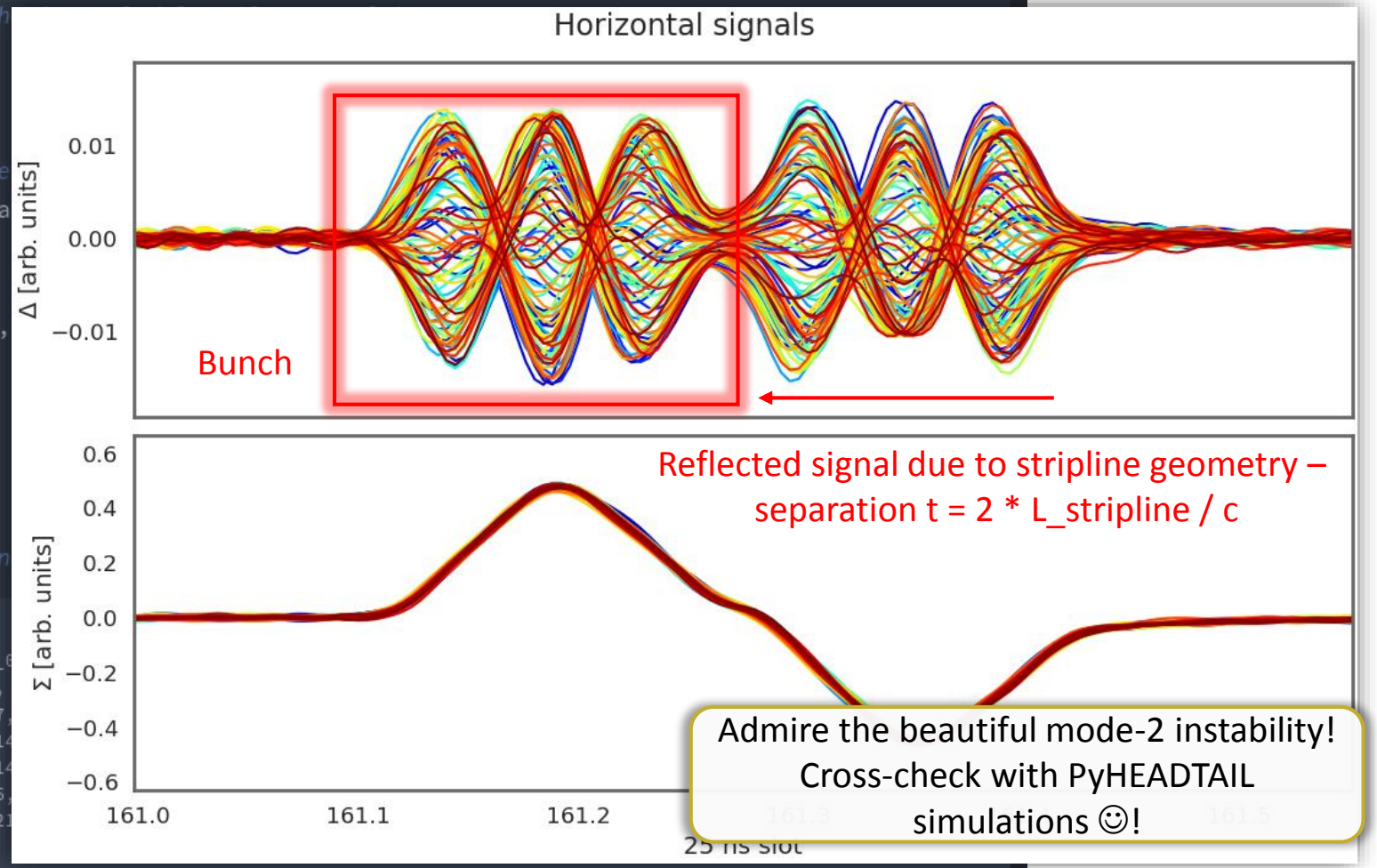
```
--> Open file /home/kli/cernbox/Data/OP_DATA/SPS/HEADTAIL/2017_10_0
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
42, 43, 44, 45, 46, 47, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67,
0, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 11
133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 1
3, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186,
205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 21
range(0, 100)
2.3062e-05
2.3062e-05
```



Post-processing – plot data

```
49      # Iterate over the planes (here only horizontal) and signals (sigma/delta)
50      for ax, (plane, signal) in zip(axes, htf.planes_signals[:2]):
51          # Get the data for this plane/signal
52          # You can also use the syntax htf.data.h
53          # but it is less convenient if you have
54          data = htf.data[plane][signal]
55
56          # Well, I actually really do not like Je
57          colors = plt.cm.jet(np.linspace(0, 1, da
58
59          for k in htf.populated_turns:
60              tt, yy = data.get(k, htf.bunches[0],
61              tt = tt / 25e-9
62              yy = smooth(yy)
63              ax.plot(tt, yy, c=colors[k])
64
65      fig .suptitle("Horizontal signals")
66  #       fig.savefig(htf.filename.replace("h5", "pn
```

```
--> Open file /home/kli/cernbox/Data/OP_DATA/SPS/HEADTAIL/2017_10_0
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
42, 43, 44, 45, 46, 47, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67,
0, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 11
133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 1
3, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186,
205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 21
range(0, 100)
2.3062e-05
2.3062e-05
```

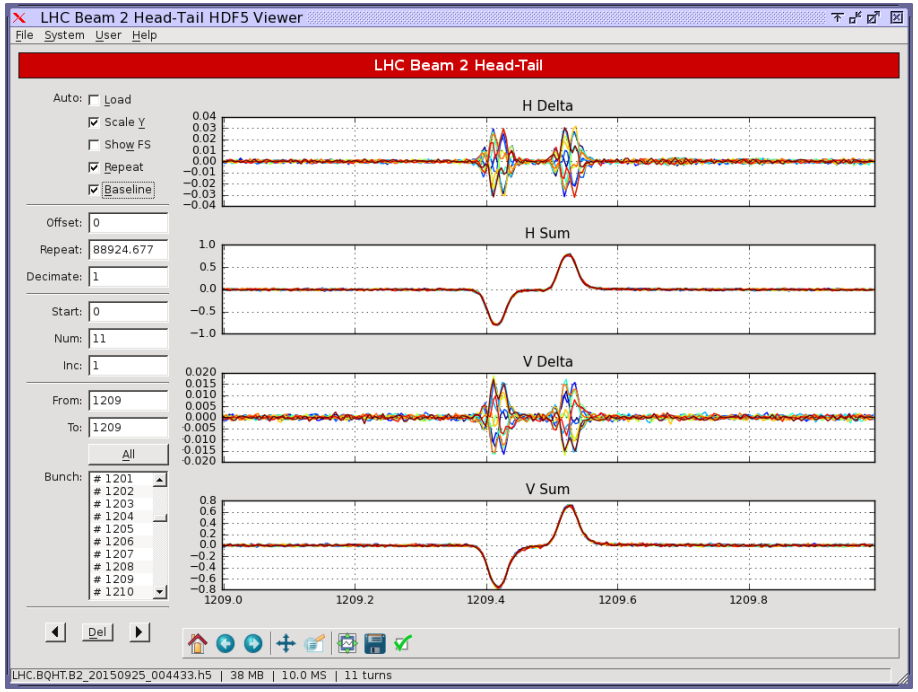


Signpost

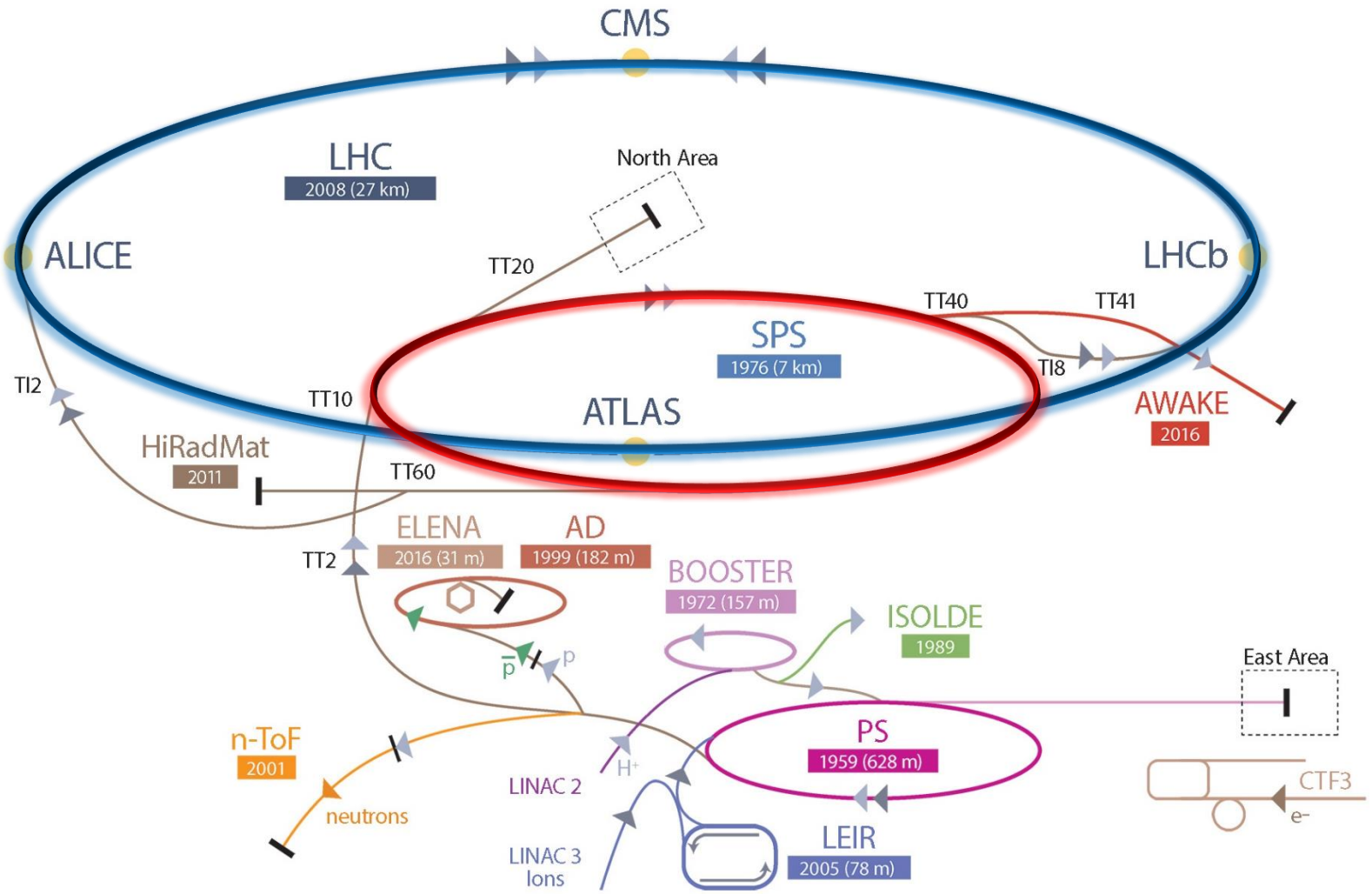
Up to now, we have acquired knowledge on the **basic concepts, ideas and use cases** for the Head-Tail monitor, **the instrument itself and its usage**, as well as **its post-processing library**.

We will spend the rest of the lecture looking into **some real world examples of past uses of the Head-Tail monitor**. We will display it's usefulness as well as the signals obtained which were important parts of the analysis of the underlying encountered problems.

- Content
  - Background
  - Instrument / Usage
  - Post-processing (HDF5)
  - Examples / Gallery

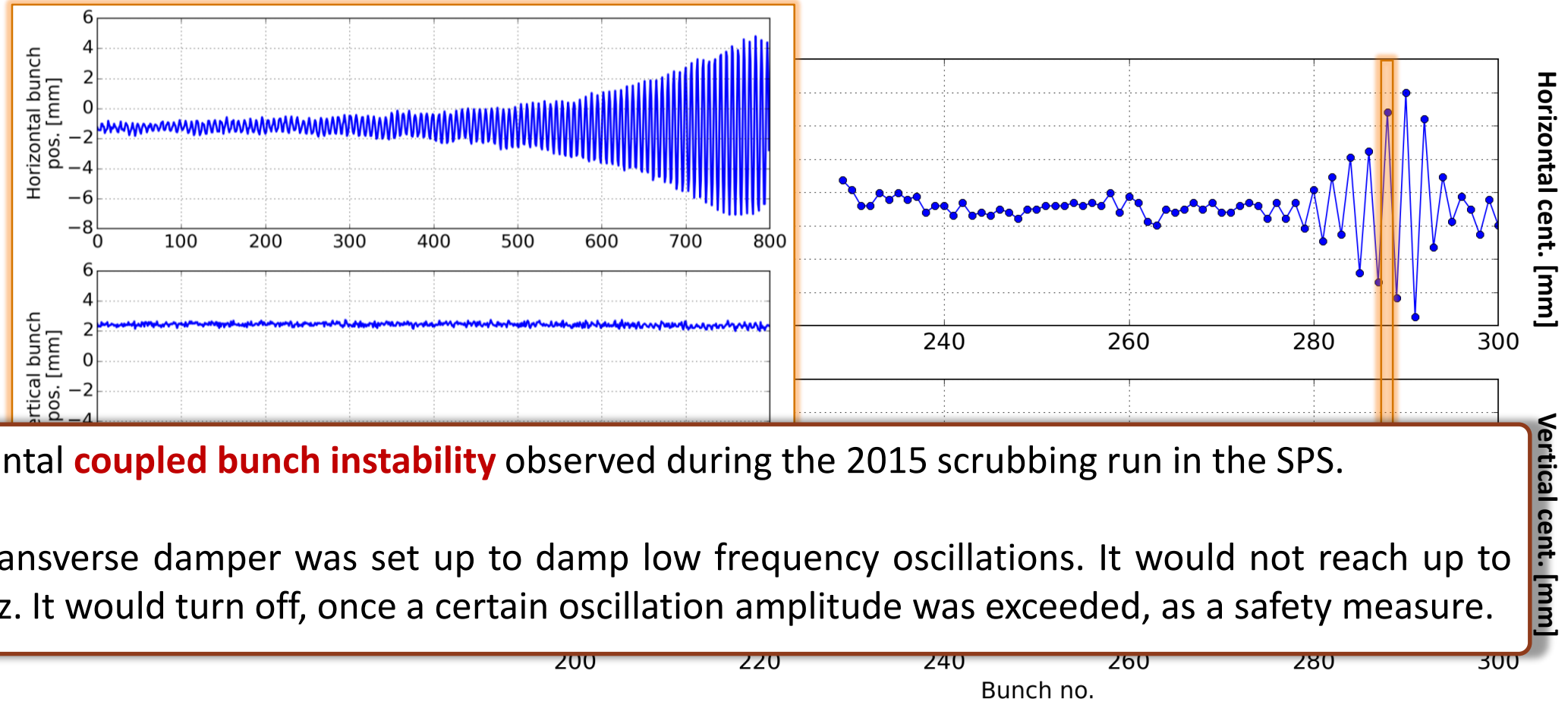


The CERN accelerator complex



Example: coupled bunch instability

- Coupled bunch instabilities vs. single bunch instabilities
  - Depends on: long range vs. short range wakefields
  - Mitigated by: transverse damper systems and Landau damping

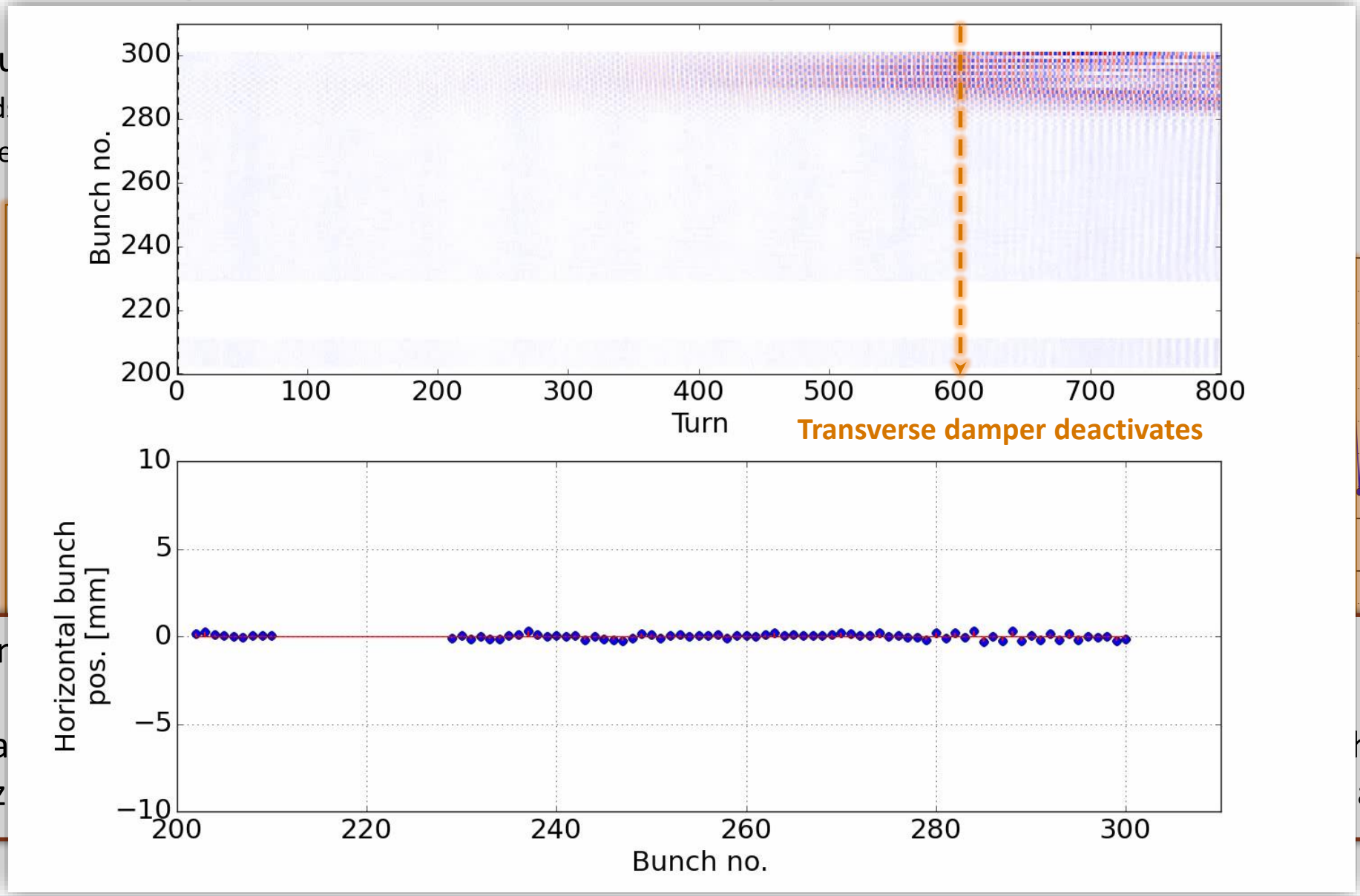


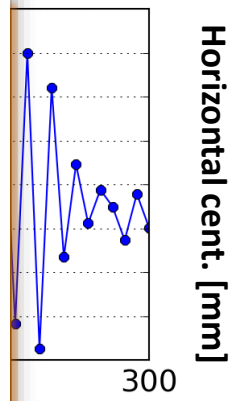
Horizontal **coupled bunch instability** observed during the 2015 scrubbing run in the SPS.

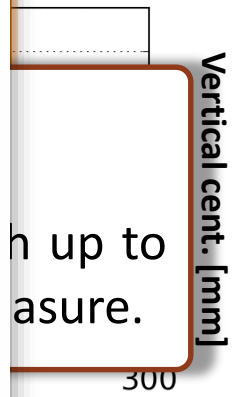
The transverse damper was set up to damp low frequency oscillations. It would not reach up to 20MHz. It would turn off, once a certain oscillation amplitude was exceeded, as a safety measure.

Example: coupled bunch instability

- Coupled bu
  - Depend
  - Mitigate







Horizon

The tra
20MHz

h up to
asure.

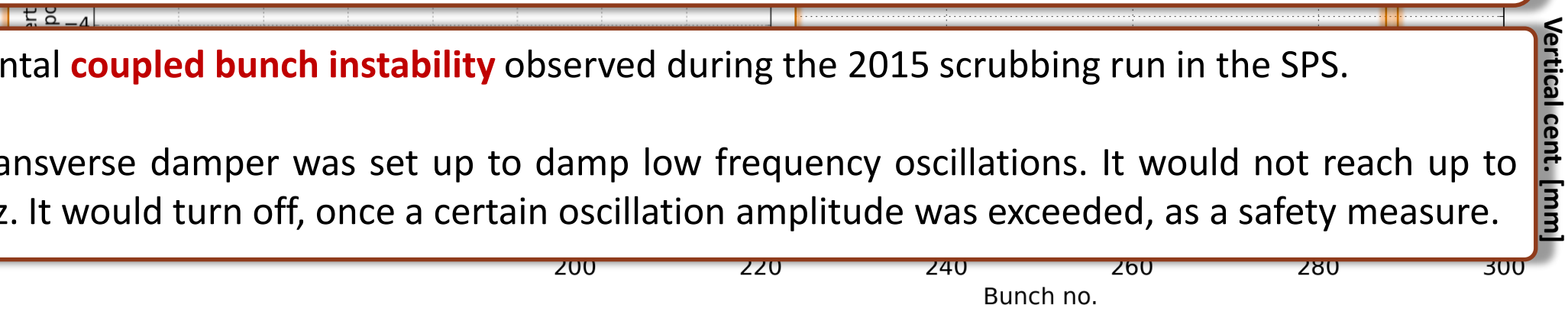
Example: coupled bunch instability

- Coupled bunch instabilities vs. single bunch instabilities
  - Depends on: long range vs. short range wakefields
  - Mitigated by: transverse damper systems and Landau damping

As a consequence, bunches blow up and intensity or even the full beam are lost during the cycle.

During some adjustments in 2017 the **bandwidth of the transverse damper up to 20MHz** could be fully exploited, successfully suppressing this instability.

Instead, **a new type of instability emerged**…



Horizontal **coupled bunch instability** observed during the 2015 scrubbing run in the SPS.

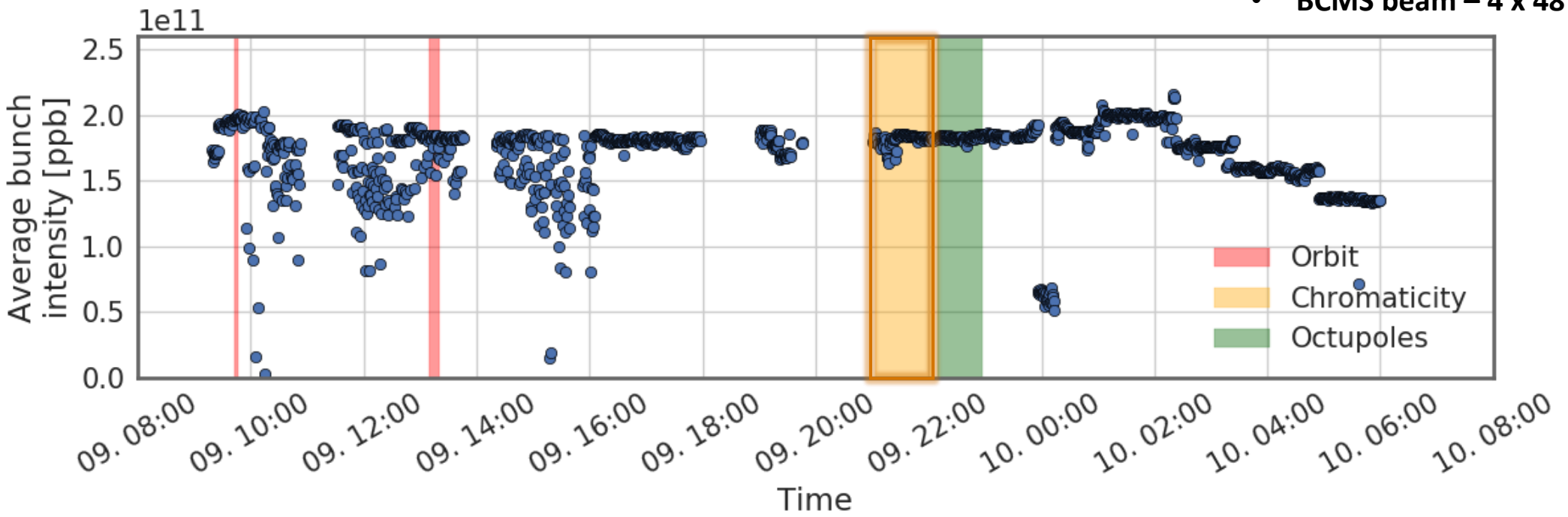
The transverse damper was set up to damp low frequency oscillations. It would not reach up to 20MHz. It would turn off, once a certain oscillation amplitude was exceeded, as a safety measure.

Example: single bunch instabilities

- Pure centroid vs. intra-bunch motion (slow headtail instability)
  - Relation between bunch length and impedance spectrum

- **Investigation beam stability and incoherent losses as a function of chromaticity for high intensity beams.**

- **BCMS beam – 4 x 48 bunches**

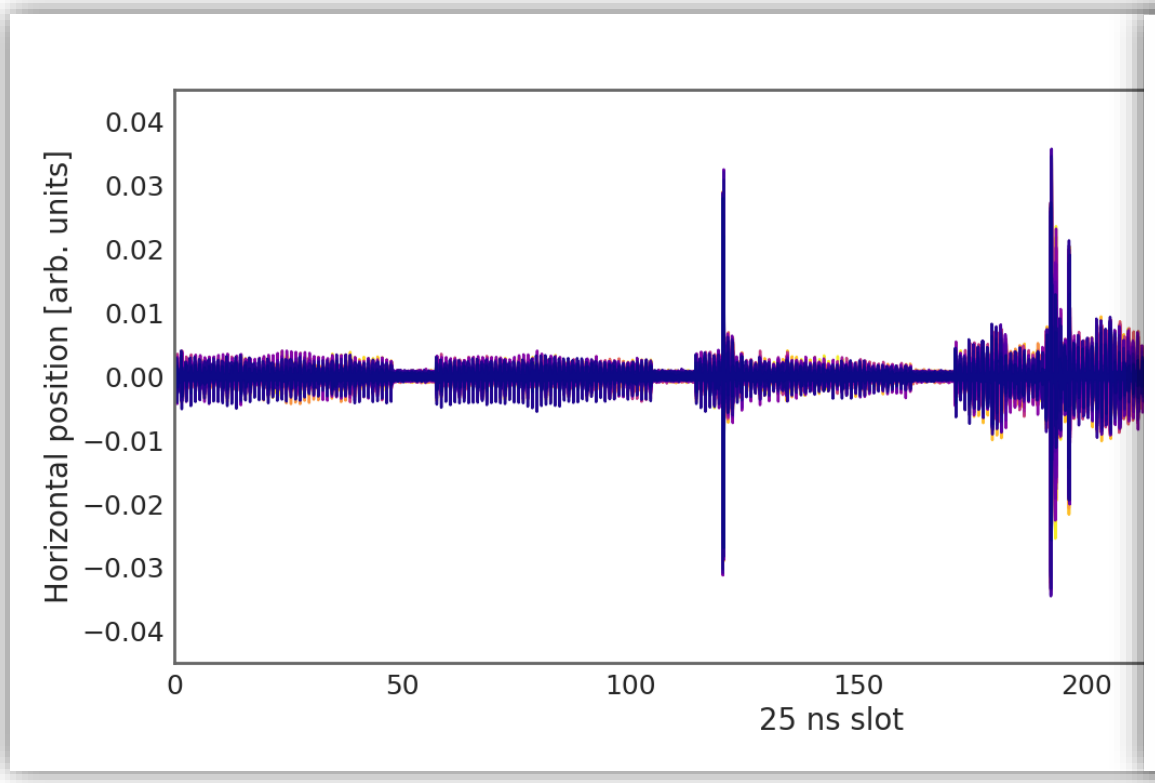


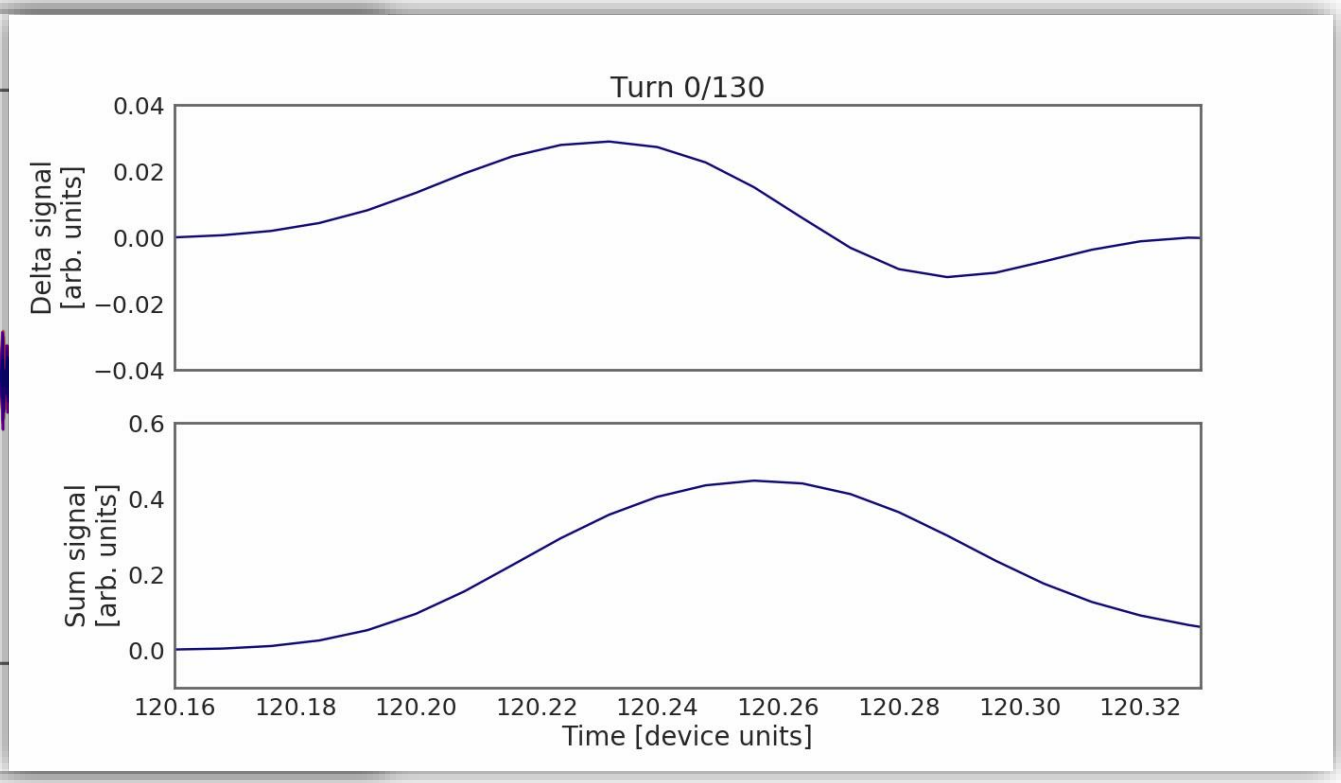
The horizontal coupled bunch instability **at 20 MHz did no longer appear** with the transverse damper now having been fine adjusted resulting in finite gain at these frequencies.

Example: single bunch instabilities

- Pure centroid vs. intra-bunch motion (slow headtail instability)
  - Relation between bunch length and impedance spectrum

- **Investigation beam stability and incoherent losses as a function of**





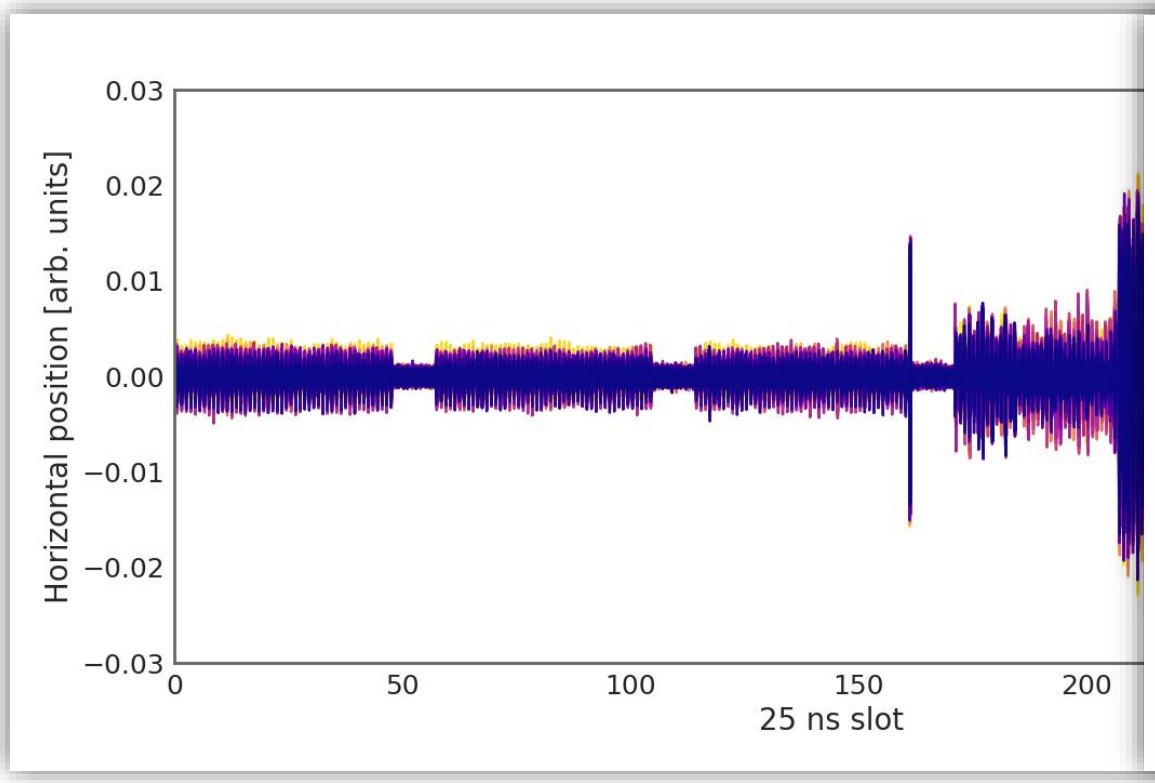
Time

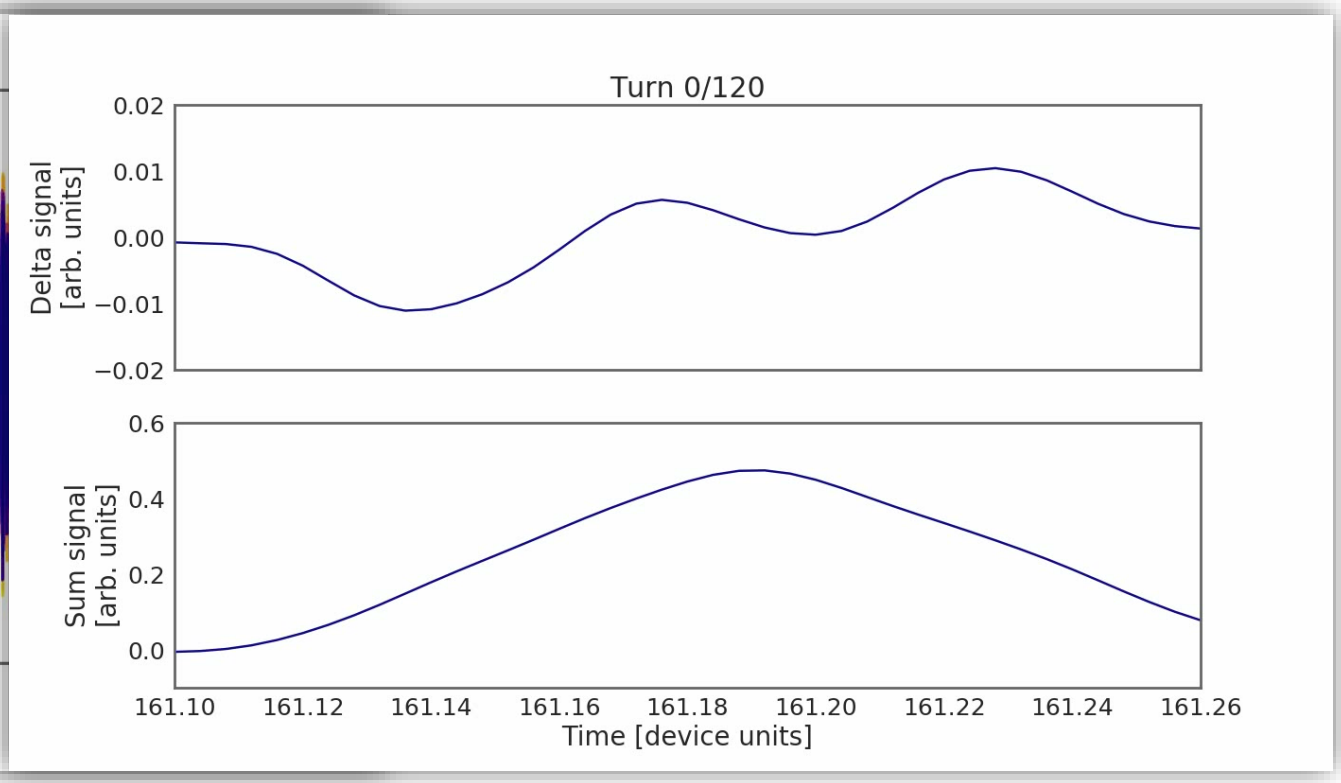
Setting the **chromaticity to 0.2** (normalized units) yielded **instabilities that were not mitigated by the transverse damper**.

Example: single bunch instabilities

- Pure centroid vs. intra-bunch motion (slow headtail instability)
  - Relation between bunch length and impedance spectrum

- **Investigation beam stability and incoherent losses as a function of**



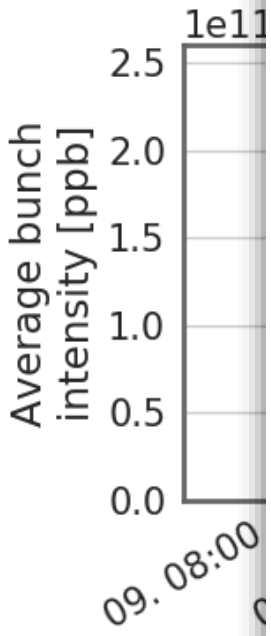


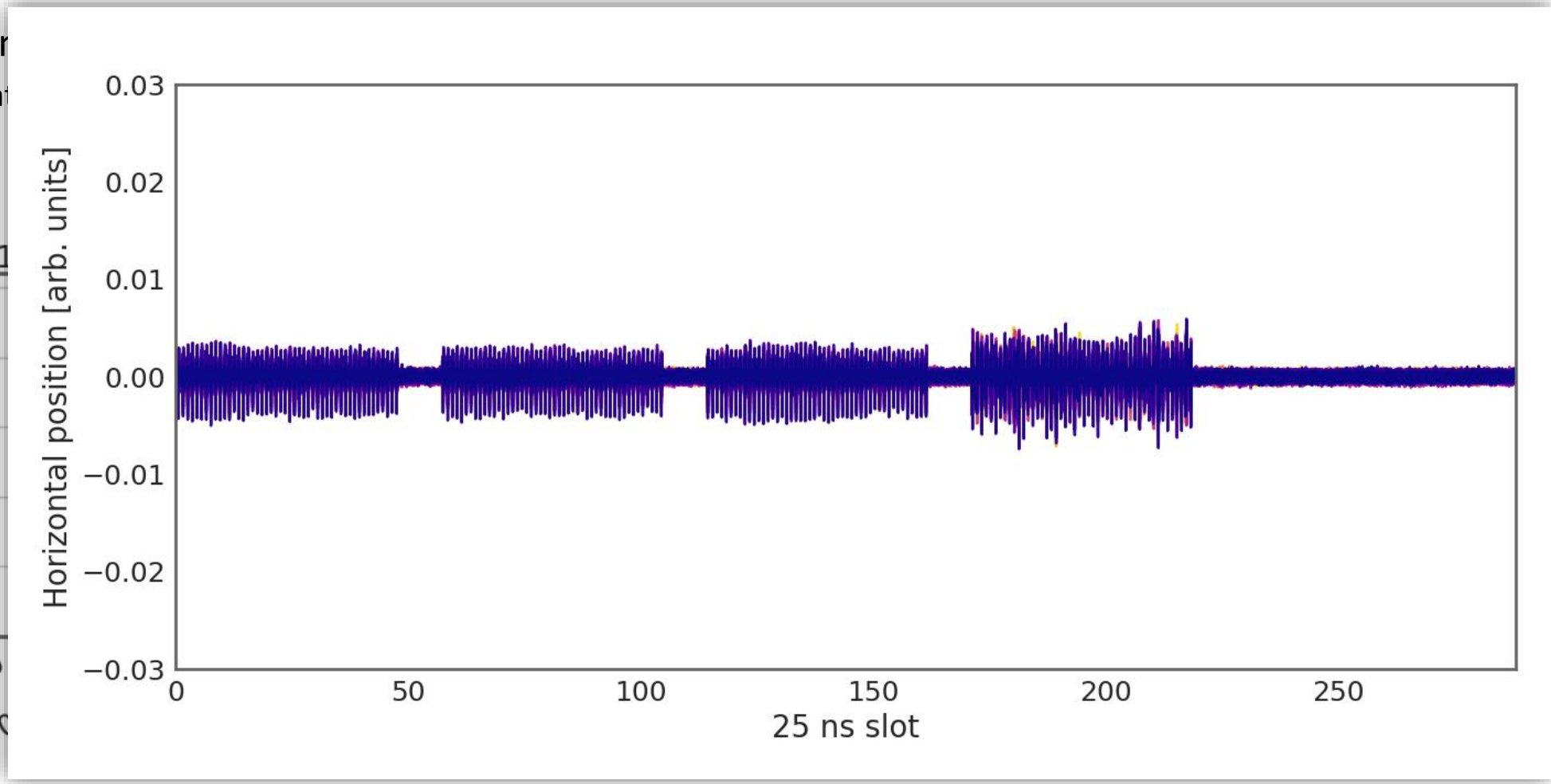
Time

Setting the **chromaticity to 0.4** (normalized units) yielded **instabilities that were not mitigated by the transverse damper**.

Example: single bunch instabilities

- Pure cer
  - Relat





bility and unction of sity beams.

es

Setting the **chromaticity to 0.6** (normalized units) the **instabilities were suppressed**.

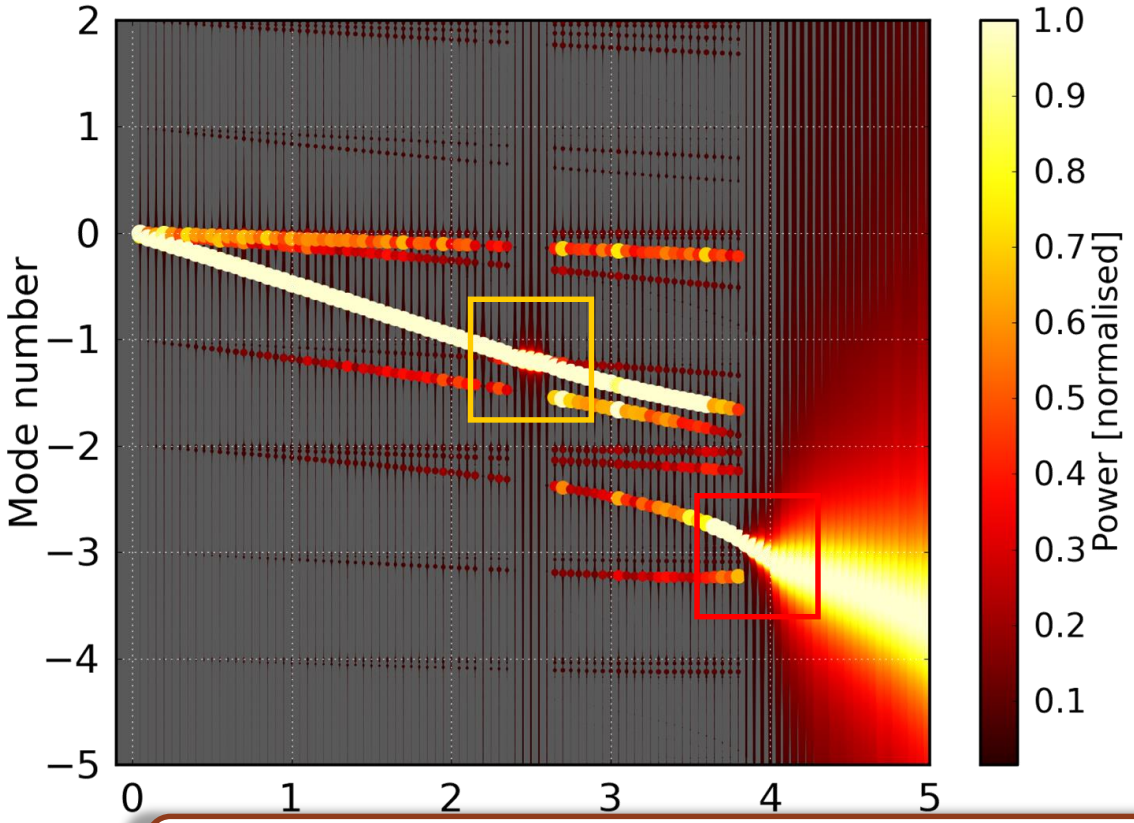
Background – instabilities

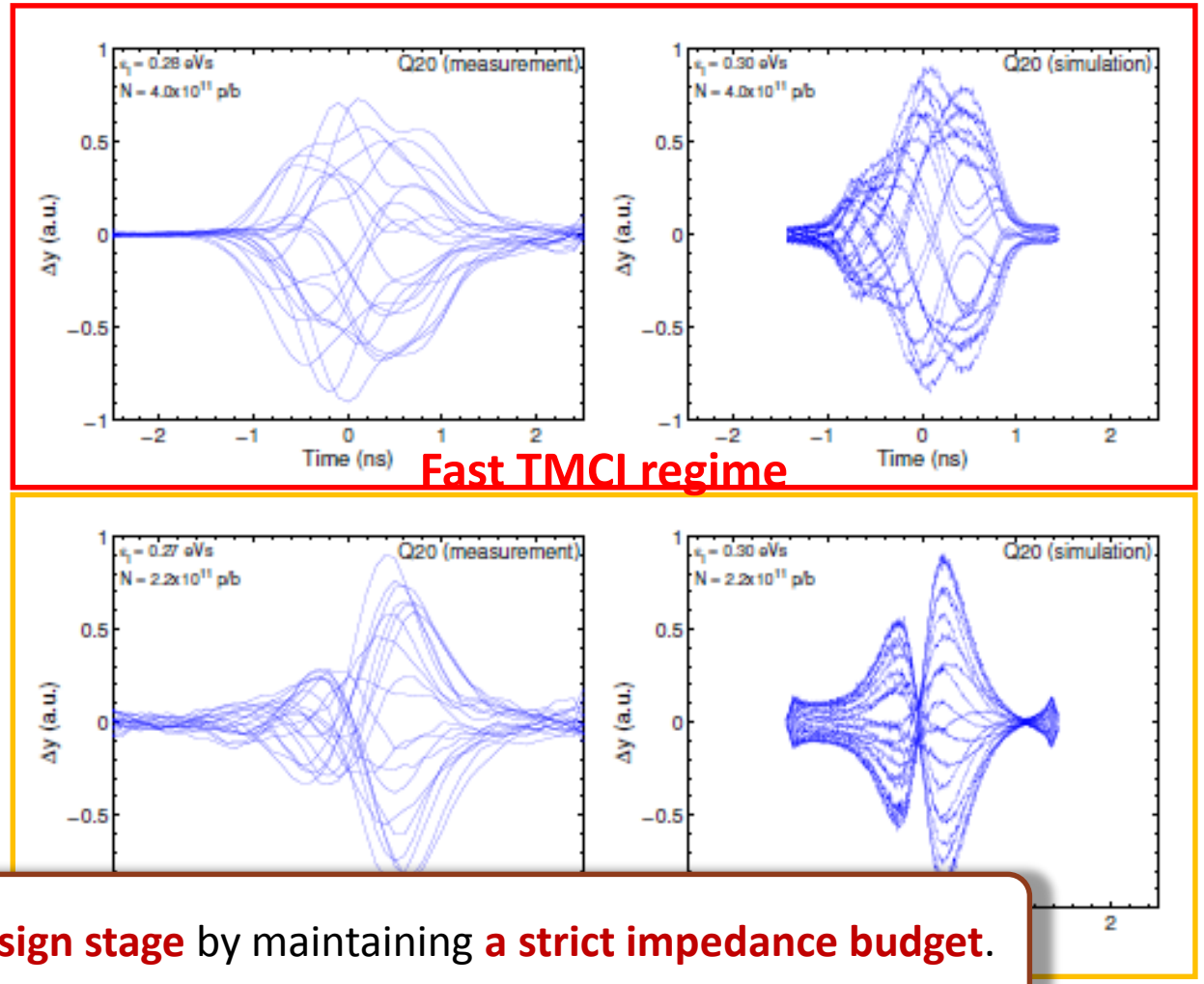
- The Head-Tail monitor has been very important to characterize **the slow Head-Tail instability in the SPS**; this type of instability could potentially become **a limitation for LIU beams** and it is important to complement simulation studies with **actual measurements in order to consolidate the theoretical understanding of the instability mechanisms**. This will help to devise adequate means of mitigation.

- Another example where the Head-Tail monitor proved its usefulness are studies of the **fast headtail instability** also known as the **transverse mode coupling instability (TMCI)**!

Transverse mode coupling instability in the SPS

- TMCI in the SPS (modeled using a broadband resonator impedance at around 1.3GHz)





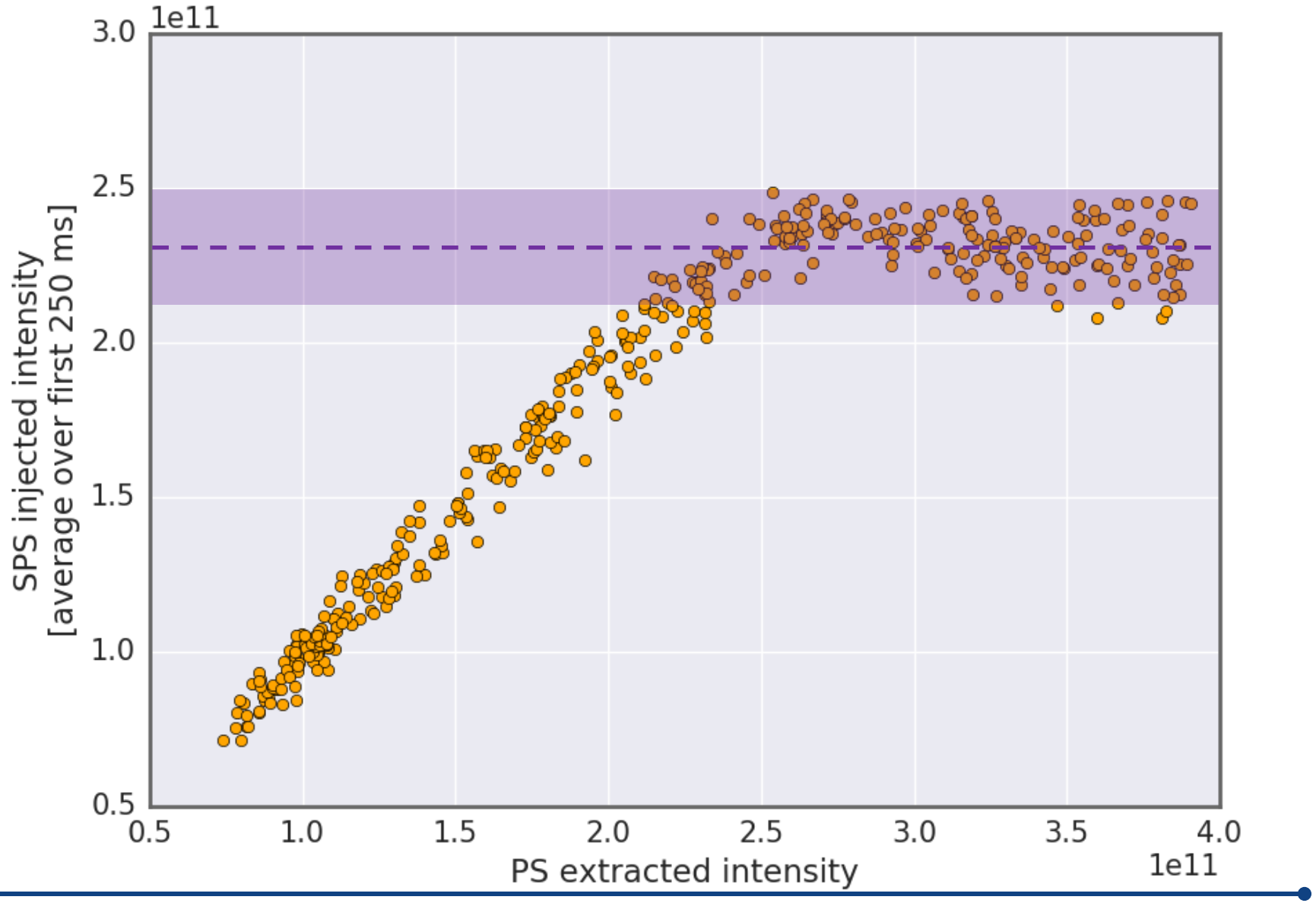
Mitigation of TMCI is usually done **already at the design stage** by maintaining **a strict impedance budget**.

H. Bartosik

An intensity scan in the SPS from 2017

- Intensity scans clearly reveal an **intensity limitation at around 2.4e11 ppb**

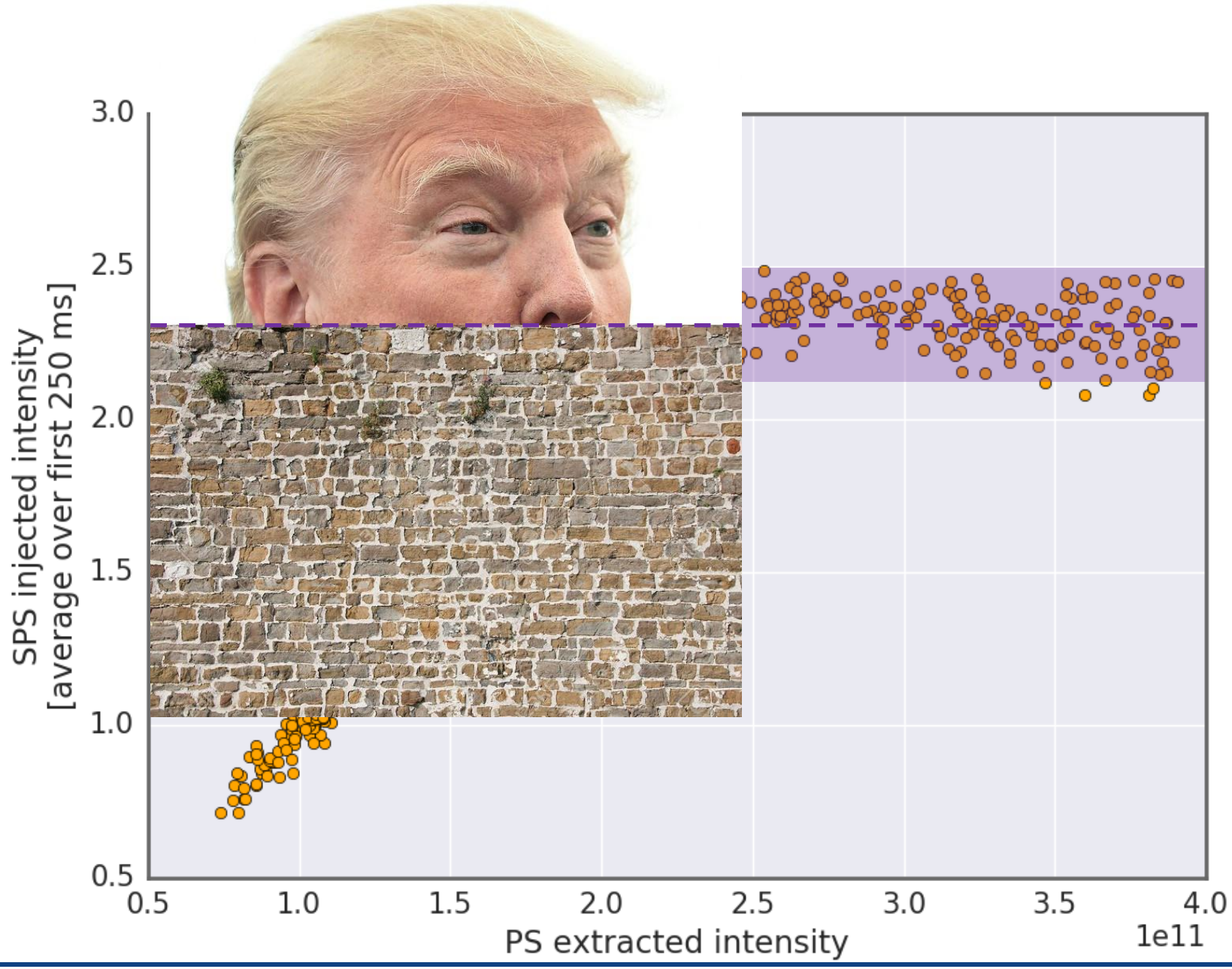
- This limit manifests itself via a strong instability accompanied by **very fast losses**. This is the transverse mode coupling instability (**TMCI**).



TMCI threshold

- Intensity scans clearly reveal an **intensity limitation at around 2.4e11 ppb**

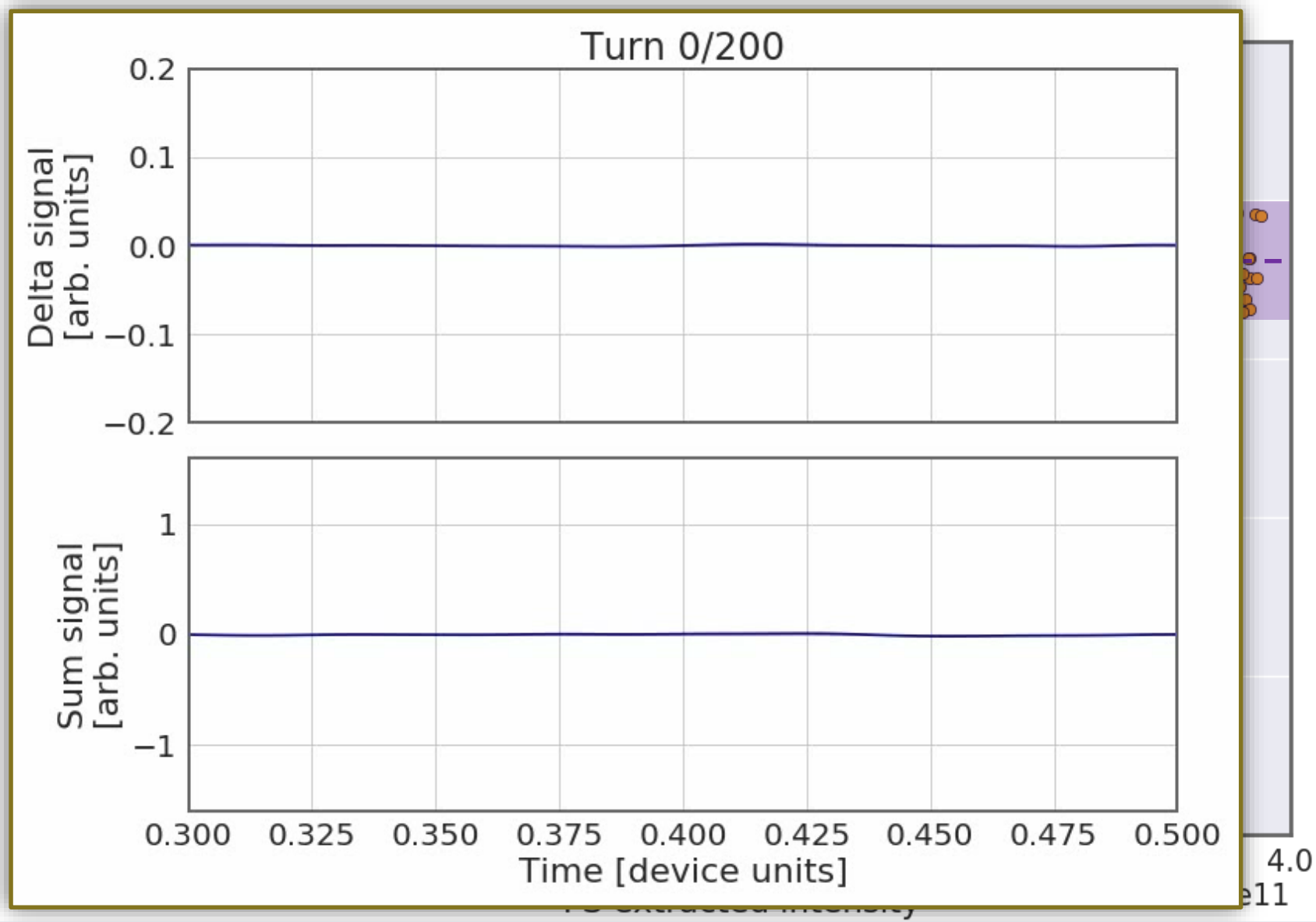
- This limit manifests itself via a strong instability accompanied by **very fast losses**. This is the transverse mode coupling instability (**TMCI**).



An intensity scan in the SPS from 2017

- Intensity scans clearly reveal an **intensity limitation at around 2.4e11 ppb**

- This limit manifests itself via a strong instability accompanied by **very fast losses**. This is the transverse mode coupling instability (**TMCI**).

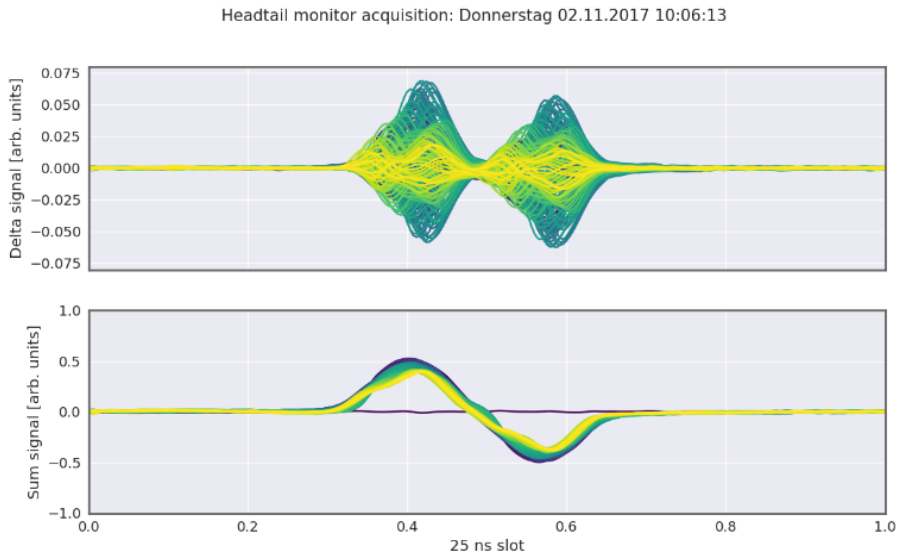


TMCI threshold

- Intensity scans clearly reveal an **inte**
  aroun
- This li
  a stror
  accom
  losses
  mode
  (**TMCI**
- The LH
  will re
  intens
  the fut

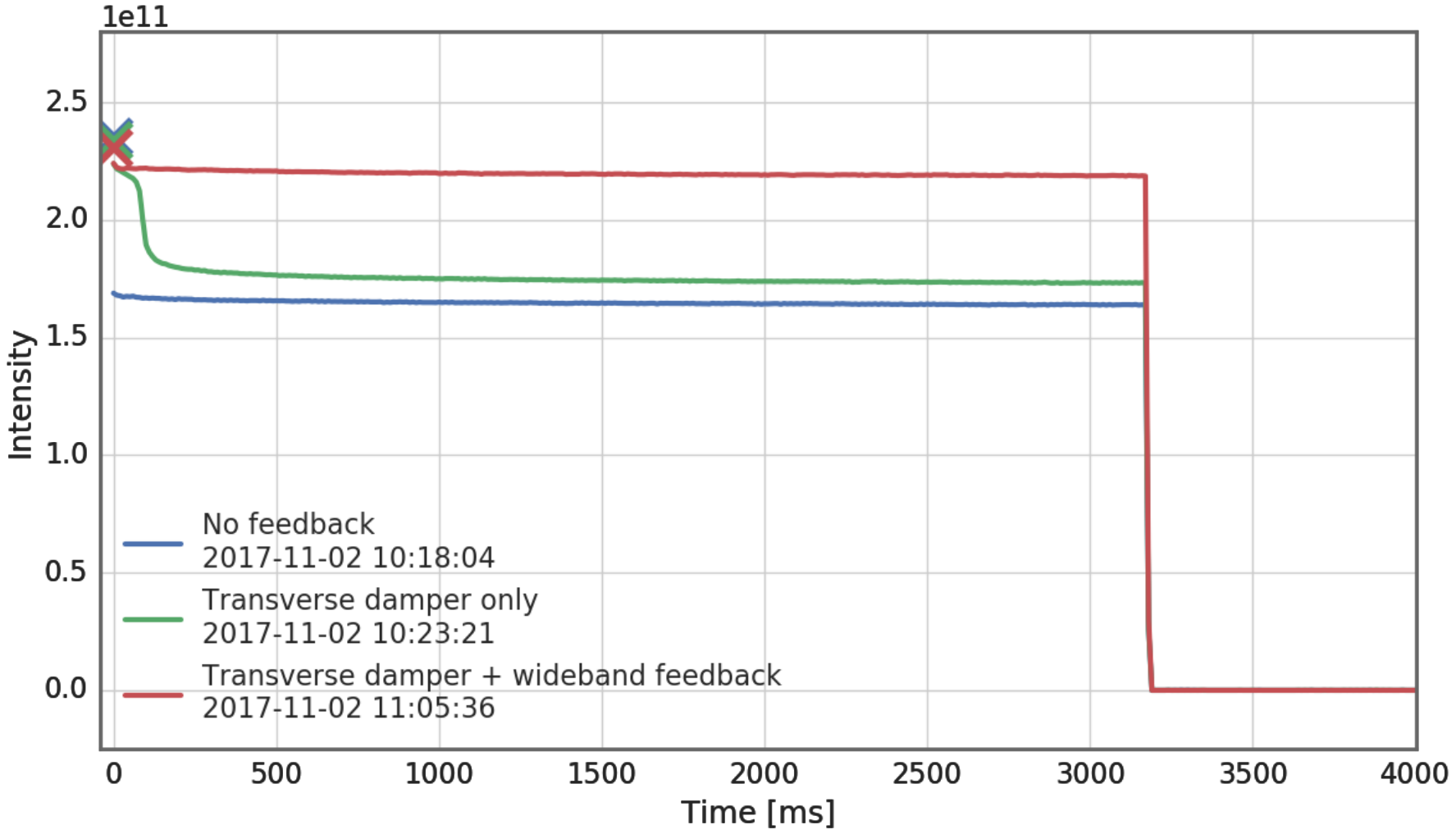
Simulations exit to **estimate the required bandwidth** for being able to damp this type of instabilities using **a high bandwidth feedback system**.

The Head-Tail monitor acquisition of the instability can give **additional information from actual measurements** on this required minimum bandwidth.



Intensity scan

- …
- Finally, the **wideband feedback system** was time aligned, configured and activated by closing the loop over the observed instability.
- The **transverse damper was kept active** to control the large amplitude low frequency motion to prevent saturation of the ADCs which would otherwise render the wideband feedback system ineffective.
- With the **two systems active**, the **losses are significantly reduced** and comparable to what is observed in absence of TMCI.



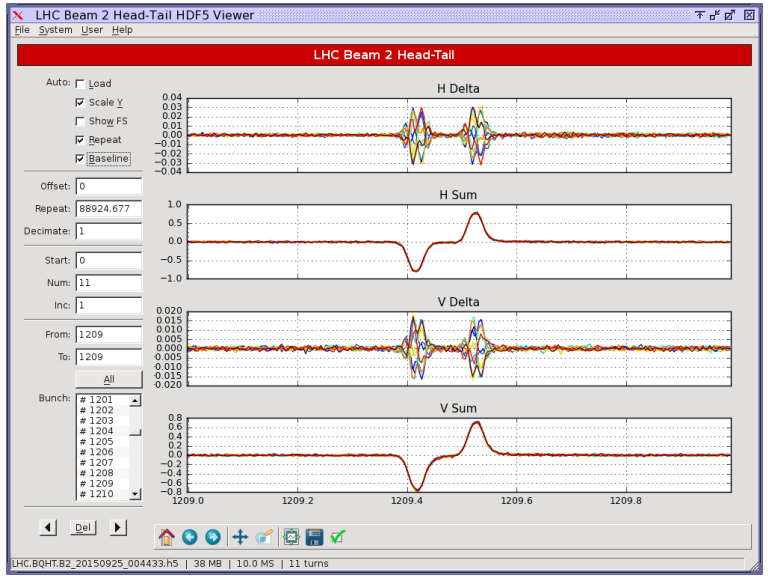
Signpost

What we wanted to convey:

1. The Head-Tail monitor is an important tool to **monitor and understand coherent instabilities**, which is one of the fundamental performance limitations for our machines.
2. The Head-Tail monitor consists of two operational interfaces to allow setting it up and to visualize its output – **we now know how to configure the instrument via the GUIs**.
3. The Head-Tail monitor comes with its own post-processing library; we can follow the example presented to **understand the use the library for our own post-processing**.

I hope you are less confused than I am…!

Questions?



More info: https://wikis.cern.ch/display/BEBI/BQHT

End

www.cern.ch

- SPS CTRVs are installed in FEC CFV-HCA4-BQHT (rack RA5208=HCA442).
- BOBR must be configured to output BST external triggers on HW bytes and set course delay:
  - hardwareByteSelectByte0Block1 = 57
  - hardwareByteSelectByte1Block1 = 58
  - coarseDelayBlock1 = 85

LTIM	CTRV	Ch	Description	Load Event	Delay	Start	Clock	Mode	Remote	OMask
SX.BQHT-TURN	0	1	BQHT TRIGGER TURN	SX.SCY-CT	1	EXT1	EXT2	MULTIPLE	FALSE	0 (2)
SX.BQHT-BUNCH	0	2	BQHT TRIGGER BUNCH	SX.SCY-CT	1	CHAINED	EXT1	MULTIPLE	FALSE	0 (4)
SX.BQHT-HNUM	0	3	BQHT TRIGGER HARMONIC NUMBER	SX.SCY-CT	924	CHAINED_STOP	EXT1	MULTIPLE	FALSE	0 (8)
SX.BQHT-REP	0	4	BQHT TRIGGER REPETITIONS	SX.SCY-CT	1	SELF	CHAINED	MULTIPLE	FALSE	0 (16)
SX.BQHT-PP-ARM	1	1	BQHT TRIGGER ARM FROM PRE-PULSE	SX.SCY-CT	1	NORMAL	EXT1	NORMAL	FALSE	2+4+8+32=46
SX.BQHT-CT-ARM	1	2	BQHT TRIGGER ARM FROM TIMING EVENT	SX.SCY-CT	0	NORMAL	1KHZ	NORMAL	FALSE	0 (4)
SX.BQHT-CT-MUL-DEL	1	3	BQHT MULTIPLE TRIGGER FROM TIMING EVENT DELAY	SX.SCY-CT	1	CHAINED_STOP	1KHZ	MULTIPLE	FALSE	0 (8)
SX.BQHT-CT-MUL-REP	1	4	BQHT MULTIPLE TRIGGER FROM TIMING EVENT REPETITIONS	SX.SCY-CT	1	SELF	CHAINED	MULTIPLE	FALSE	0 (16)
SX.BQHT-INST-ARM	1	5	BQHT TRIGGER ARM FROM INSTABILITY	SX.SCY-CT	1	EXT1	1KHZ	MULTIPLE	FALSE	0 (32)

- Multiple triggering
  - Note that the LTIMs SX.BQHT-CT-MUL-DEL and SX.BQHT-CT-MUL-REP allow generating a series of trigger pulses with a fixed delay after the initial SX.BQHT-CT-ARM event.
  - For example, if SX.BQHT-CT-ARM is at t=0ms, SX.BQHT-CT-MUL-DEL delay=1000 and SX.BQHT-CT-MUL-REP delay=3; the trigger pulses will be generated at t=0ms, t=1000ms, t=2000ms and t=3000ms.

- Connections

Source Module	Source Connector	Signal	Dest. Module	Dest. Connector
BST P0 INTERFACE	BUNCH CLK LVDS	BUNCH-CLOCK	CTRV 0	EXT CLK 1
BST P0 INTERFACE	TURN CLK LVDS	TURN-CLOCK	CTRV 0	EXT CLK 2
BST P0 INTERFACE	HW BYTE HI 0	PRE-PULSE	CTRV 1	EXT CLK 1
BST P0 INTERFACE	HW BYTE LO 7	INSTABILITY TRIGGER	CTRV 1	EXT START 1
CTRV 0	OUT 3	TRIGGER	SCOPE	EXT TRIGGER
CTRV 1	OUT 1	ARM	CTRV 0	EXT START 1