LOCO for Booster

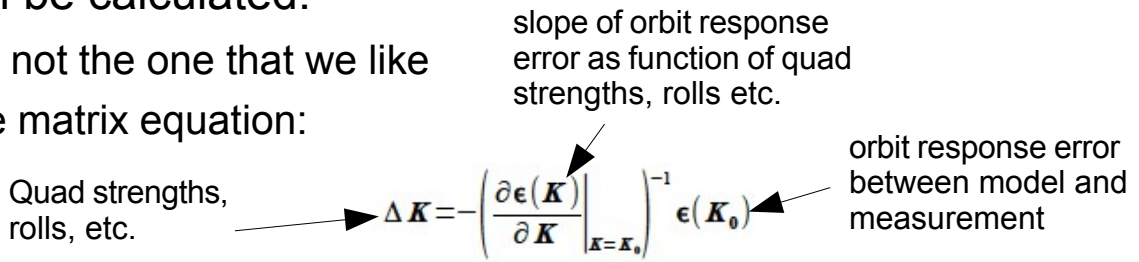
C.Y. Tan
12 Feb 2014

Background

- M. McAteer and A. Petrenko started this whole business (2012?)
  - First correction in 2012.
  - Programs are a collection of ACL scripts, TCL scripts, Mathematica programs, Octave programs, and elegant lattice files.
- I started working on this around March of 2013.
  - Clean up the code. Goal is to move nearly everything to C++.
  - Move to MADX rather than elegant

What is LOCO?

- LOCO (Linear optics from closed orbit) is used to measure the optics using data collected from the all BPMs when a 1 bump is introduced.
- Every dipole corrector is used sequentially and every BPM is used to measure the beam position.
  - The slope dx/dkick is called the orbit response of the beam. IMO, this is the brilliant insight!
- From the orbit response, the Twiss parameters + (other params like quad strengths, bpm calibrations, rolls) can be calculated.
  - The lattice is probably not the one that we like
  - solution of non-square matrix equation:

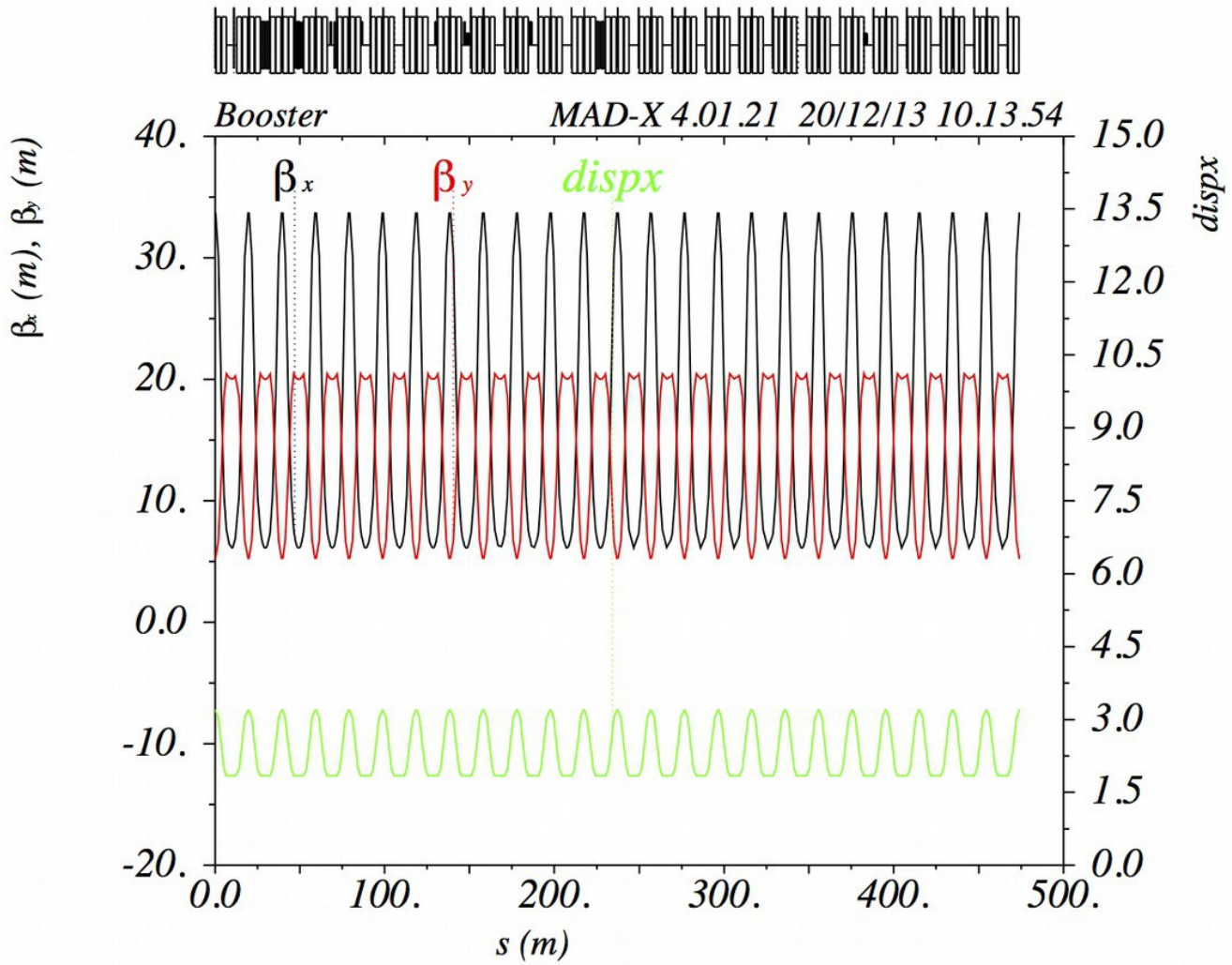


- Corrections are calculated from the measurement and loaded into the corrector quads QL and QS.
- LOCO is used to measure the lattice again and hopefully the optics get closer to the ideal.

Latest LOCO in C++

- After converting from scripts to C++
  - Speed up is substantial (on a Mac)
    - 176 s per break point to 107 s per break point, i.e. 39% faster.
    - For 32 breakpoints: 94 minutes (~1.5 hours) to 57 minutes.
    - If you're running LOCO scripts on Heimdahl, well, 8 hours or more!
  - However, LOCO is naturally parallelizable, so in principle, results for should be obtainable in < 2 minutes if many CPUs are used
    - Next step with OpenMPI with 30 nodes on the control system.
  - Everything is downloadable at redmine.fnal.gov
    - Wiki
    - Git repository of source code
- Write up of the theory, implementation and results is in progress.
  - Preliminary versions of *Optics corrections in Booster* in pdf format is available from me.
  - Should be available soon once the first pass of LOCO has been downloaded into Booster and made operational.

Ideal Lattice (model)

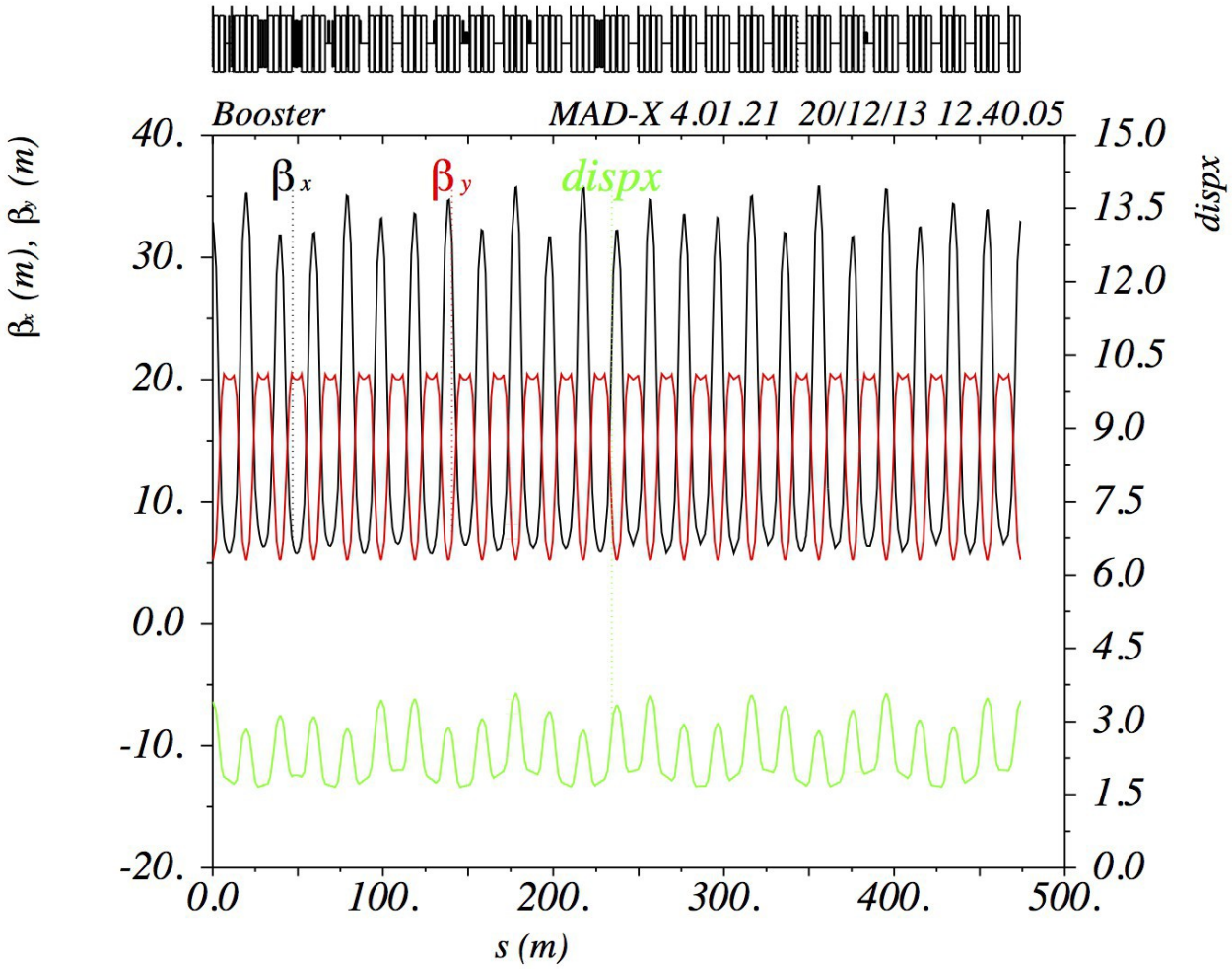


Ideal lattice is when QL and QS have zero current.

However, in real machine QL and QS are not zero.

Pseudoquads(QLerr, QSerr) introduced that sits on top of every QL and QS and LOCO uses these quads to find the errors in the lattice. And in the perfect Universe (QL-QLerr) = 0 and (QS-Qserr) = 0

Dogleg introduces distortions (model)



In principle, putting in the -QLerr and -QSerr found by LOCO into QL and QS, we should find the ideal lattice.

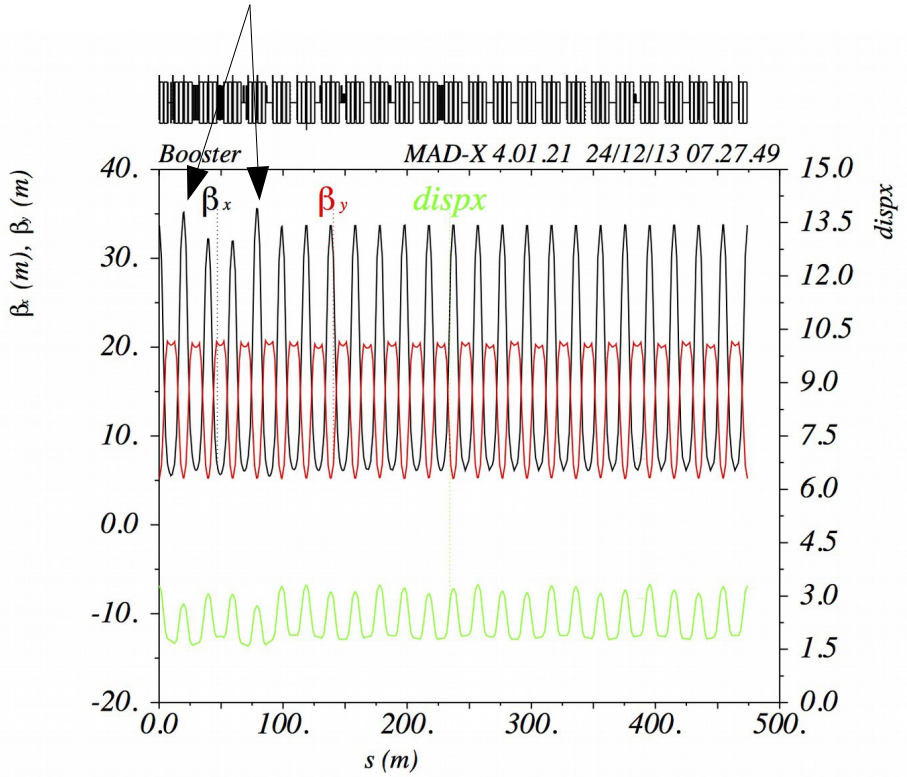
But we will over-correct because the doglegs introduce substantial optics distortion in the horz plane and the **QLerr and QSerr are fitted to this distorted model lattice**.

Therefore, once we get QSerr and QLerr, we fix this distortion by hand.

Note; Effect diminishes as the energy of the beam increases.

Correct with small subset of horz quads (model)

Not perfect but good enough



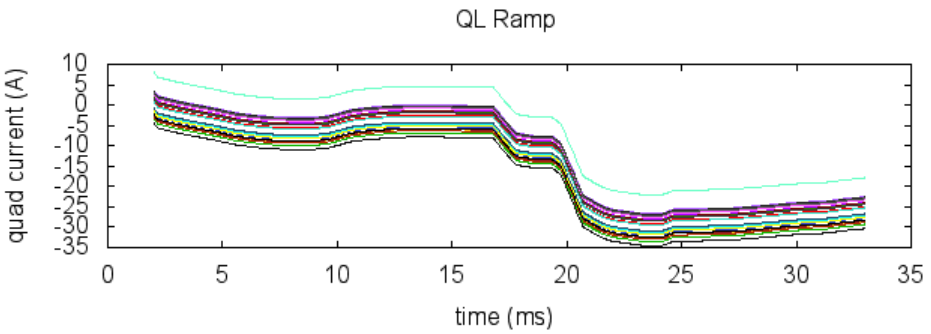
Lower the strength of these quads as the momentum increases using inverse proportionality.

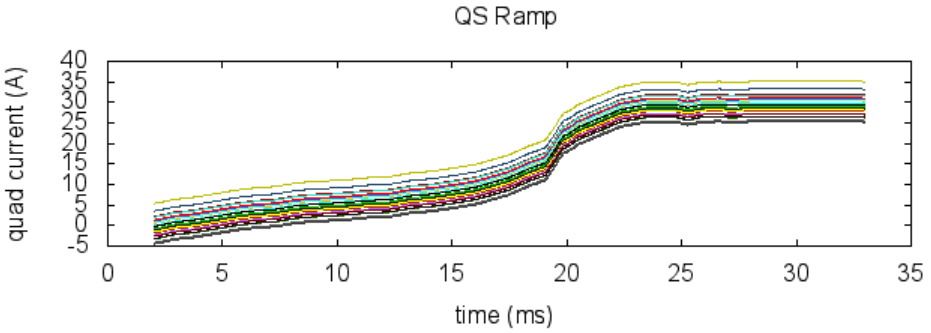
Suggested by V. Lebedev.

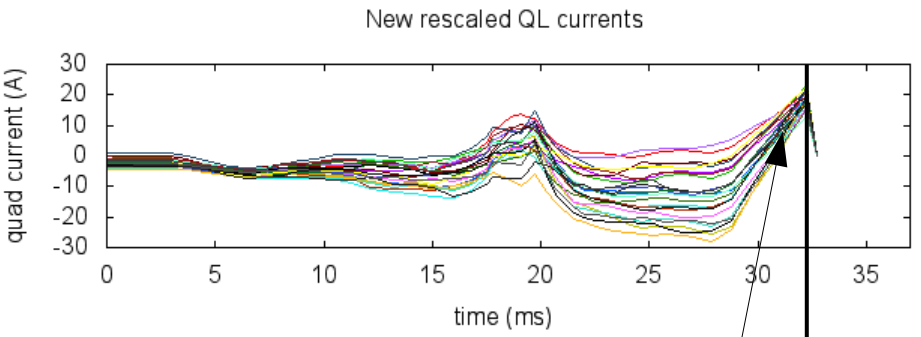
Corrections are all in the horizontal plane.

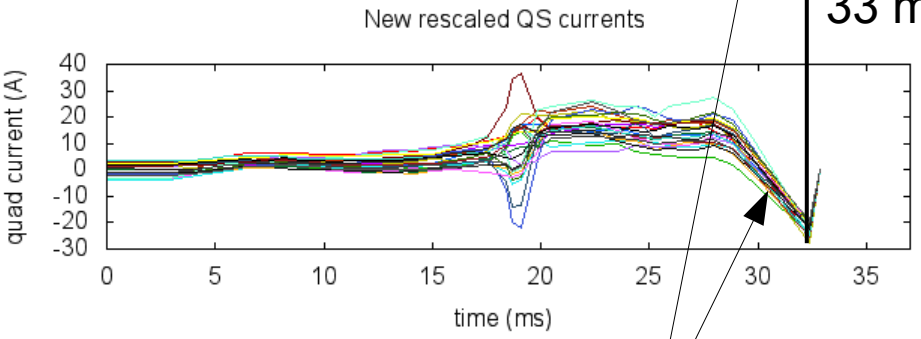
QS quadrupoles	$k_{i=j}$ (T/m)
QS24	0.403011
QS01	0.298202
QS02	0.185324
QS03	0.290666
QS04	0.372706
QS05	0.4282
QS06	-0.0412226

Corrections sent to Booster









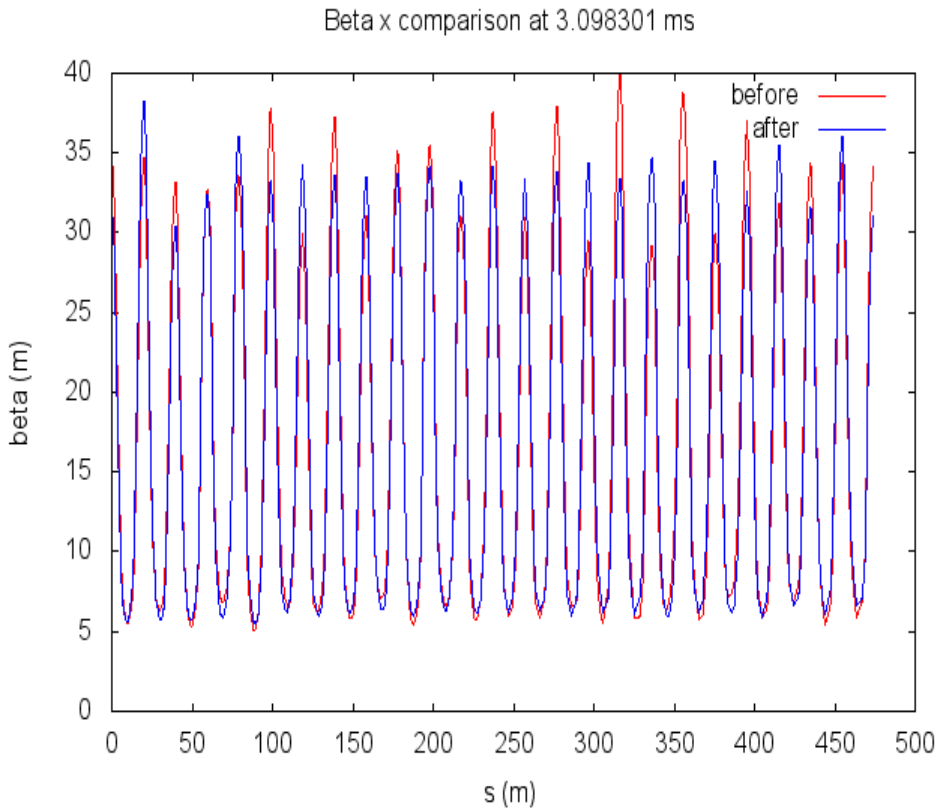
33 ms

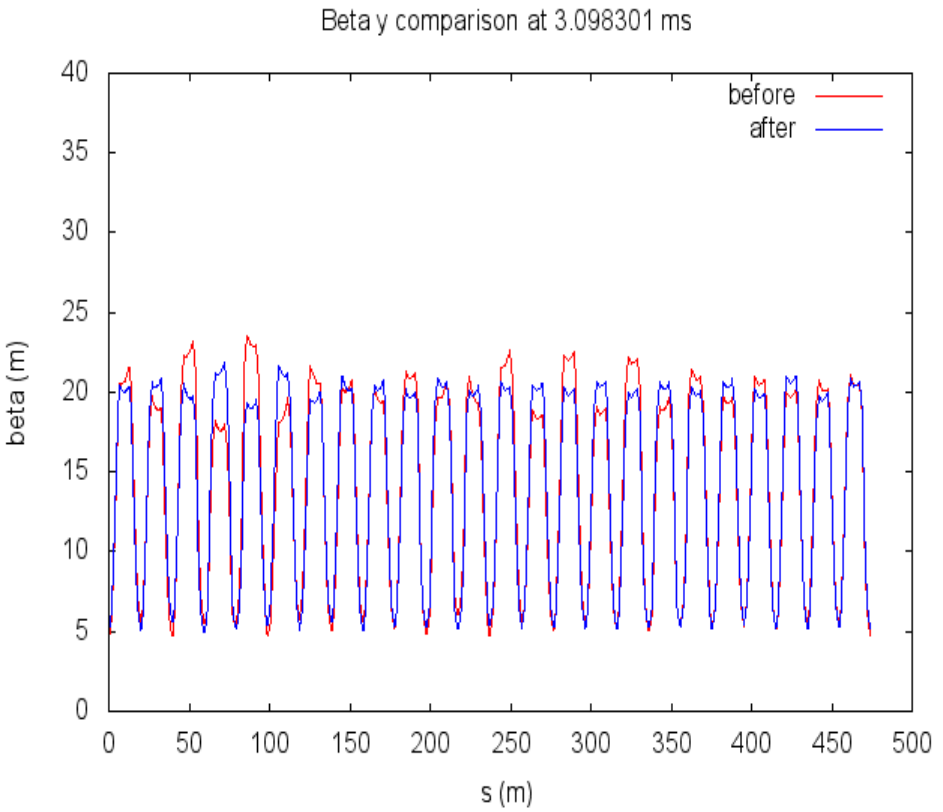
Changes are not small!

Remember this

QS horizontal
QL vertical

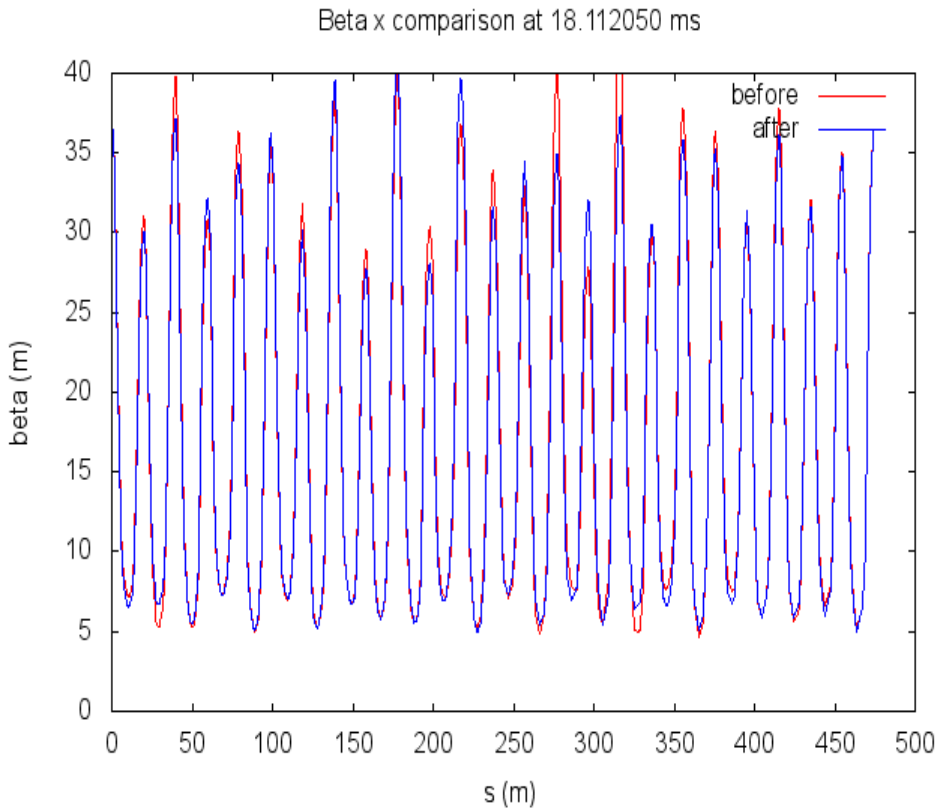
Booster lattice before and after correction (3 ms)

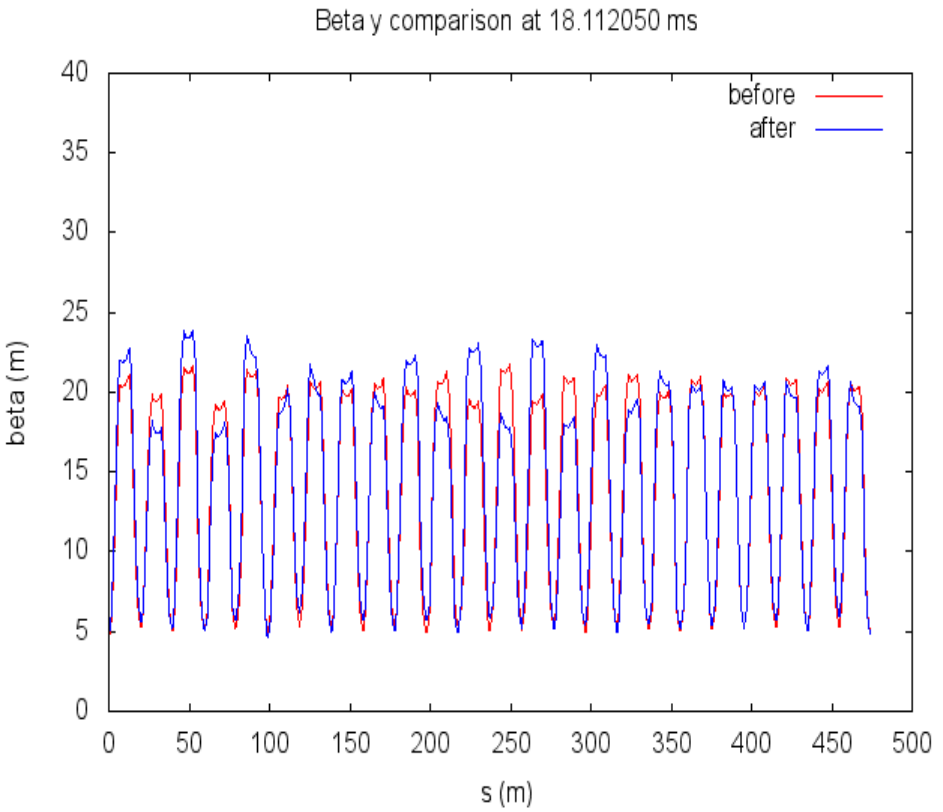




At injection: horizontal and vertical does look better.

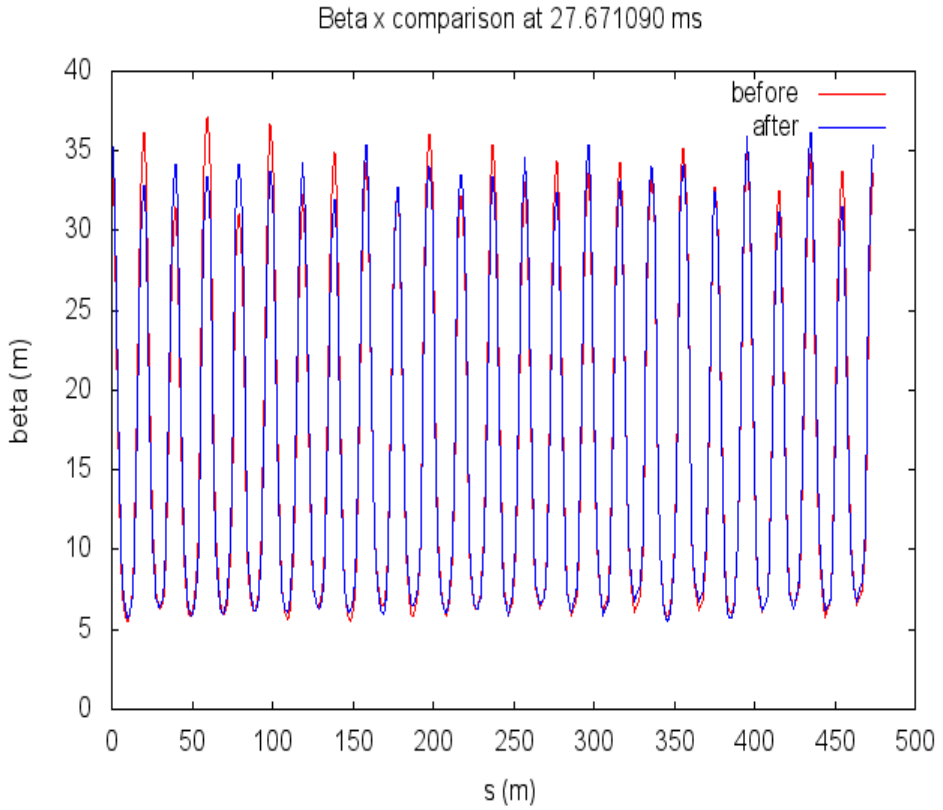
Booster lattice before and after correction (at transition)

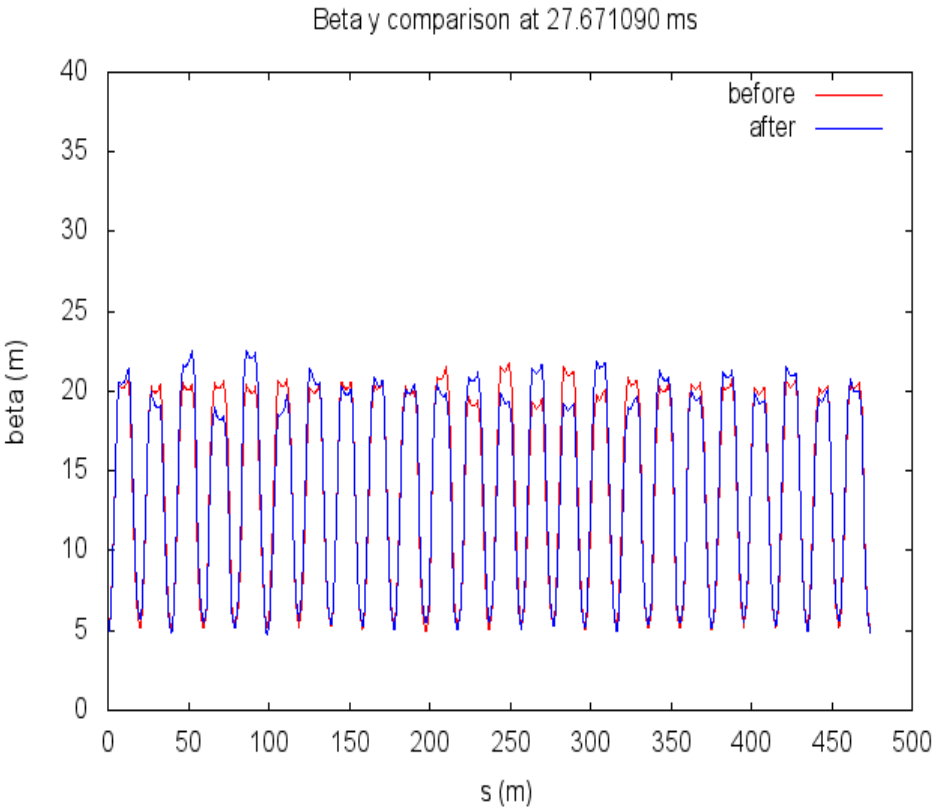




At transition: things look pretty bad. RPOS?

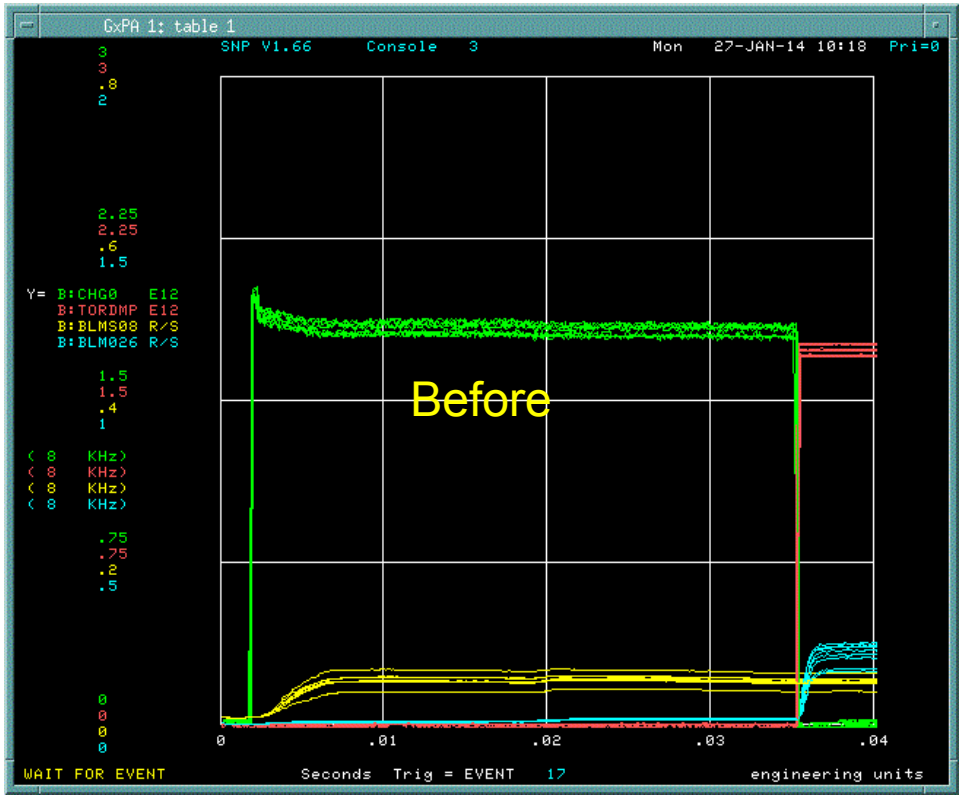
Booster lattice before and after correction (extraction 27 ms)

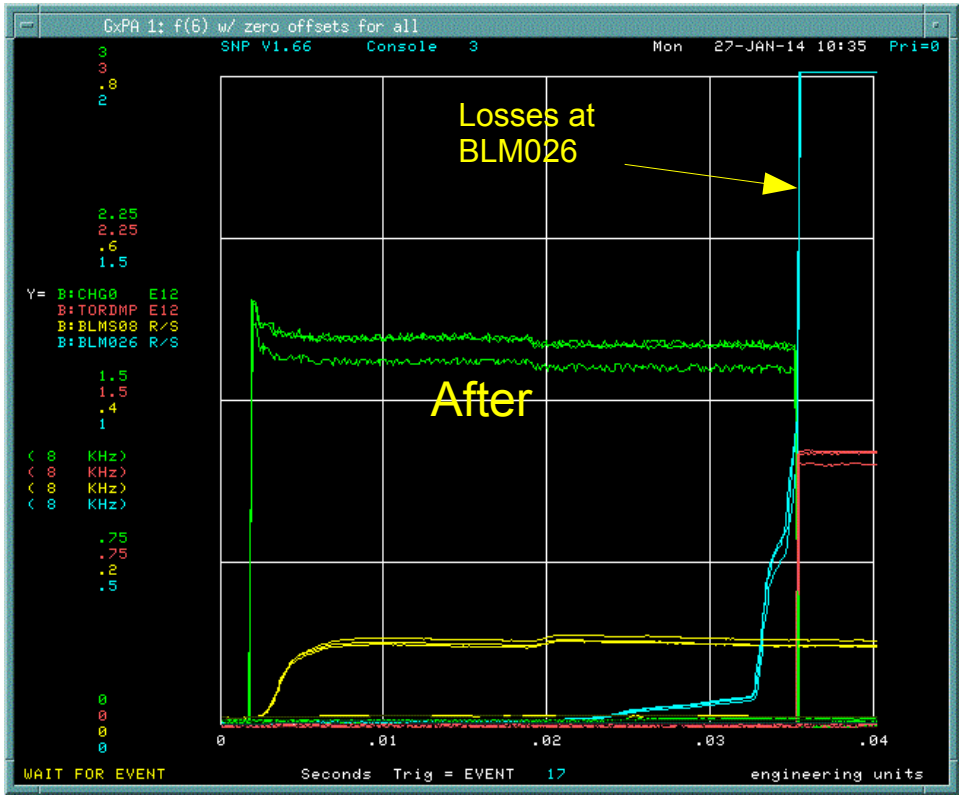




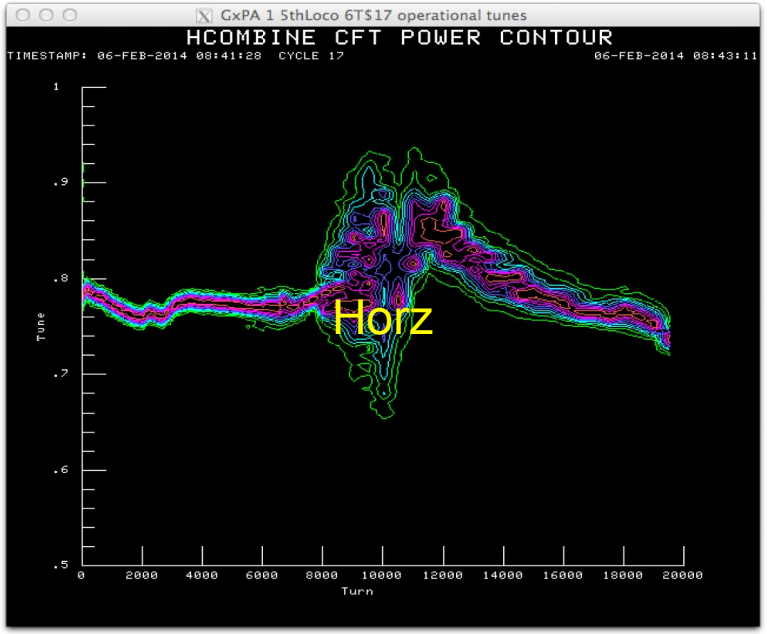
At extraction: effect not as large.

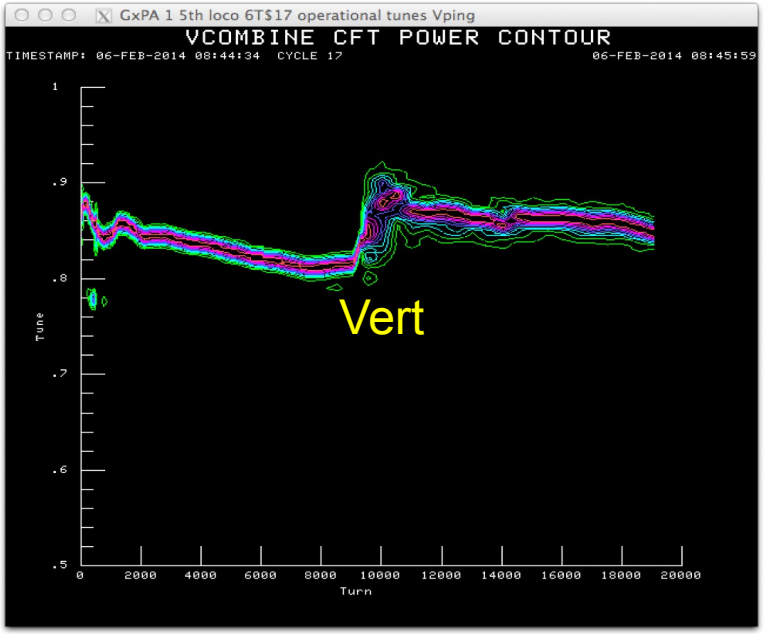
Losses before and after correction



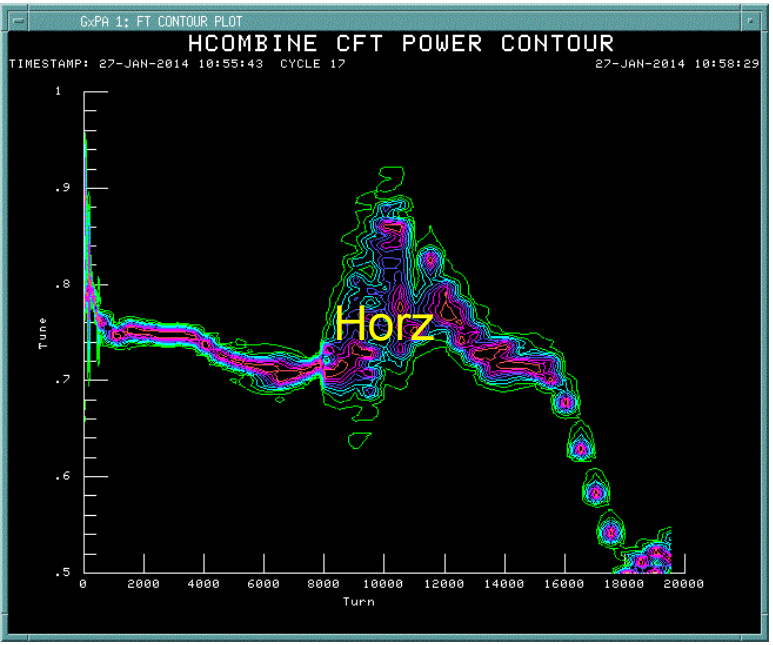


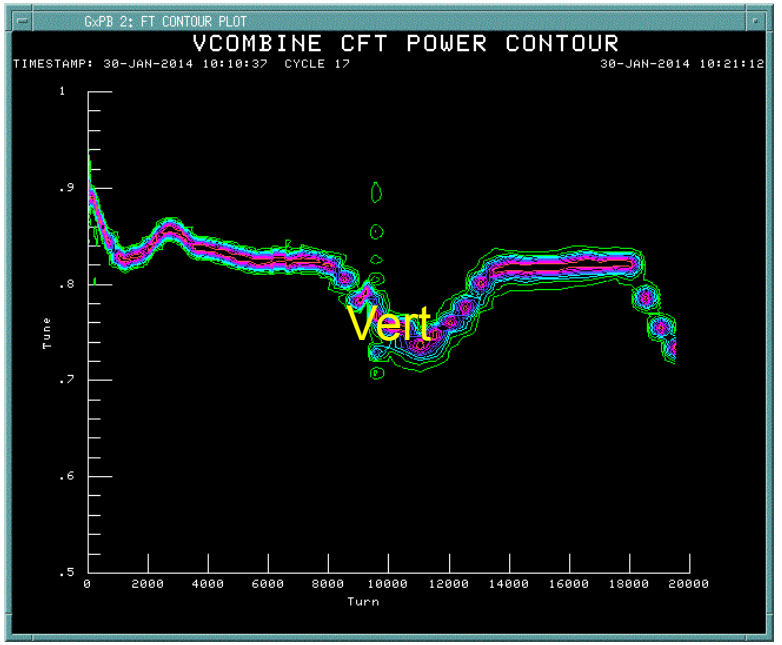
Tunes after before and after correction





Before



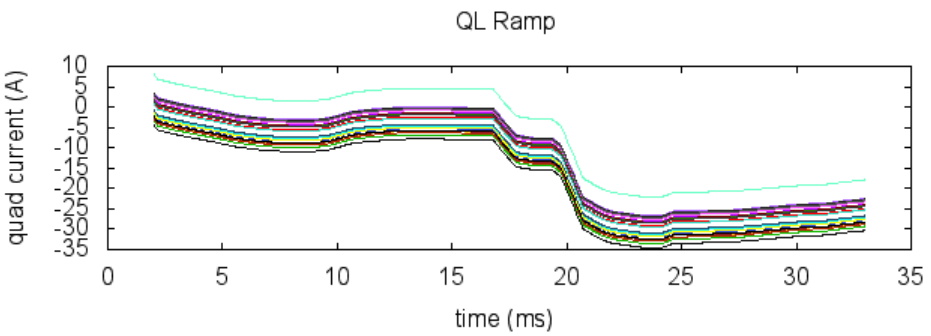


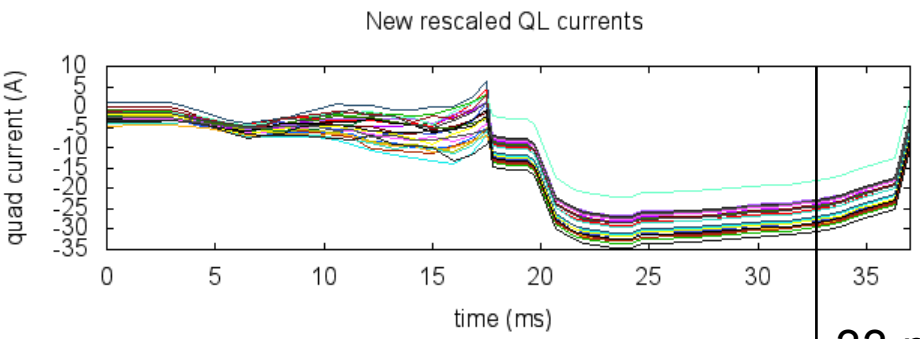
Note tune excursions at the end. Not good! Recall QL and QS current settings at the end.

Fixes

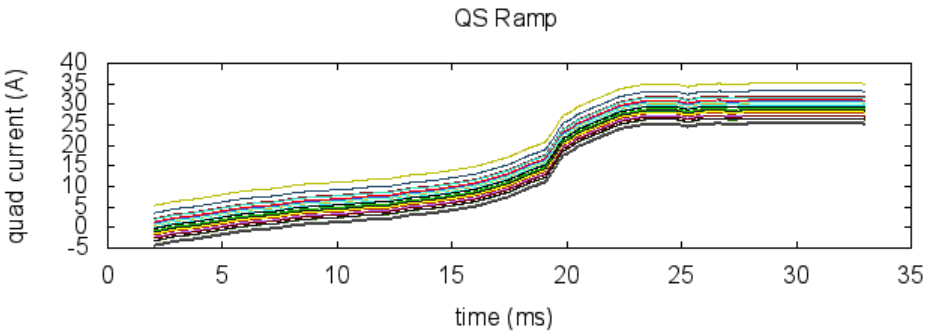
- Move tunes at the end.
  - Doing this does not help losses.
  - Plus optics after transition doesn't seem to have been corrected too well.
- Only correct optics up to transition. Leave optics as is above transition.
  - This is the option that is being pursued.

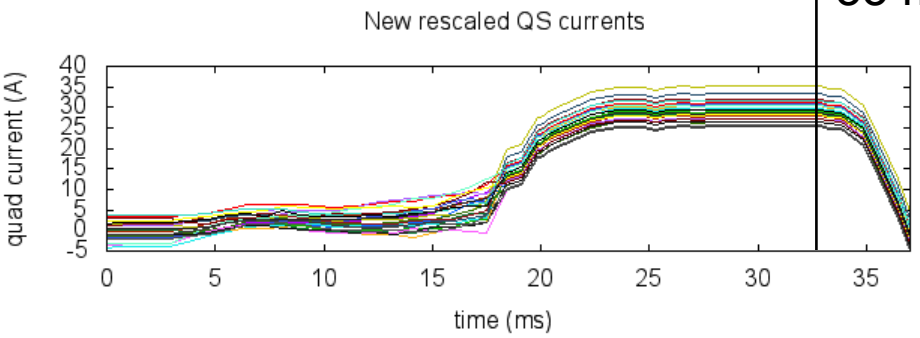
New Fix





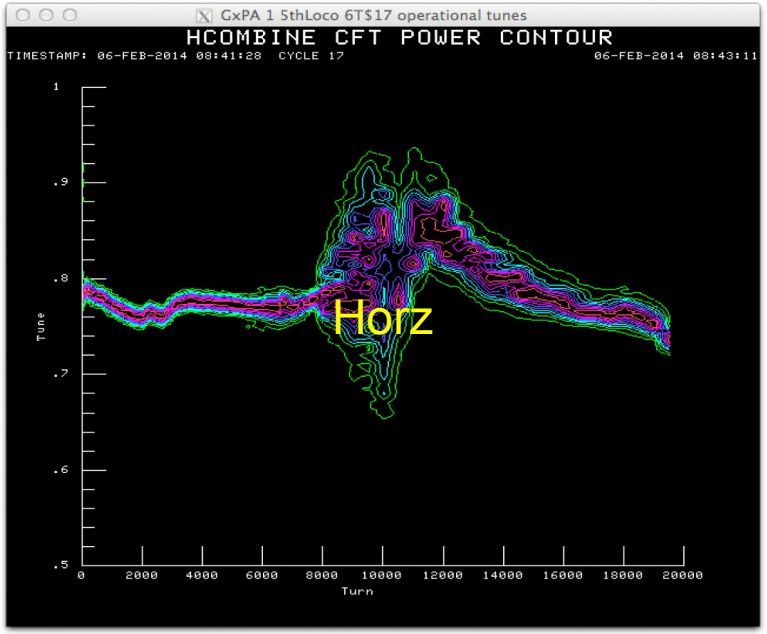
33 ms

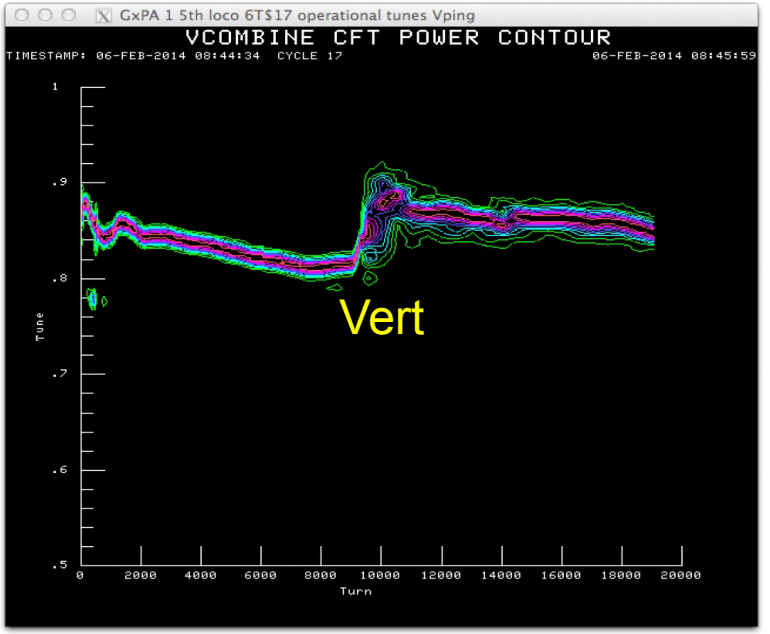




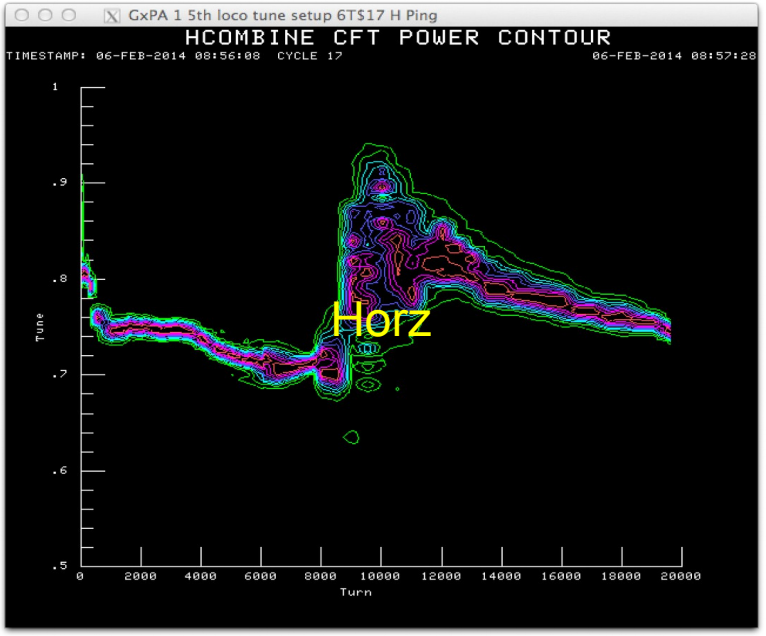
Changes not so drastic compared to previous corrections

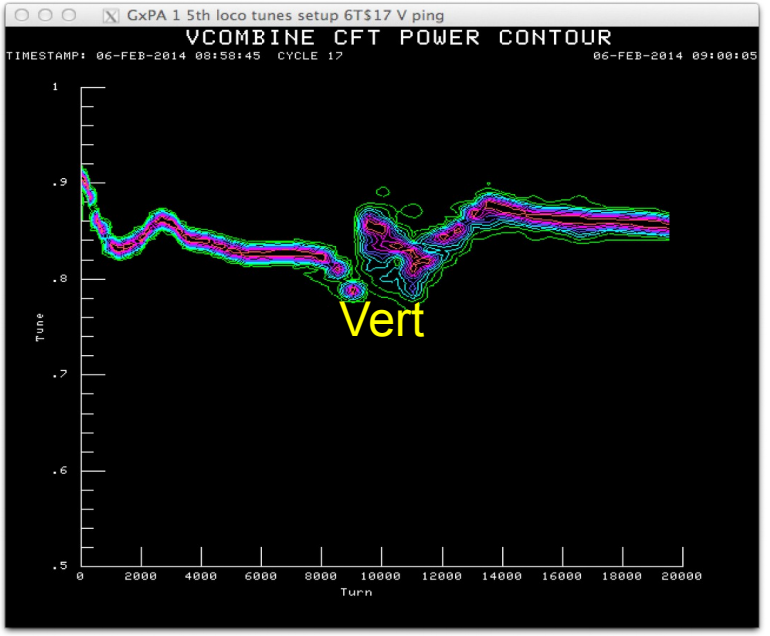
Tunes after before and after correction





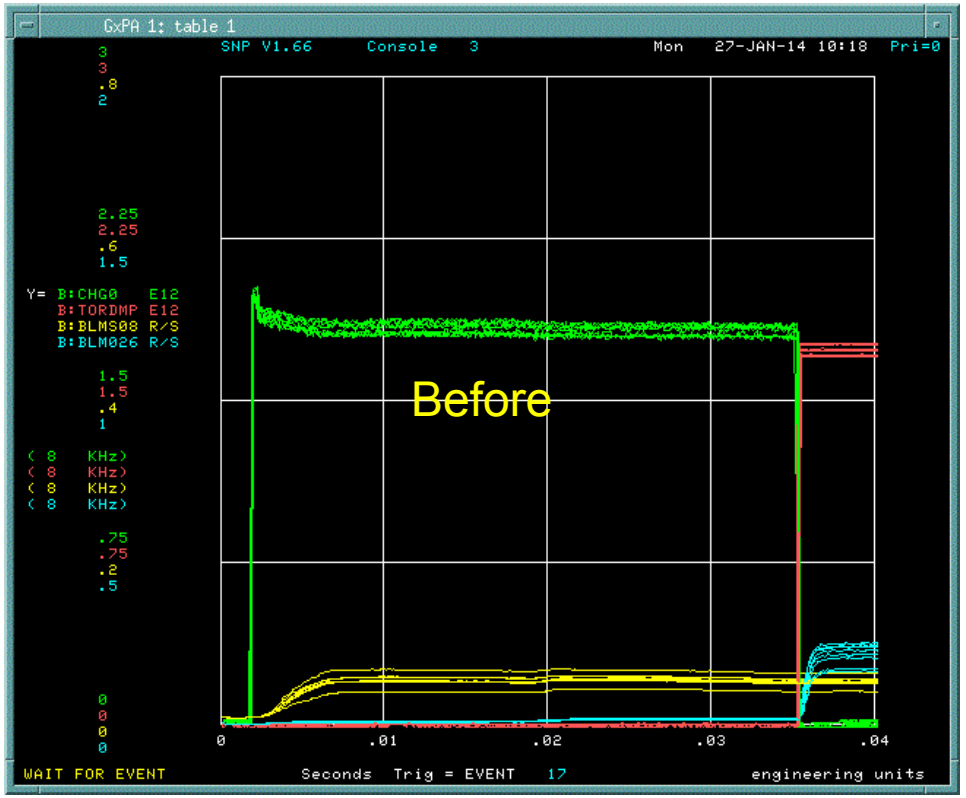
Before

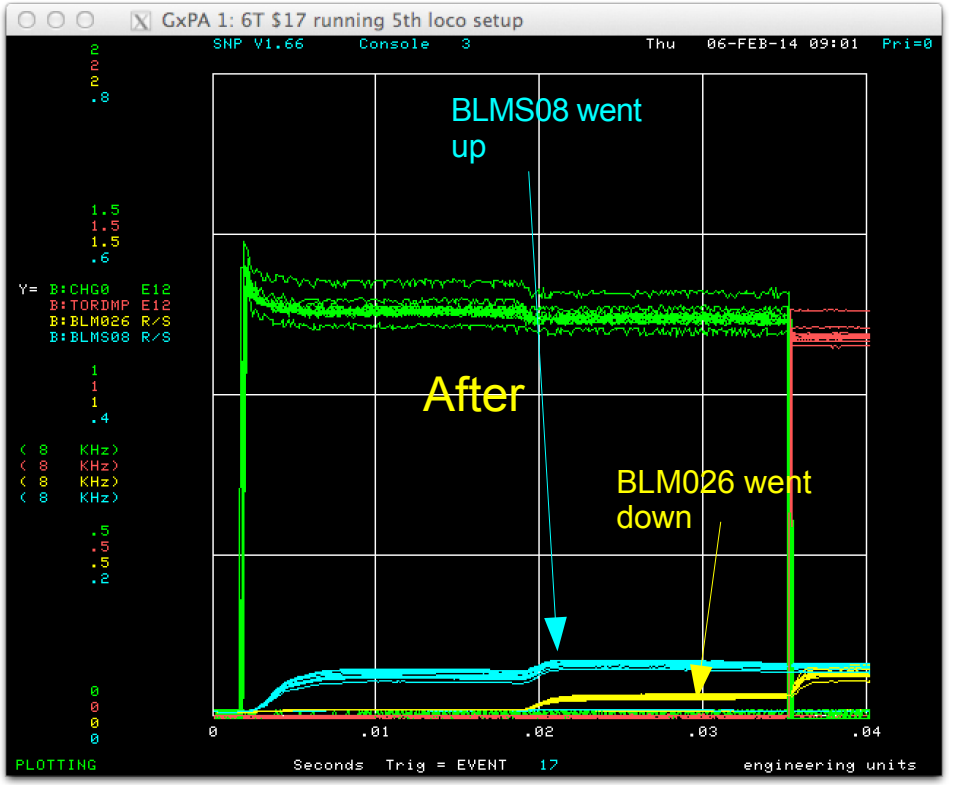




Horz tunes are lower. Need to move it in the next round.

Losses before and after correction





Probably need to move some tunes.

Some questions and observations

Jacobian

- Least squares error solution using SVD inversion of

$$\Delta \boldsymbol{K} = -\left(\frac{\partial \boldsymbol{\epsilon}(\boldsymbol{K})}{\partial \boldsymbol{K}} \bigg|_{\boldsymbol{K}=\boldsymbol{K}_0} \right)^{-1} \boldsymbol{\epsilon}(\boldsymbol{K}_0)$$

  - For Booster: solve for 442 parameters using a matrix that is 9700 ˣ442=428,7400 elements using matrix inversion with SVD.
  - Problem is over-constrained – too many equations for few unknowns.
  - Why do have non-fixed rolls of every quad, bpm rather than to have different rolls at every break point?
  - Why do we have gains for kickers and BPMs? Shouldn't it be one or the other?
- Is constructing the matrix equation the best way to solve our problem for finding the global minima? Can we recast the problem so that a Jacobian need not be calculated. For example: simulated annealing?

Plans for operations

- Get B15 have the ability to change tunes in individual breakpoints without it reverting to a prior tune ramp.
- Tune up transition.
- Increase beam current.
- Lower losses.
- Move from Table 6 to Table 1.

More observations and to do list (incomplete)

- Can data collection be faster? Right now it takes
  - $17 occurring every 1.67 s takes about 60 minutes per plane
    - For every dipole (96 dipoles)
      - Make 3 dipole changes
        - For each dipole change
          - Measure orbits 6 times
  - Theoretically
    - 1.68*(number of dipoles)*(number of dipole changes)*(number of measurements) =1.68*96*3*6= 48 minutes
    - Assuming that it takes zero time to squirt data back from the BPMs from 32 break points up the ramp.
  - The extra time comes from no beam, losses so that data needs to be retaken etc. Looks like it is as fast as anyone can make it.
  - Tricks anyone????
- Port LOCO to MPI to speed up analysis.
  - 30 nodes available. Will work on this soon.