LHC feedbacks
OP training 2016
You are in the CCC and you measure the tune with the probe bunch.
You find that the tune is too high by +0.01.
You decide to correct the tune, but apply only a correction of -0.006.
You send the correction.

In terms of control theory you have made a **closed loop feedback** with **integral control** and a ‘feedback gain’ of 0.6.

The LHC orbit, radial position and tune feedback work on the same principles. In this simple example you find all ingredients of the feedbacks.

So maybe there is no black magic – there shall be light!
The closed loop feedback principle

- Compare measured signal with reference → define an error signal.
- The error signal is processed by the Controller that calculates a correction and applies it to the System.
- The System response is re-measured.
- The cycle starts again.

Closed loop control as opposed to feed-forward converges even in the presence of errors in the Controller (model)!
What is hidden in the LHC FB controller?

1. Defines the error on tune or orbit: measurement-target.

2. Calculates a correction for tune or orbit in terms of trim quadrupoles or orbit corrector kicks.

3. Applies the control gain (algorithm).

4. Converts the gradients / angles to PC currents (if needed), apply rate limits.

5. Sends correction to the PC gateways.

The control part is actually split into two steps:

- Calculation of a correction based on the physics response,
- Application of the control gains to the correction
The correction calculation
– beam physics
The correction of the tunes is based on the same **knobs** that are also used the **standard tune correction**.

- *Trims are send to the RTQF/D circuits (32 circuits in total for the 2 beams).*

### Tune PC configuration for the feedbacks

![Image of Tune PC configuration for the feedbacks](image-url)
The calculation of the correction for tune is based on the same **knobs** that are also used the **standard tune correction**.

For the tune case the matrix is directly expressed in current change (no intermediate strength step).

\[ \Delta \tilde{I}[A] = R_{Q} \Delta \tilde{Q} \]

To good approximation one has:

\[
\begin{pmatrix}
\Delta I_{RQTU}[A] \\
\Delta I_{RQTD}[A]
\end{pmatrix} = \frac{E[GeV]}{450} \begin{pmatrix}
0.0260 & 0.0045 \\
0.0043 & 0.0223
\end{pmatrix} \begin{pmatrix}
\Delta Q_{h} \\
\Delta Q_{v}
\end{pmatrix}
\]

### Full B1 matrix elements @ 450 GeV

<table>
<thead>
<tr>
<th>Element</th>
<th>Ch</th>
<th>Cv</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPMBB UA23.ROTD.A1231/REF</td>
<td>0.0043</td>
<td>0.0223</td>
</tr>
<tr>
<td>RPMBB UA23.ROTD.A1282/REF</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RPMBB UA27.ROTD.A3361/REF</td>
<td>0.0043</td>
<td>0.0223</td>
</tr>
<tr>
<td>RPMBB UA27.ROTD.A3382/REF</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RPMBB UA43.ROTD.A3481/REF</td>
<td>0.0043</td>
<td>0.0223</td>
</tr>
<tr>
<td>RPMBB UA43.ROTD.A3482/REF</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RPMBB UA47.ROTD.A4581/REF</td>
<td>0.0043</td>
<td>0.0223</td>
</tr>
<tr>
<td>RPMBB UA47.ROTD.A4582/REF</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RPMBB UA63.ROTD.A5681/REF</td>
<td>0.0043</td>
<td>0.0223</td>
</tr>
<tr>
<td>RPMBB UA63.ROTD.A5682/REF</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RPMBB UA67.ROTD.A6781/REF</td>
<td>0.0043</td>
<td>0.0223</td>
</tr>
<tr>
<td>RPMBB UA67.ROTD.A6782/REF</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RPMBB UA83.ROTD.A7881/REF</td>
<td>0.0043</td>
<td>0.0223</td>
</tr>
<tr>
<td>RPMBB UA83.ROTD.A7882/REF</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RPMBB UA87.ROTD.A8181/REF</td>
<td>0.0043</td>
<td>0.0223</td>
</tr>
<tr>
<td>RPMBB UA87.ROTD.A8182/REF</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RPMBB UA23.ROTF.A1231/REF</td>
<td>0.0260</td>
<td>0.0045</td>
</tr>
<tr>
<td>RPMBB UA23.ROTF.A1282/REF</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RPMBB UA27.ROTF.A2381/REF</td>
<td>0.0260</td>
<td>0.0044</td>
</tr>
<tr>
<td>RPMBB UA27.ROTF.A2382/REF</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RPMBB UA43.ROTF.A3481/REF</td>
<td>0.0259</td>
<td>0.0044</td>
</tr>
<tr>
<td>RPMBB UA43.ROTF.A3482/REF</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RPMBB UA47.ROTF.A4581/REF</td>
<td>0.0259</td>
<td>0.0044</td>
</tr>
<tr>
<td>RPMBB UA47.ROTF.A4582/REF</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RPMBB UA63.ROTF.A5681/REF</td>
<td>0.0260</td>
<td>0.0044</td>
</tr>
<tr>
<td>RPMBB UA63.ROTF.A5682/REF</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RPMBB UA67.ROTF.A6781/REF</td>
<td>0.0260</td>
<td>0.0045</td>
</tr>
<tr>
<td>RPMBB UA67.ROTF.A6782/REF</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RPMBB UA83.ROTF.A7881/REF</td>
<td>0.0260</td>
<td>0.0044</td>
</tr>
<tr>
<td>RPMBB UA83.ROTF.A7882/REF</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RPMBB UA87.ROTF.A8181/REF</td>
<td>0.0260</td>
<td>0.0044</td>
</tr>
<tr>
<td>RPMBB UA87.ROTF.A8182/REF</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
The orbit is more complex because of the number of sensors and actuators. There is no simple set of orthogonal knobs.

The position change at a BPM $u_i$ can be related to the orbit corrector deflection $\theta_j$ with the response matrix based on the relation presented earlier.

$$u_i = R_{ij} \theta_j$$

$$R_{ij} = \frac{\sqrt{\beta_i \beta_j} \cos(|\mu_i - \mu_j| - \pi Q)}{2 \sin(\pi Q)}$$

In vector / matrix format (N != M, at the LHC N ~ 2M):

$$\vec{u} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{pmatrix} \quad \mathbf{R} = \begin{pmatrix} R_{11} & R_{12} & \ldots & R_{1M} \\ R_{21} & R_{22} & \ldots & R_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ R_{N1} & \ldots & \ldots & R_{NM} \end{pmatrix} \quad \vec{\theta} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_M \end{pmatrix}$$

A change of the correctors by $\Delta \theta$ leads to a position change:

$$\Delta \vec{u} = \mathbf{R} \Delta \vec{\theta}$$

For a correction we measure $\Delta u$ and want to obtain $\Delta \theta$!
Orbit response in YASP

You can inspect the response matrix (by corrector = by column)
To describe the coordinates of a point $P$ in space, one usually chooses a reference system $(e_1,e_2)$ that is orthogonal, i.e. the angle between any pair of axis vectors is $90^\circ$.

In addition the basis vectors are usually normalized, i.e. they length is ‘1’ unit.

In this example, $(e_1,e_2)$ constitutes an orthogonal and normalized base of the 2-D space.

Usually the choice of the reference system is arbitrary, or just given for practical reasons (simplicity…).

A transformation matrix can be used to move from one coordinate system to another one, i.e.:

$$
\begin{pmatrix}
 x'_1 \\
 x'_2
\end{pmatrix} = M
\begin{pmatrix}
 x_1 \\
 x_2
\end{pmatrix}
$$

It is also possible to choose a non-orthogonal reference system – but with the drawback that computations become (much) more difficult!
In the context of beam steering, we have a position and a kick space, both of very high dimension (N and M). The two spaces are coupled by the response matrix \( R \).

The ‘heart’ of the steering problem: orthogonal directions in the corrector space are transformed into non-orthogonal direction in monitor space by \( R \) (and vice-versa…) !!

This is due to the accelerator lattice that couples the responses !!

The mathematical difficulty of steering is due to the fact that the individual corrector responses do not form an ORTHOGONAL basis of the monitor space!
Question:

Is it possible to find a basis of the corrector space (by mixing correctors) such that a transformation by $R$ preserves orthogonality?

Answer:

YES it is possible – this is what SVD (Singular Value Decomposition) does for us !!!

The price to pay: instead of working with single PHYSICAL correctors, one will have to work with mixtures of correctors.
The SVD decomposition builds corrector combinations that are referred to as ‘eigenvectors’ of $R$ (although they are not really in the mathematical sense):

$$\tilde{v}_j = \left( \begin{array}{c} v_{1j} \\ v_{2j} \\ \vdots \\ v_{Mj} \end{array} \right) \quad \text{with} \quad R \tilde{v}_j = w_j \tilde{z}_j = w_j \left( \begin{array}{c} z_{1j} \\ z_{2j} \\ \vdots \\ z_{Nj} \end{array} \right)$$

The $M$ solutions (as many as they are correctors) have the desired property of being orthogonal and normalized:

- $\tilde{v}_j$ for the corrector space (they form a complete base of the space)
- $\tilde{z}_j$ for the monitor space

The ‘eigenvalues’ or ‘weights’ $w_j$ is proportional to the r.m.s. position change obtained by applying the corresponding corrector eigenvector:

$$\text{rms} \left( R \tilde{v}_j \right) = w_j \text{rms} \left( \tilde{z}_j \right) \approx \frac{w_j}{\sqrt{N}}$$

The larger $w_j$, the more efficient the eigenvector!
SVD decomposition examples

Sorted eigenvalues for the SPS and LHC rings. There are as many solutions as correctors!

SPS H plane

Ratio max/min ~ 100
Very regular lattice!

LHC H plane – 2 coupled rings – IR1&5 squeezed

Ratio max/min ~ 10’000
‘Near singular’ solutions in the LHC IRs

‘Singular’ solutions (i.e. very poor ratio response/kick strength).
As the eigenvalues decrease, the associated eigenvectors correspond to increasingly local 'structures'.

LHC OFB usually operates with 350-450 solutions.
Corrections with SVD

1. The position measurement is decomposed uniquely into the monitor eigenvectors. The residual is the uncorrectable part of the measurement.

2. The coefficients $c_j$ are obtained from a very simple and very fast operation (known as ‘scalar’ product).

3. The same $c_j$ are used to compute the correction kicks $\Delta q$, because there is a direct correspondence between the eigenvectors vectors in position and corrector space. The correction may use between 1 and $N$ eigenvectors (the ‘user’ choice).

$$\vec{u} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{pmatrix} = \sum_{i=1}^{N} c_j \vec{Z}_j + \text{residual}$$

$$c_j = \sum_{k=1}^{N} u_k z_{j,k}$$

$$\Delta \vec{q} = -\sum_{i=1}^{K} \frac{c_j}{w_j} \vec{v}_j$$

Note: this operation can also be cast into matrix form.
The response matrix must be build and its SVD decomposition must be performed in advance.

- Takes ~10-20 seconds per plane and optics at the LHC.

The number of eigenvalues is defined for each plane.

For every orbit that is acquired, the feedback removes the reference and applies the SVD inversion (previous slide) to obtain the corrector kicks.

How does one choose the number of eigenvalues?

- More eigenvalues:
  - Better correction (more local structures are corrected),
  - More sensitivity to errors (BPMs).

- In practice:
  - Ramp, squeeze: 390/420 eigenvalues H/V (max ~520).
  - Stable beams: 40 eigenvalues → only correct global distortions, do not interfere with lumi scans, levelling…
The control part
So far we have taken the input data and calculated a correction that would in principle correct 100% of the error.

If we would apply directly this correction, the feedback would become unstable in most situations (not for a simple steady drift!).

To control the loop stability we apply now the control gains in the form of a PI (proportional – integral) digital (discrete) controller:

\[ y_n = y_{n-1} + K_p (e_{n-1} - e_{n-2}) + K_i T_s e_{n-1} \]

- \( n \) denotes the loop iteration,
- \( T_s \) is the loop period,
- \( y_n \) is the corrector setting at loop iteration \( n \),
- \( e_n \) is the output of the tune/orbit correction (previous slides) at iteration \( n \),
- \( K_p \) and \( K_i \) are the proportional/integral gains.
The control part – in words

We are at iteration n:

- We start from the last corrector setting $y_{n-1}$,
- We calculate the last error correction $e_{n-1}$, multiply it by the integral gain $K_i$ and the period $T_s$ → integral controller term,
- We take the difference between the last error correction $e_{n-1}$ and the one before $e_{n-2}$ and multiply by the proportional gain $K_p$ → proportional controller term.
- We add the two terms to $y_{n-1}$ and obtain the new correction setting $y_n$.

$$y_n = y_{n-1} + K_p (e_{n-1} - e_{n-2}) + K_i T_s e_{n-1}$$

$n$ denotes the loop iteration,
$T_s$ is the loop period,
$y_n$ is the corrector setting at loop iteration $n$,
$e_n$ is the output of the tune/orbit correction (previous slides) at iteration $n$,
$K_p$ and $K_i$ are the proportional/integral gains.
Rule of thumb:

\[ \frac{1}{T_s} \geq 25 \times (\text{highest frequency to be corrected}) \]

- In general a factor 100 is often used…
- Delays in the loop (at the LHC delay = 1 period) degrade the performance.
Typical default gain factors:

<table>
<thead>
<tr>
<th>Loop</th>
<th>Kp</th>
<th>Ki</th>
<th>Ts (s)</th>
<th>Ki*Ts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orbit</td>
<td>0.0025</td>
<td>0.1571</td>
<td>0.04</td>
<td>0.0063</td>
</tr>
<tr>
<td>Radial</td>
<td>0.0100</td>
<td>0.6283</td>
<td>0.04</td>
<td>0.0251</td>
</tr>
<tr>
<td>Energy</td>
<td>0.0050</td>
<td>0.3142</td>
<td>0.04</td>
<td>0.0126</td>
</tr>
<tr>
<td>Tune</td>
<td>0.0500</td>
<td>3.1416</td>
<td>0.08</td>
<td>0.2513</td>
</tr>
</tbody>
</table>

For the orbit we send ~ % of the correction 25 x per second...

- The **gain scaling** factors set through the sequencer or YASP multiply both K_p and K_i.

- The information on the FB gains for orbit and radial loops can be found in YASP in the FB control panel →

- There is margin to increase the gains, but then the sensitivity to errors (optics !) and delays is much higher: can drive the FB unstable.
Orbit FB gain & damping

Gain scaling = 4
Damping ~4 s

Gain scaling = 1
Damping ~15 s
Tune FB gain and damping

Standard ramp gain in 2015

Damping ~12 s
The following gain scaling factors were used in 2015, they may be updated in 2016.

**OFB:** $GF = 1 \Rightarrow$ damping in $\sim 15$ s
- Injection $GF = 1$,
- Ramp $GF = 2$,
- Squeeze $GF = 1$,
- SB $GF = 0.2$.

**QFB:** $GF = 1.5 \Rightarrow$ damping in 12 s
- Ramp $GF = 1.5$,
- Squeeze $GF = 0.5$. 
The hardware - actuators
The trim quadrupole circuit:

- The parallel protection resistor $R_{//}$ has an important side effect for feedbacks because it diverts current from the magnet.

**Table:**

<table>
<thead>
<tr>
<th>Circuit Type</th>
<th>Magnet Type</th>
<th>Nmag</th>
<th>$L_{tot}$</th>
<th>$R_{par}$</th>
<th>REE</th>
<th>$R_{crow}$</th>
<th>$U_{th}$</th>
<th>$dt$</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQTD/F</td>
<td>MQT</td>
<td>8</td>
<td>0.248</td>
<td>0.25</td>
<td>0.7</td>
<td>0.05</td>
<td>0.1</td>
<td>0.02</td>
</tr>
</tbody>
</table>

$L = 0.031 \ (H) \quad R_{//} = 0.25 \ (H) \quad \text{per magnet}$

$\tau = L/R = 0.125 \ (s)$
Trim quadrupoles

- The circuit transfer function $G(\omega)$ (→ magnet) is modified by the // resistor, there is a strong roll-off > 1 Hz.
  - As the frequency is increased, more and more energy (current) goes into $R_{//}$ and bypasses the magnet.
  - No point in trying to go to fast. And the QPS does not cope with fast trims either!

\[ |G(\omega)| = \frac{|I_{\text{mag}}|}{|I_{\text{tot}}|} = \frac{|Z(\omega)|}{|L\omega|} = \frac{1}{\sqrt{1 + (L\omega / R_{//})^2}} \]
Magnets for beam steering

- The dipole orbit correctors.
  - ~520 orbit correctors in each plane (⇔ next to focusing quadrupole).
    - There are ≈ twice as many BPMs as correctors.
  - To very good approximation the deflection $\theta$ is proportional to the current $I$.
  - Typical max. ramp rate: $dl/dt \approx 0.01 \times I_{\text{nom}} / s$

\[
\theta [rad] = 0.33 \cdot \frac{BL_{\text{mag}} [Tm]}{p [GeV / c]} \cdot \frac{I [A]}{I_{\text{nom}} [A]}
\]

<table>
<thead>
<tr>
<th>Magnet type</th>
<th>B [T]</th>
<th>$L_{\text{mag}}$ [m]</th>
<th>$BL_{\text{mag}}$ [Tm]</th>
<th>$I_{\text{nom}}$ [A]</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCBH(V) @1.9K</td>
<td>2.93</td>
<td>0.647</td>
<td>1.90</td>
<td>55</td>
</tr>
<tr>
<td>MCBCH(V) @1.9K</td>
<td>3.11</td>
<td>0.904</td>
<td>2.81</td>
<td>100</td>
</tr>
<tr>
<td>MCBCH(V) @4.5K</td>
<td>2.33</td>
<td>0.904</td>
<td>2.11</td>
<td>80</td>
</tr>
<tr>
<td>MCBYH(V) @1.9K</td>
<td>3.00</td>
<td>0.899</td>
<td>2.70</td>
<td>88</td>
</tr>
<tr>
<td>MCBYH(V) @4.5K</td>
<td>2.50</td>
<td>0.899</td>
<td>2.25</td>
<td>72</td>
</tr>
<tr>
<td>MCBXH @1.9K</td>
<td>3.35</td>
<td>0.45</td>
<td>1.51</td>
<td>550</td>
</tr>
<tr>
<td>MCBXV @1.9K</td>
<td>3.26</td>
<td>0.48</td>
<td>1.56</td>
<td>550</td>
</tr>
<tr>
<td>MCBWH(V) @ 290 K</td>
<td>1.1</td>
<td>1.7</td>
<td>1.87</td>
<td>500</td>
</tr>
</tbody>
</table>

$1.9 \text{Tm} \Leftrightarrow \theta = 96 \mu\text{rad} @ 6.5 \text{TeV/c}$
The power converters are controlled by a FGC (Function Generator Controller) with a digital control loop.

- **For orbit and tune correctors the cycle time is 80 ms (12.5 Hz).**

Each converter has a ramp rate \((\frac{dI}{dt})\) and acceleration rate \((\frac{d^2I}{dt^2})\) limit.

- Limits are defined by the PC and/or the Quench Protection System (QPS).

The FGC will ensure in its control loop that the ramp rate and acceleration limits are respected. The current changes will be clamped at the maximum rate in case of requests exceeding the limit.

The FGC has two inputs for current reference: function/setting input and a real time input. At every period it will combine the two sources and send the sum as new reference to the converter.
FGC architecture

Supervision and Control Layer

TCP/IP Network

Power Converter Gateways

Function Generator Controllers

FGC layer
20ms cycle, 2.5Mbps

Field-Bus

WorldFIP bus
The FGC cycles (WorldFip bus) are aligned to the UTC second distributed over the Machine Timing System (MTG) with a period of 12 seconds (to have many flexible sub-periods).

⇒ corrector magnet PCs are aligned to the ‘even’ UTC seconds.

The WorldFIP cycle is 20 ms.
The FGCs enforce a rate limit (\(\frac{dI}{dt}\) limit) which leads to signal deformation and effectively to gain reduction and phase errors of the corrections if you exceed the limits!

**Always below rate limit – perfect tracking**

**Above rate limit – clamping**

Blue = request, Red = actual current
Software architecture and operation
The servers run on machines **cs-ccr-ofc1** and **cs-ccr-ofsu1**

**BFSU FESA class:**
**BFBLHCSettings**

- 2 parts in Orbit FB:
  - **BFSU** = LHC FB Service unit (old OFSU)
  - **OFC Orbit feedback Controller**
The role of the BFSU is to protect the OFC from outside clients and to offload it from service work (like SVD decompositions…).

- **All settings of OFC pass through the BFSU – ‘proxy’**.

For the orbit the OFC/BFSU play the role of data concentrators, they are the source of the orbit data (@ 1 Hz).

- **All trajectory data is provided from JAVA concentrators, the same applies to bunch-by-bunch orbits**.

For the tune, the spectra come from the BBQ FESA classes, the BFSU only publishes the tune & coupling values that are used by the QFB.
Both tune and orbit FBs are embedded inside the OFC.

The OFB operates at 25 Hz, but the PC only swallow corrections at 12.5 Hz – in practice ½ of the OFB corrections are not used.

- We had planned to move the OFC to 12.5 Hz in 2015, but in the end we did not dare to change…

The QFB operates at the frequency of the BBQ data rates. In 2015 we had data @ 4 Hz (with 4000 turns).

- In 2016 we will try to move to 8 Hz (with 2000 turns) → can increase the gain (start of ramp…).
- If someone changes the BBQ frequency, it effectively affects the QFB performance!
The reference orbits for the cycle are associated to the BP type, an application (restricted use) is available to assign reference orbits to matched points.

The reference orbits corresponding to a BP (ramp, squeeze) are loaded by a sequencer task.
The reference orbits are generally constructed as (exception for stable beams reference $\leftrightarrow$ measured orbit):

**Base orbit + overlays** (xing, separation, ULO etc bump).

The base orbit is the same for all references of a given hypercycle (in fact the same for all physics hypercycles).

- **There will be a different base orbit for ATS and for the ballistic optics.**

A new version is in preparation:

- The standard bumps will be imported from the LSA BP settings,
- The reference orbits will be stored as functions by BPMs (standard BP).
Example of BPM functions in a squeeze (here IP1 separation constant at 1 mm on the squeeze).

- There are only values for the matched points.
To visualise 2000 functions (!) the settings can be imported into YASP as a collection of reference orbits and then displayed.
Once imported as orbits, the reference can be used in various GUI components (evolution in 1D/2D, TCT beam position, interpolation to any element...).
The tune reference functions are not so clean, they are still based on a collection of linear tune changes…. An improvement is planned (later in 2016 or 2017).

- **BP time functions for reference, gains etc (like we have them for PCs) will be introduced in the OFB.**

This is why we try to maintain the tune change in well localized points of the cycle (and avoid complicated change schemes).

A complication arises from the fact that for the Q peak finder one also needs the functions for the tune peak search → BBQ FESA class.

- **The BBQ FECs are performing the peak searches, the OFC only receives and processes the results.**
- **Remember than in 2015 we improved a lot the tune quality by placing bunches into the ADT witness (lower) gain region + better FFT filter settings.**
For the QFB no changes has to be applied when an optics is changed. The response of the RQTF/D is \(~\)independent of the optics.

- But the matrix does change for ATS \(\rightarrow\) MDs !!

For the orbit FB the response of the IR CODs ad BPMs (1,2,5,8) changes, sometimes by a lot (for example the Q4 CODs).

- Ideally one should follow step by step the optics in the squeeze.
- In practice we have never commissioned the automatic update of the optics (code exists!).
- In practice we use one optics for the ramp (note this year with squeeze and optics change) and for the squeeze. We will probably use:
  - 5m \(\beta^*\) optics for the ramp & squeeze,
  - 60-100 cm \(\beta^*\) optics for the squeeze.
The RT trims are regularly fed forward into the functions (Q and orbit).

This ensures that the FB has less work (→ better stability, less gain needed) and that we can survive short timeout of the FBs.

And to be able to operate without QFB if there is no tune….

After FF the typical RT trims:

- Rms orbit trims are < 1-2 μrad, and max trims < 5-6 μrad.
- Tune trims < 0.002 (exception for start of ramp).

If orbit trims reach or exceed 8-10 μrad (in std operation) it may be a sign that there are bad BPMs….
Example of RT trims during commissioning (here 90m), before feed-forward.

Max RT trims ~15 μrad

Tune mismatch (‘bump’) between matched points + overall drift
What happens when a COD trips and the OFB is on?

- The SVD inversion that is used inside the OFB is trying to use the failing COD (obviously that is where the error is!) → **the correction will not work**.

- The only way to recover the situation is to **re-compute the SVD WITHOUT the failing COD**: then the correction will work well...

- If the SUE induced 60A failures increase too much this year, it will be **difficult to rescue the beam through the OFB**: too many changes would have to be implemented.

- It may however be possible to implement a ‘simple’ COD failure rescue loop in YASP (for stable beams!):
  - **Stops the OFB**.
  - **Uses a trick to recompute the SVD in 1-2 seconds**.
  - **Takes control to correct the orbit as frequently as possible for 1-2 minutes**.
  - **Shift crew would have to reconfigure the OFB once the event is over to resume smooth stable beams**.
  - **I will test this at injection by tripping some CODs early on…**
The BPM crate temperature will be available as fixed display (inside the FB FD or in stand-alone).

Residual temp oscillations cause some of those (small) orbit structures