On Tevatron TuneTracker,

Brief Status report on Uncoalesced & Coalesced Beams

Paul Lebrun

Fermilab

December 30-January 3 2002

Outline

- Goal and Scope of this project.
- Status, end of 2002
- Algorithm used in fitting, and C++/Java implementation.
- Examples of fits
- Documentation on newly created ACNET devices,
- Prospects.

- *Note : This document is rather long, slides with smaller fonts are technical details or documentation for internal purposes.*

Tevatron Tune Tracking: Goal & Scope

- A Difficult task: Automatic fits of the Tune Spectrum Analyzer data is hard, as it is just a mess of broad bump, narrow signals, and "mostly noise" (especially for coalesced beams)
- Goal : express "the art of picking the right line" into a reproducible algorithm that can be implemented on a modern computer, and can be run at ~ 1 Hz
- Scope: This could be a rather short term project, as the new Shottky will provide better information. Running at higher frequency, the device will pick-up less noise (hopefully), and we propose to "drive the beam" to measure "real coherent noise", when this device is set to measure betatron tunes.

Tevatron Tune Tracking: Goal & Scope, II

- However, this work could also be construed as a long term project: We need to establish a proof of principle that such difficult fits can be done, explore software solutions, search for algorithms..
- Ultimate goal: having the capability of provide credible information for a possible feedback loop for correcting tune/chromaticity. We are a long way from this! Yet, this project is (was ?) part of our RunII plans and is required to reach a store to store transition duration of ~ 3/4 hours. (without having to cycle Tev though 6 “hysterical” cycles)

A bit more on history and motivation

- *The first goal was simply to be able to record the Tevatron tune electronically, and automatically, (instead of having to rely on "human touch" to select the right line), and store the result in SDA.*
- *In addition, I must admit, I got curious to learn how these tunes are measured. I was told that "there is a little bit of black art in choosing the right line". I hope such this bit of secret magic can be described and implemented on a computer!*
- *The project is interesting from the Computer Science aspect: tracking 72 (or more!) tunes (and chromaticity!) independently from each others, for both X and Y planes, and do this "real time", will require high rate D.A., and significant computing power. A parallel implementation will probably be required. In addition, these fits must be intelligent enough to distinguish noise from real signal. The software must be "fault tolerant", must report "the best numbers it can come up with.." Thus, there is a little bit of an "expert system" aspect to it, as the package must be "aware of this black magic". Finally, results must be reported to ACNET.*

Brief Status and History.

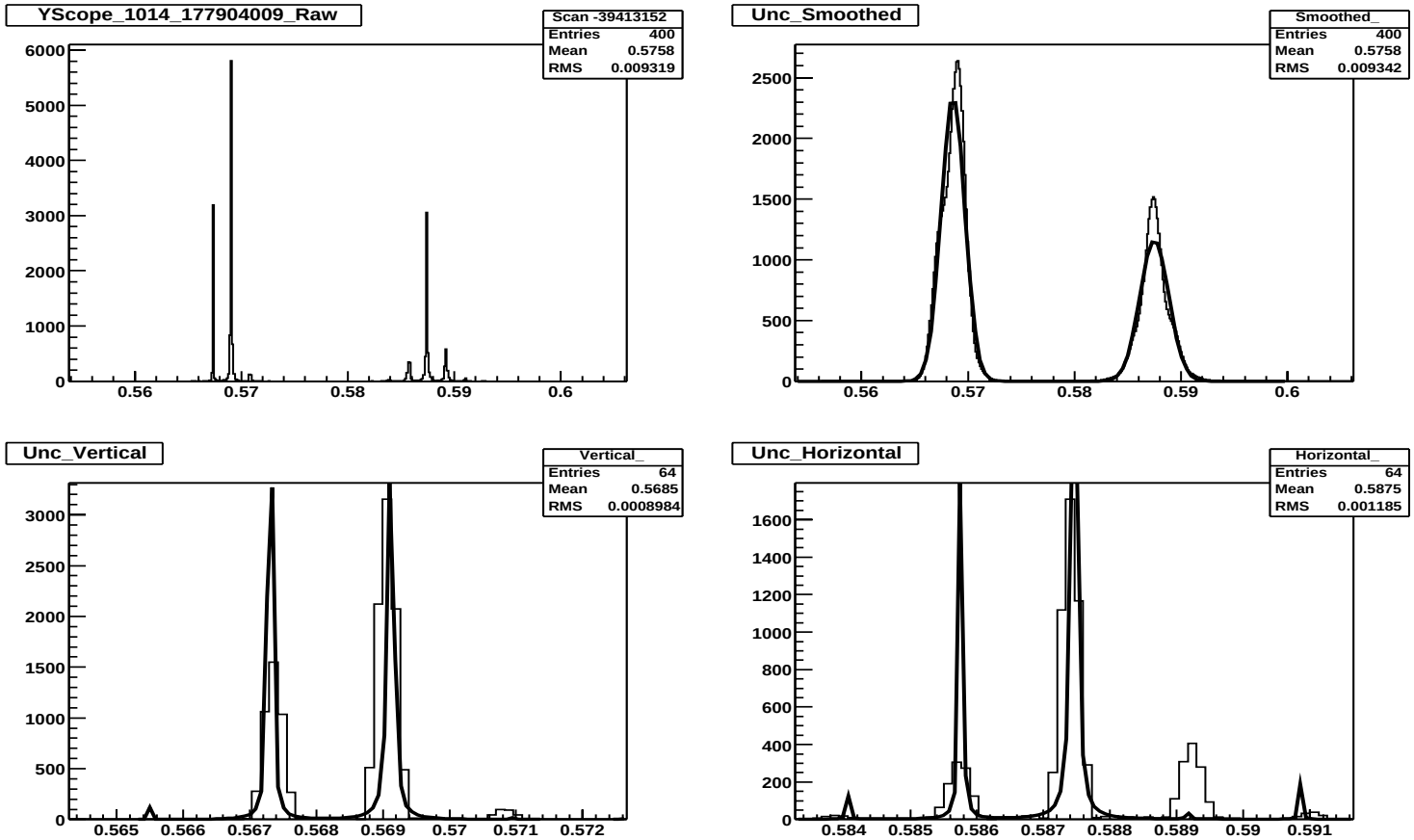
- The vsamcr files (from the C44 page) have been analyzed by John Marraffino, using a C++ root based fitting program, showing that some information could be gained.
- An HP3561A “box” has been connected to the existing proton shottky signal by Dean Still and Charlie Briegel…
- Who has also written a nice ACNET Read Wave Form utility..
- Which, thanks to Ron Renchenmaker, Kevin Cahill, Jim Patrick, … I am able to read on the development system “nova.fnal.gov”
- And fit, using the infrastructure written by John M., based on the root package.
- And, thanks to a XML-RPC based library written in collaboration BD/CDF, we are now writing the result of the fits to ACNET
- Which are “datalogged” on node “Inst2” and the D44 1Hz Archiver.

Comments of software strategy..

- *C++ has been chosen for the core fitting package, because*
  - *Compiled language, great CPU performance, with efficient use of pointers.*
  - *Language of choice in HEP, we can borrow fancy fitting package. ➔ Using ROOT, and at a later stage the new minimization package written by M. Fischler, M. Paterno, D. Sachs.*
  - *OO, a bit safer, more readable, and cleaner than C*
  - *Dynamical use of data structures prevent us from using plain old Fortran, anyway.*
  - *Java Implementation "postponed" (or rejected…) until we have good fitters with equivalent CPU performance…*
- *Java has been chosen for the reading the spectra because this is what is supported…*

  (We also wanted to learn the DAQ/Control of the futur..)
- *Interface between the C++ and Java code is a straight ASCII file. Relying on Unix I/O subsystem to sort out the locks.. (the read in the C++ code occasionally fails gracefully, as the file is being overwritten. We just try again after sleeping a few hundreds mSec.) A cleaner way would be to implement a UNIX shared memory segment between the Java Native process and the C++ process.. To be done..*
- *Writing the tune numbers to ACNET via XML-RPC, because it is callable from C++, thereby avoiding yet an other clumsy file interface.*

Uncoalesced Beam, taken during Mike Martens Tev. Tune studies, Dec 11 2002. 16:36

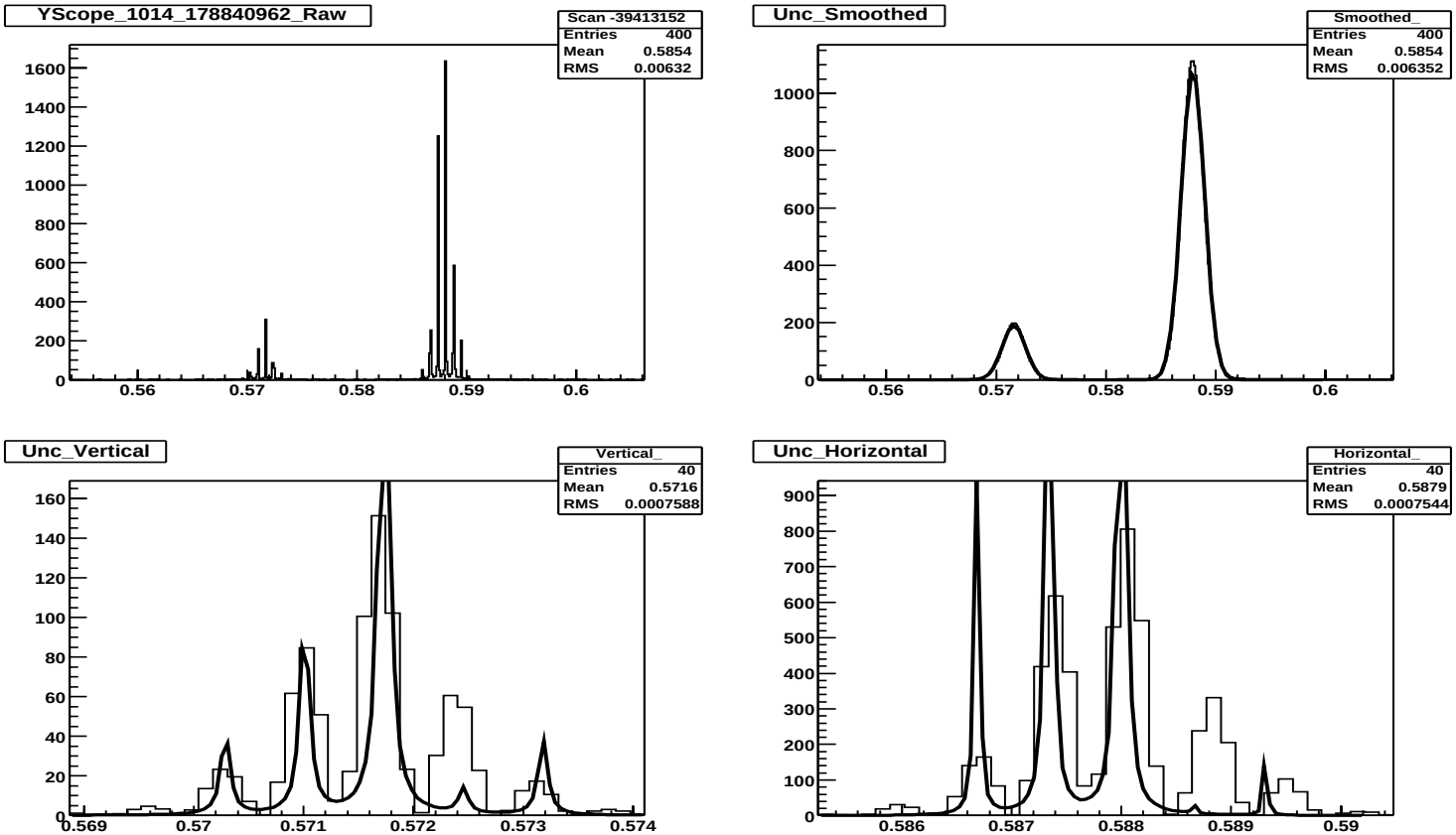


Tunes, V = 0.569115, H = 0.587459, Synch split, H = 0.001812, V = 0.00170, Predicted = 0.00168

Algorithms..Uncoalesced..

- First, Histogram, on a linear Y scale.
- Scale such the noise level (~-80to 70 db) corresponds to few counts per bin.
- Smear (or smooth), on a big scale: every bin content is spread, Gaussian wise, to neighboring bins. This is just a Gaussian convolution or “transform”
- Fit Two Gaussians. This determines the broad value of the Horizontal and Vertical tunes.
- Make two distinct new histograms, one for each region, using the original data.
- Smooth, Cern algorithm, two times.
- Fit with 5 Breit-Wigners, with same widths and same frequency splitting between satellites and main line.

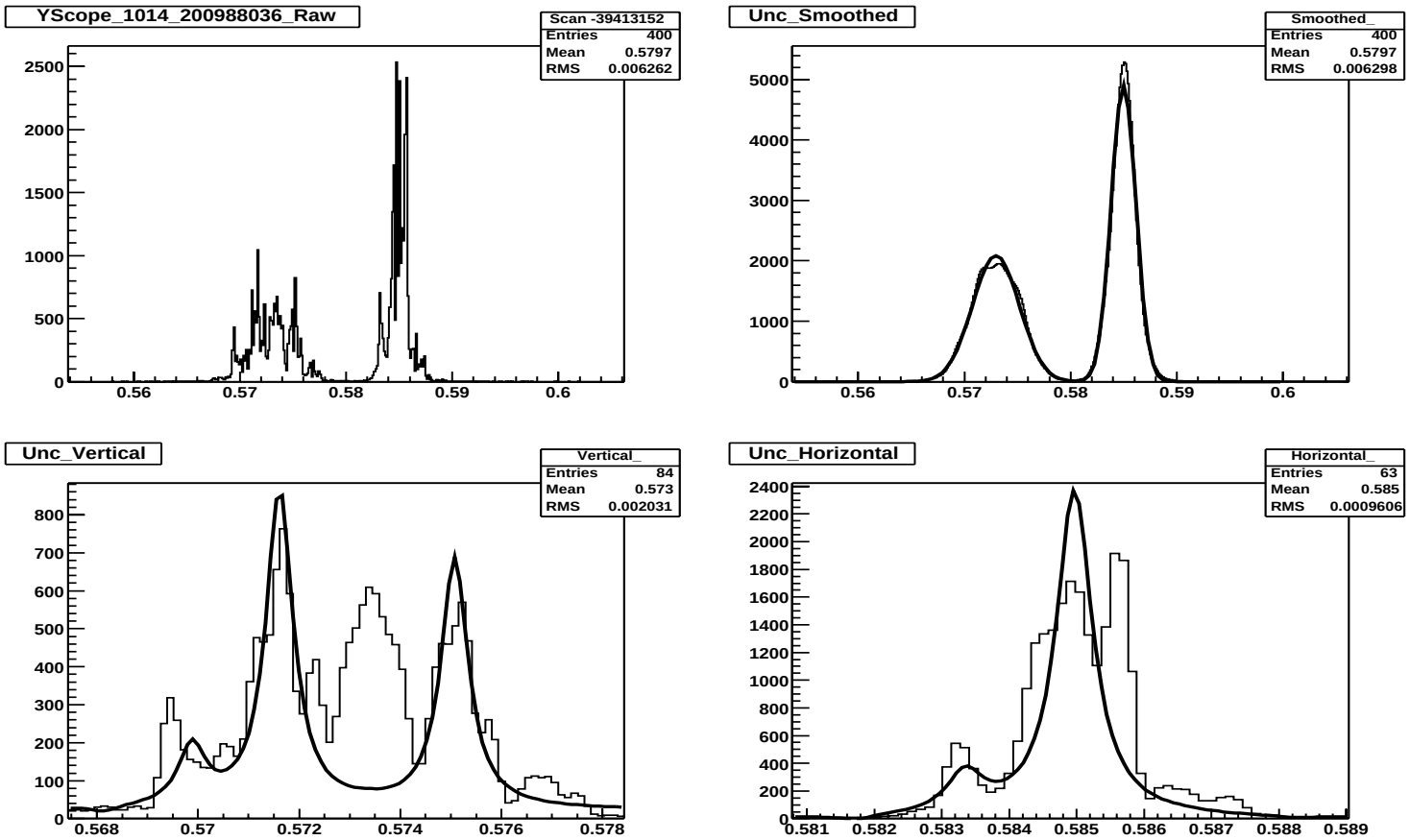
Previous plot at 150, now at 980, same beam..



Tunes, V = 0.571733, H = 0.587999, Synch split, H = 0.0007232, V = 0.000659, Predicted = 0.0065

Back to 150, a bit later..

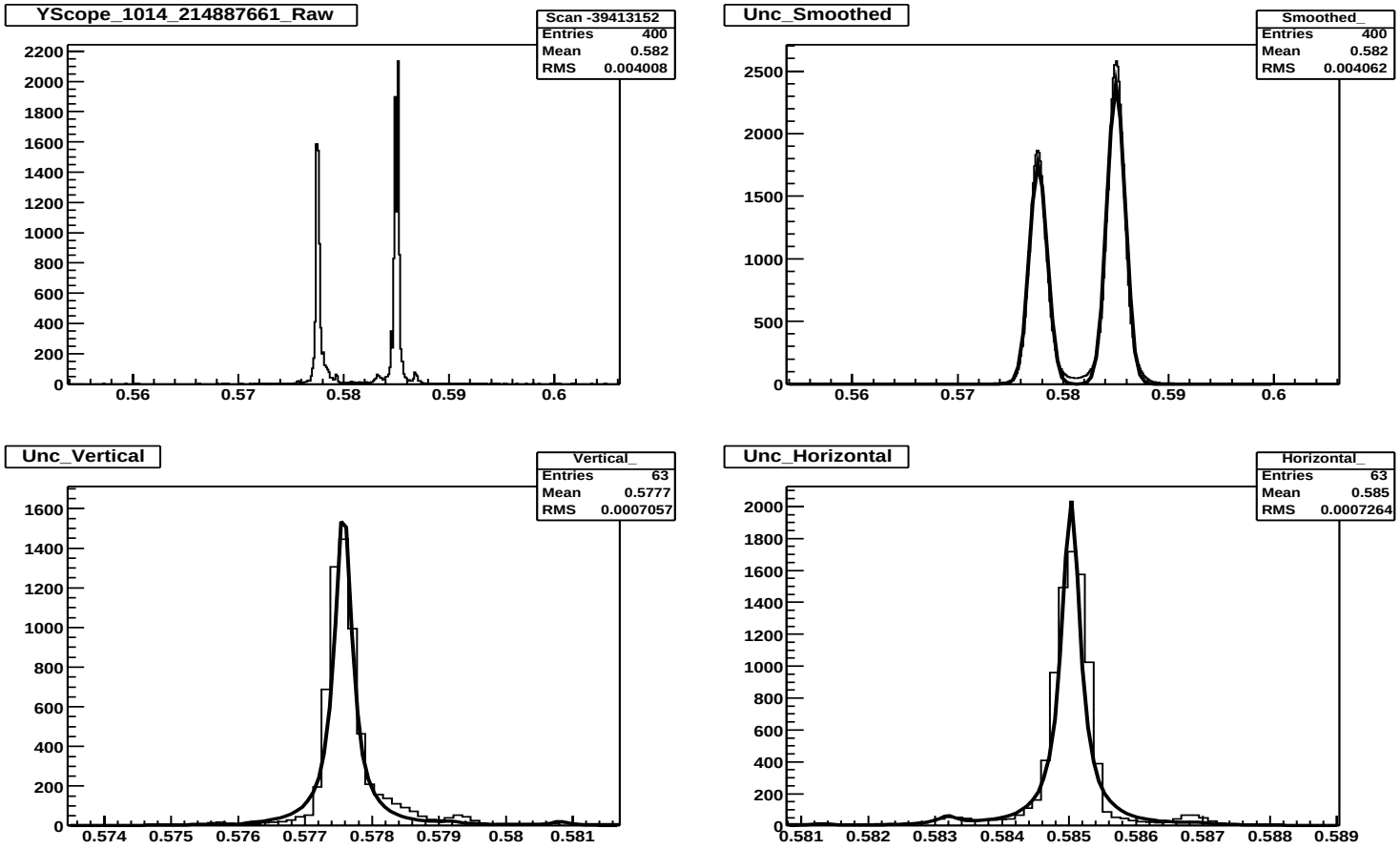
23:04, Dec 11



Despite missed bumps, Synch split, H = 0.0017312, V = 0.0016207, Predicted = 0.00166

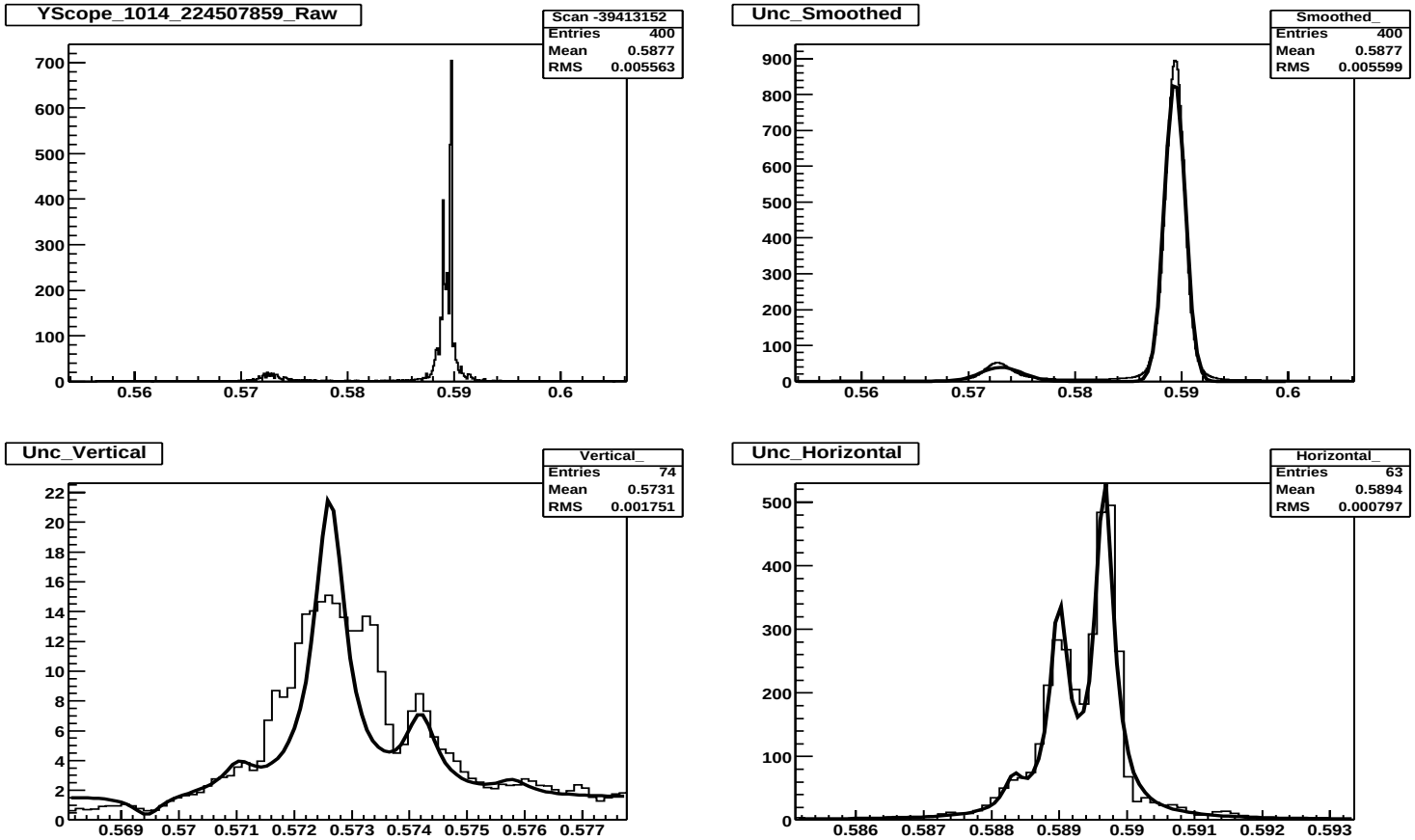
Owl shift…

05:05, Dec 12



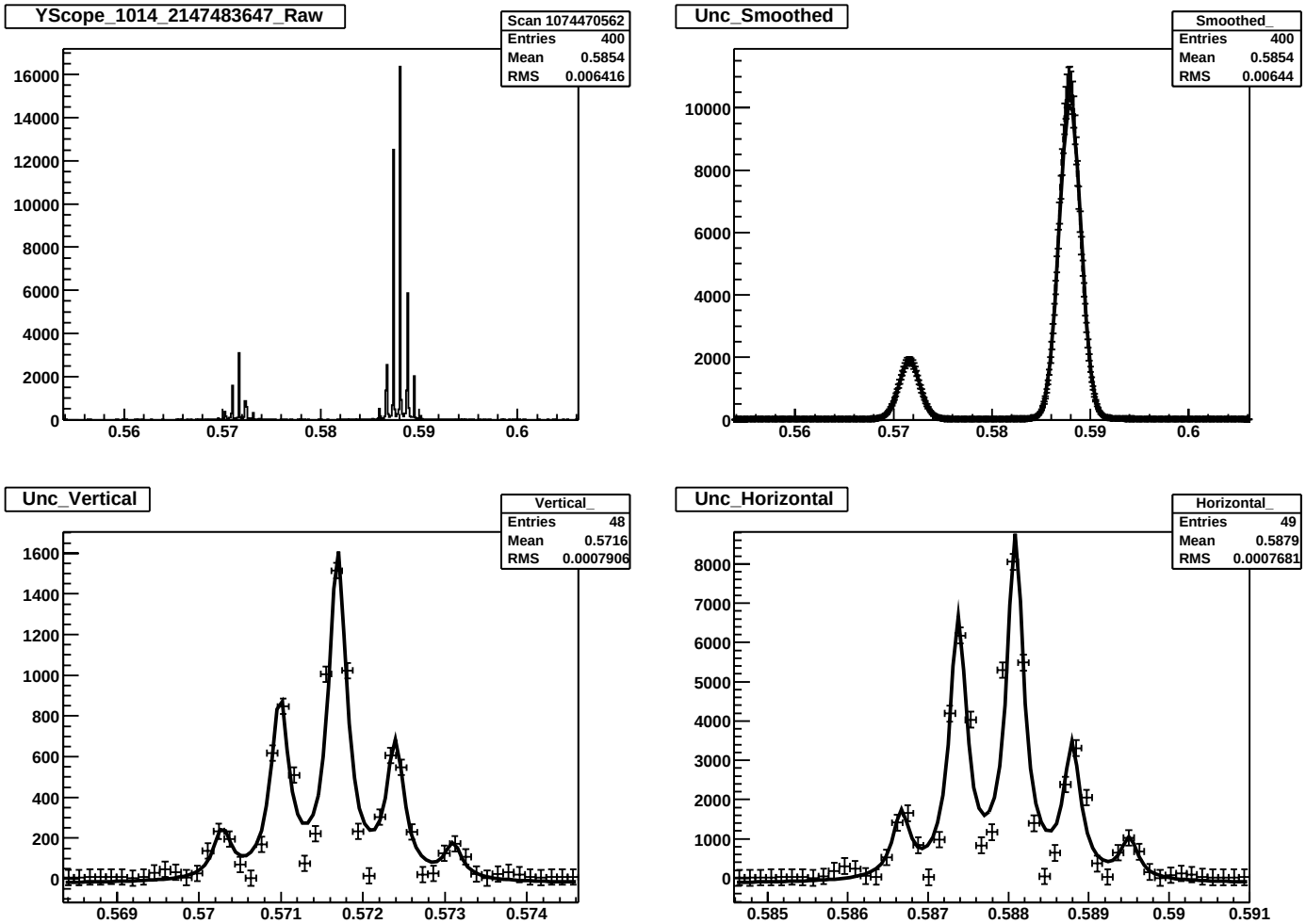
Despite weak bumps, Synch split, V = 0.001799, Predicted = 0.00165

Owl shift…



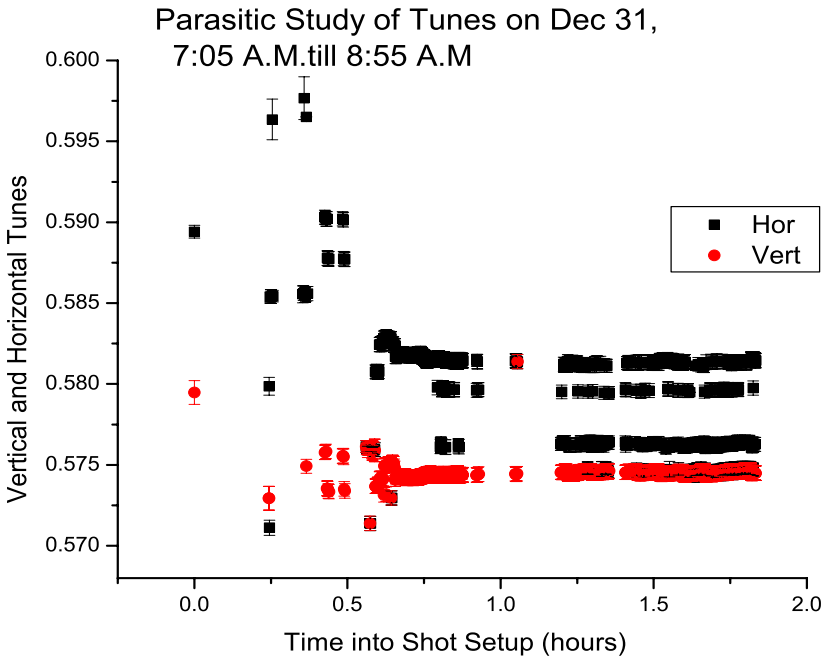
Despite weak Vertical signal (no tickling, I presume) (10 db above noise),
we got a meaningful Vertical tune measurement, Sync-beta (0.00185)

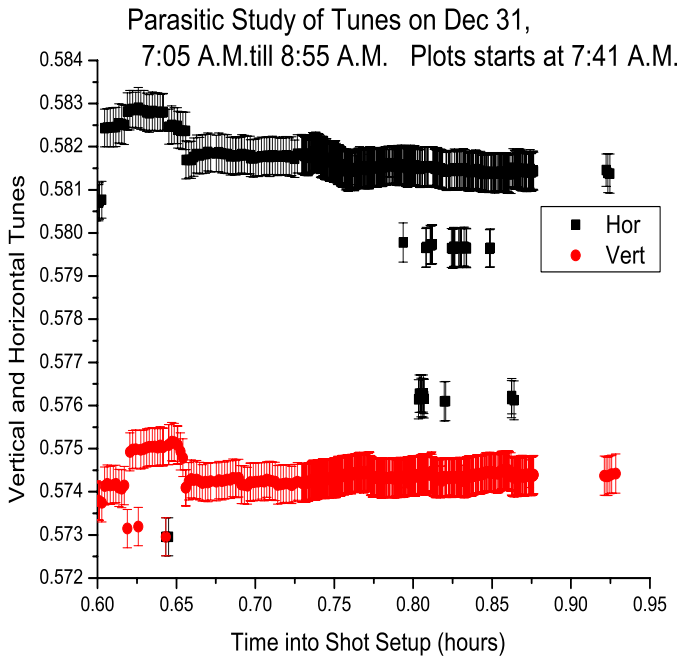
Same data, re-analyzed after algorithm improvements.



Vertical tune = 0.571692, Horizontal = 0.58808

Parasitic Studies in MCR, During regular Shot Setup.





Fitted tunes were “data-logged” (node Inst2) during the shot setup for store 2115 on December 31 2003. During the first 20 min or so, tunes were changed abruptly by operation for standard chromaticity measurement. The tune was completed around ~ 7:40 A.M. For ~ 1.5 hours, the Tev was “left alone” with coasting beam.

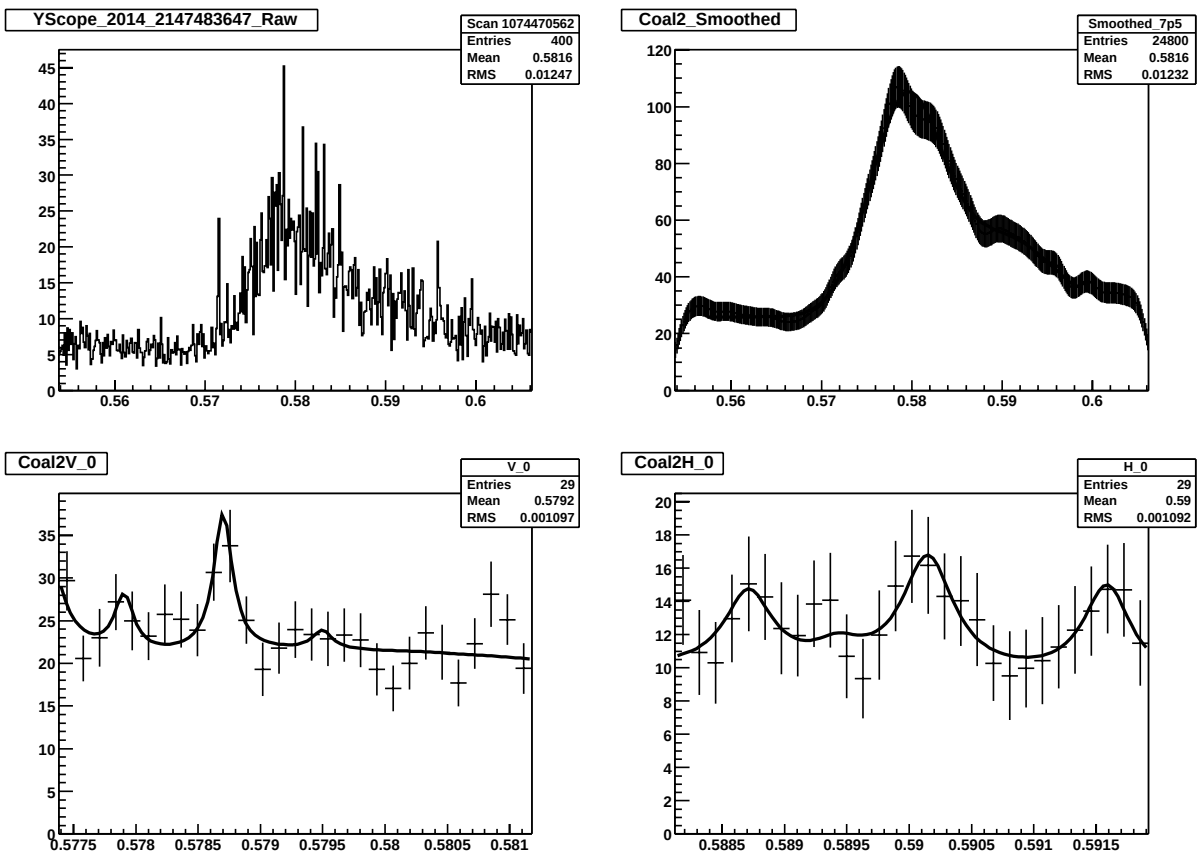
Further comments:

- *The C++/Root fitting package was restarted on nova.fnal.gov on numerous occasion, to improved the reliability of the history buffer and usage. Hence the time period with no data. (We will do this again!)*
- *This history buffer is of no use when the tunes are changing rapidly. However, they are not changing unexpectedly : The operator was changing them in a tentatively predictable way. Therefore, more software could be written to properly seed the fits.*
- *After tuning, tunes drifted slowly, towards each other. This is due a well known and documented feature of the TeV magnets.. (see plot on the right)*
- *The Horizontal tune is not perfect: occasionally, the "wrong" (lower) line is chosen, because the two tunes came a bit too close to each other, and the horizontal signal on the vertical pick-up (used in this case) was weak.*
- *However, this could be fixed by analyzing the signal coming from the horizontal strips.*
- *The sudden change in the measurement rate at t= 0.725 (07:48 A.M.) was due to a change in running mode: no display were requested, nor voluntary delay to look at the plots. The update rate was close to 0.8 Hz, close (or close enough) to our target. (the integration time of Spectrum Analyzer is about 1.25 Hz, no point trying to go faster…)*

Tevatron Uncoalesced Tune Tracking:

- Certainly do-able !
- No reason not to do it. Therefore, we should:
  - We should migrate this prototype to a stable version, running on a dedicated node
  - Get an other HP3561A (or equivalent), and D.A., so the tune can be tracked on both plane correctly.
  - Write a prototype for a “X-Y tune referee” software, to arbitrate the values from the two scope for a given plane.
  - Start to think about a feedback loop (will require numerous dedicated studies.), once the Tune tracker is deployed.

Coalesced, p-bar beams is much harder!

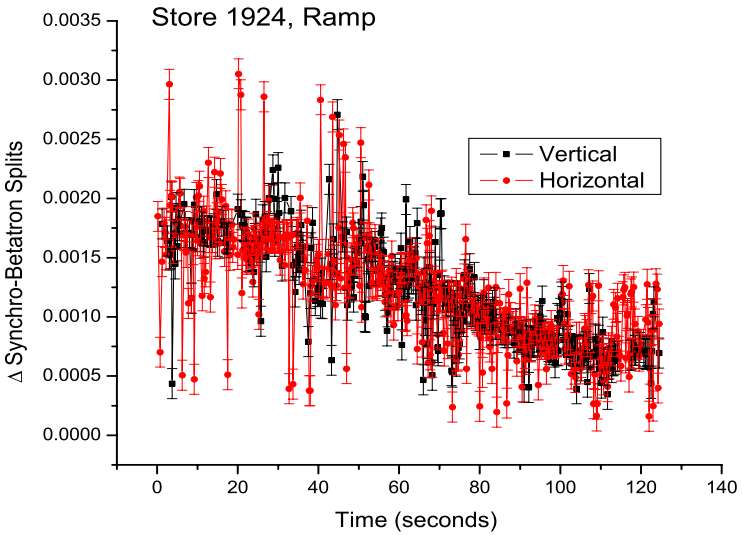


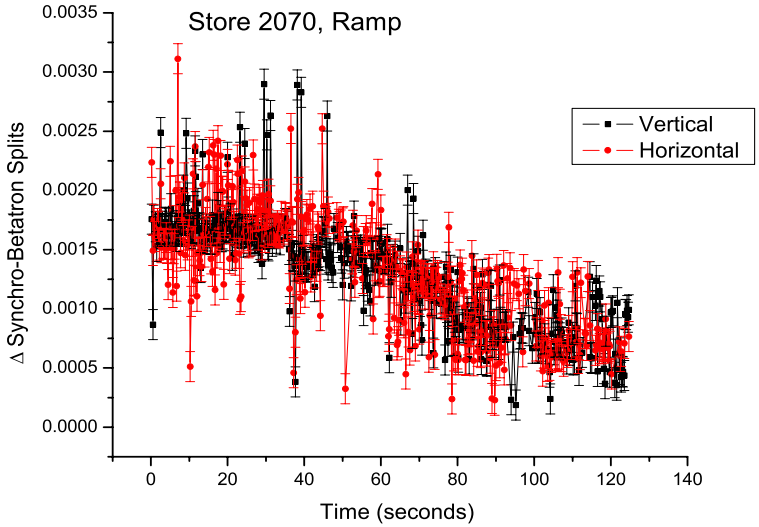
- Data taken on Dec. 16 2002, 11:38 A.M. (store 2078, ~ 2 hours into the store).
- Nothing but noise lines at this point???
- There is more than one tune !
- How do we establish a signal?
- Note : these lines are clearly beam related!

Algorithms..Coalesced Beam(s)

- Overall scenario identical to Uncoalesced. The differences are:
  - 3-Gaussian fits for the broad tunes (instead of 2). The highest tune will be ignored (this needs work, which broad signal to consider the most important relevant one ?)
  - We do these on three different Gaussian-convoluted data sets, with 7.5, 10 and 12.5 bins average, and compute averages between the fitted values. (again, such an algorithm is highly negotiable..)
  - Fit with 5 Breit-Wigners for narrow Synchro-betatron confirmation: the central line (most intense) is allowed to have a different width than the satellites.
- All cases of Coalesced beams are treated identically, although the fitter is aware of the SDA Case name, beam current and of course machine energy.

Do we see these Synchro-Betatron lines? Only on a statistical basis!

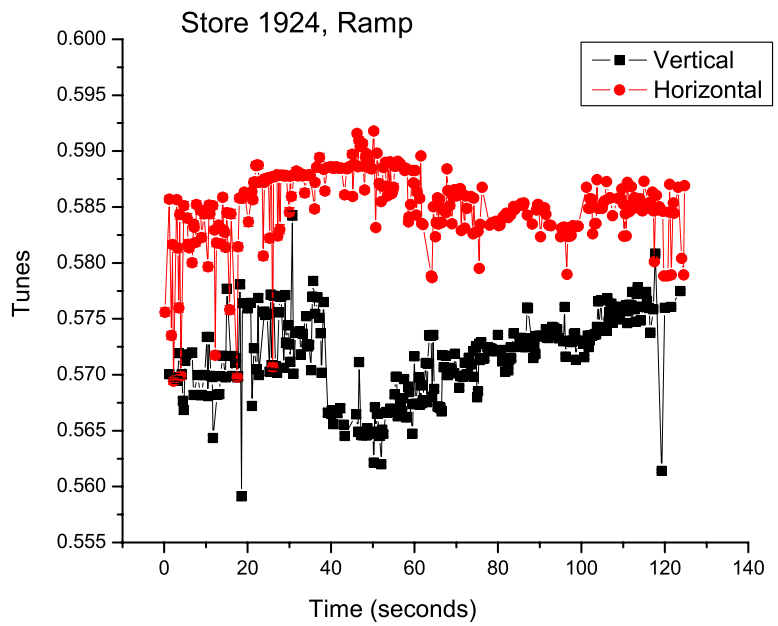


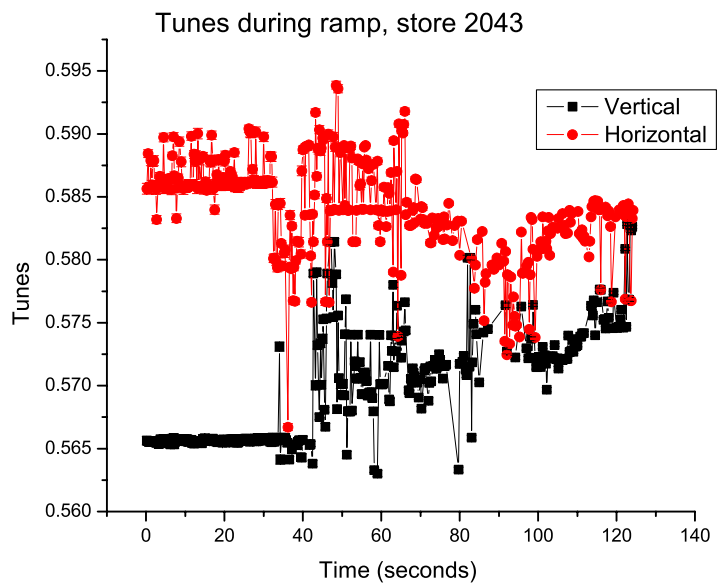


The fitted values for the synchro-betatron line tune split Δsb is plotted versus time during the ramp for store 1924 and 2070. Only results from valid 5 B.W fits (significant amplitudes for the main and at least one satellite line) enter the plot. Fits with a 100 % relative difference between the predicted Δsb and the fitted one are accepted. Yet, a broad band is clearly visible on these plots, centered (within ~10%) on the correct value of Δsb .

If we "seed" the fits with the wrong Δsb value (nominal x 1.25), this band still appears.

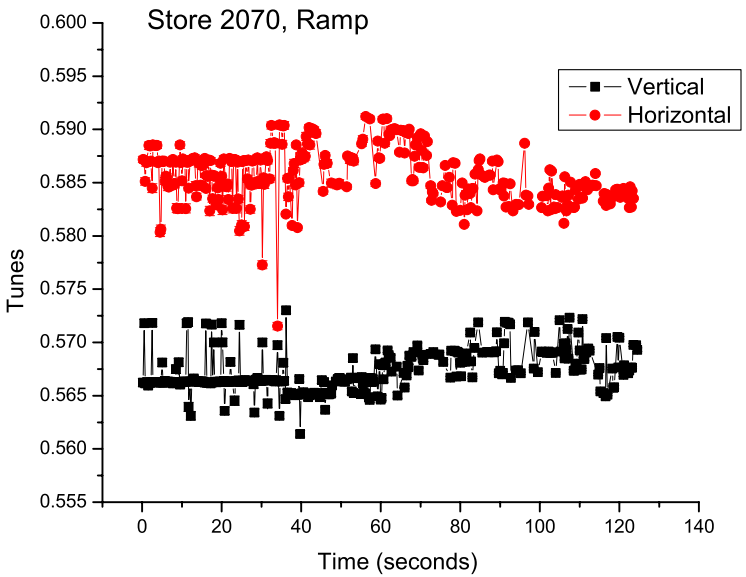
Tunes during the ramp for some stores.

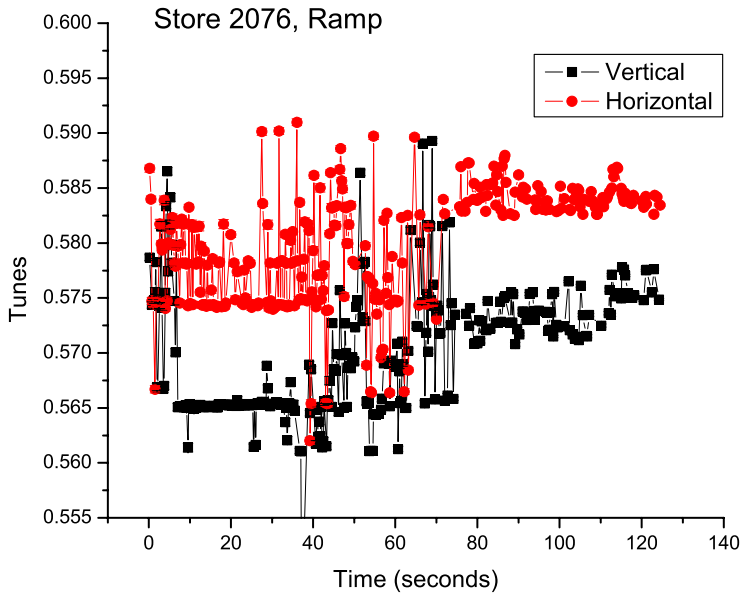




The fitted values for the Δsb tune split has to within 30% of the nominal value. The final tune is set by the sb line closest to the broad tune obtained from the Gaussian Convoluted fits. The algorithm for both planes are identical. Only about 60 to 65 % of the 500 frequency scans from the vsamcr spectrum analyzer have a valid 5BW fit (in at least one plane).

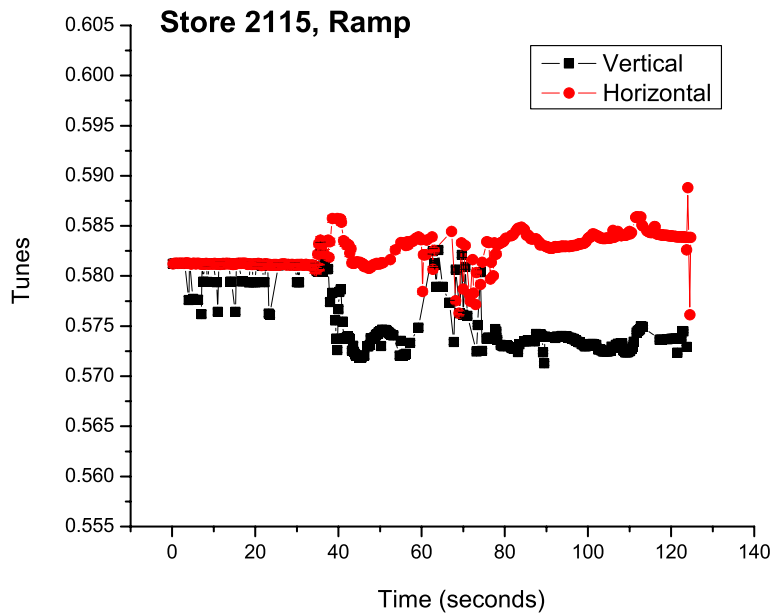
Tunes for the ramp for some stores (2070, 2076)

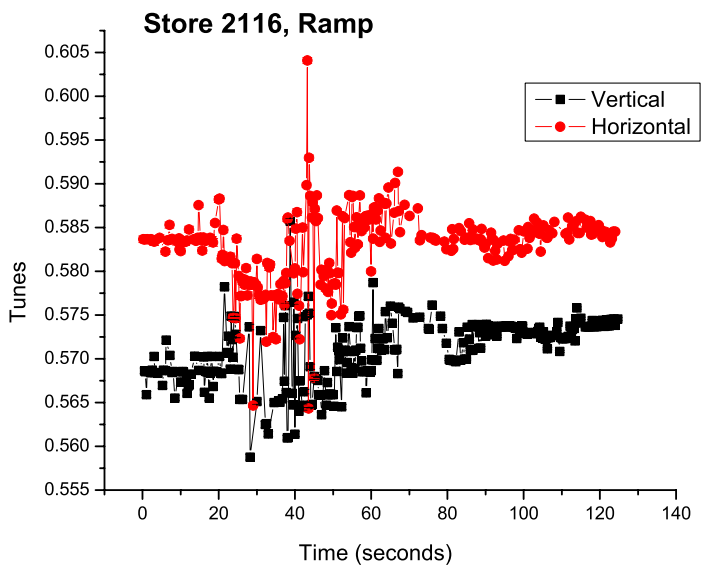




Sudden fluctuations by as much as 0.005. Worse, the vertical tunes are sometimes taken as the horizontal tunes. However, for such beams, the tune spread due to transverse amplitudes, and possibly large chromaticity could in fact explain why some bunches (or excited fraction of some bunches) oscillate at different betatron frequencies.

Tunes for the ramp for some stores. (2115, 2116)



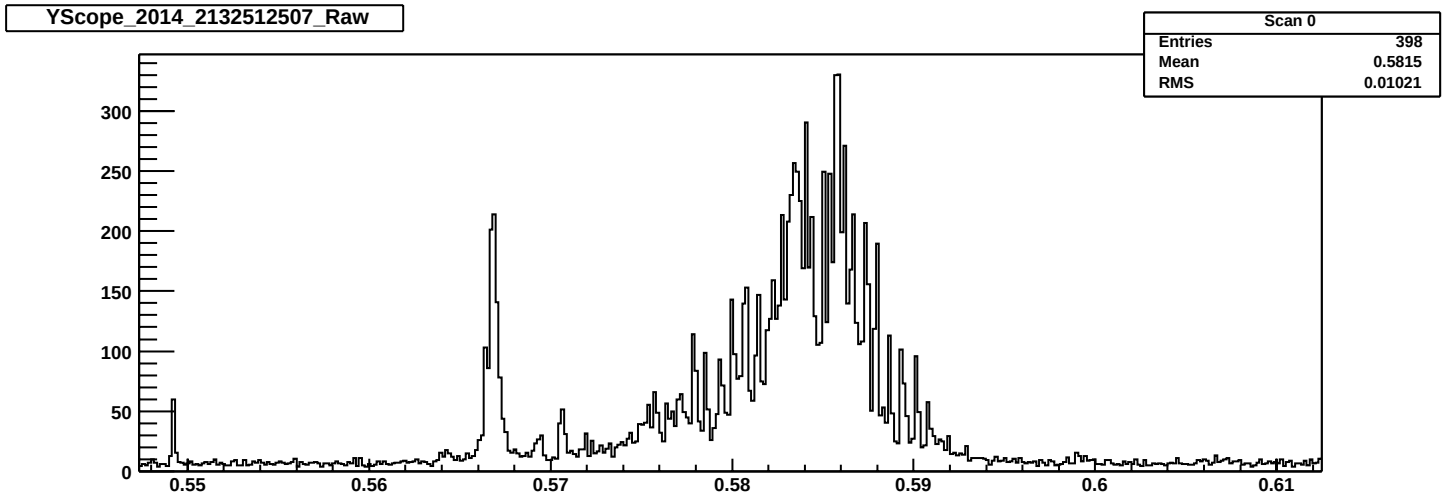


Despite these uncertainties, these tune excursions from nominal values could be of interest. A more systematic studies of such graph, along with step efficiencies and emittance growth measurement should be undertaken. Note: store 2116, characterized by lower vertical tune at 150, some excitement during the ramp, and with somewhat large proton longitudinal emittance did not last very long (quench at A11 shortly after reaching flat top) Store 2115 was a 2x0 prior to that.

Back to “noise” or “Signal” issue…

- It would be good to make this signal a bit more convincing!
  - “tickle” or “drive the beam resonantly, a bit, to increase signal. How much can we afford without blowing the emittance? Need a dedicated study, Warren Schappert and Dave McGinnis will do this.
  - Or, may in conjunction, we could look at the relative phase betweeen candidate lines.
- So far, only the “scalar” signal analyser has been used (I.e. we use the “vector” signal analyser in scalar mode, for sake of expediency). We could set the vsamcr device in “vector mode”, and collect data. True, we do not have a reference signal, but may be we do not need one, since the evidence for coherent synchro-betatron oscillation is in the relative phase between the main line and the synchrotron line(s).
- Or get a new “vector” signal analyser, so that we do not disrupt the vsamcr device..

“Ghost Lines”.. True noise? A real nuisance!.



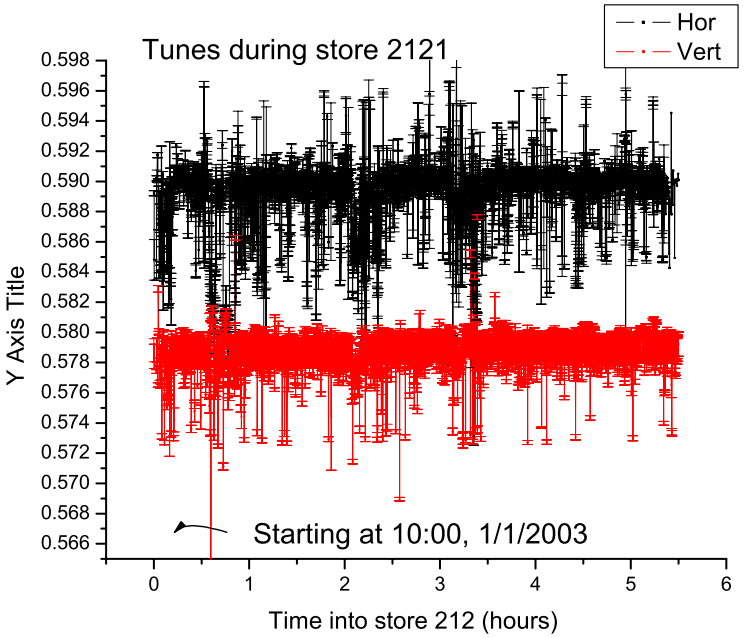
This is a snap shot of the raw spectrum (linear scale), taken during During store 2123 (January 2 2003) Multiple lines are visible. The narrow line at 0.549 is clearly “noise”, as this is outside of the tune map for coasting beam. Moreover, If we suddenly turn the beam off, they disappear! Thus, yes it is noise, but beam related noise.

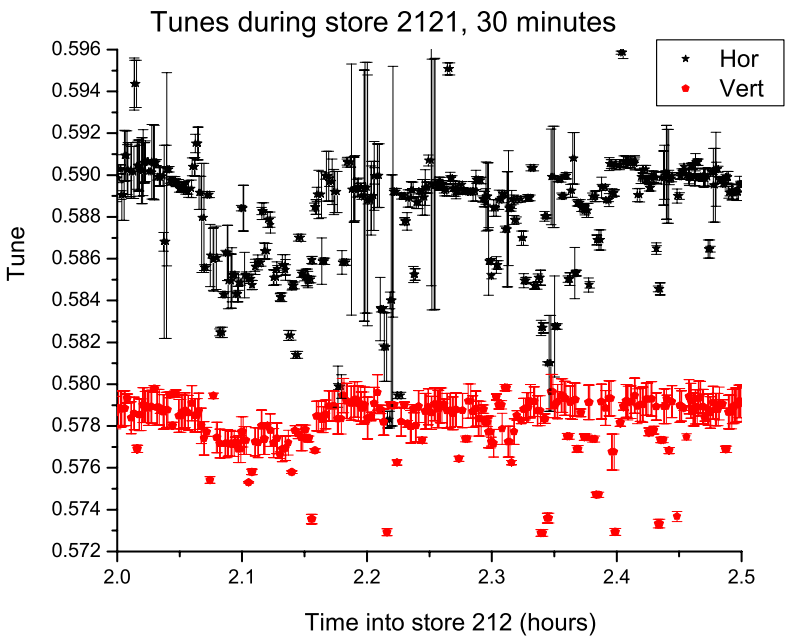
The line at 0.565 is even more troubling, because it is not quite outside the region of interest. It’s often a broader signal, and drifting! And, when it drifts into the betatron lines, it enhances the signal, further evidence for some kind of coupling with the beam.

The Ghost Line (s)…

- Yes, they sometimes drift right into the region of interest…
- In part, this is due to low signal level (only ~ 10 to 20 db above intrumental noise. If we can increase the betatron signal, this would not be such a problem.
- So far, we track only one such line, typically the one around 0.55 The fitting algorithm similar, ("smearing", Gaussian fit, followed by narrow fit). This strategy is not sufficient. An other solution must be found…Since such "noise" is coupled to the beam, somehow, it is not strictly of "instrumental" or "academical" interest, as it could potentially blow the emittance.

Tracking “some tunes” during the store…

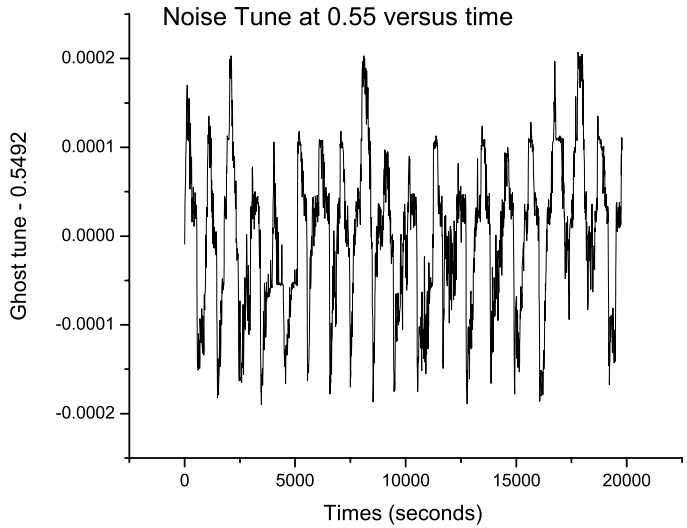


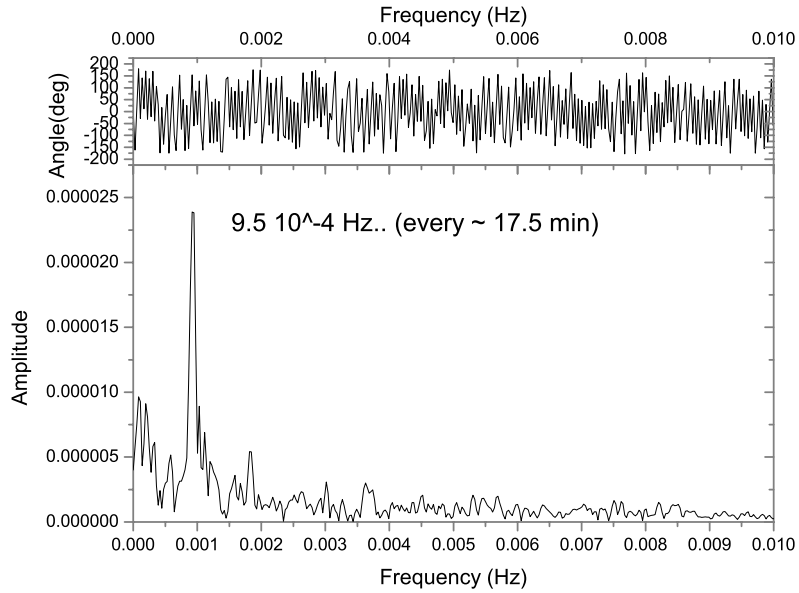


Note: when small errors are assigned, a 5 BW fit has successed, other wise, the error is 0.25 the width of the smeared Gaussians.

The tune fluctuations during the store are smaller than, or about the same as, the tune spread due to finite emittance/chromaticity. The effect of ghost line(s) could also explain such drifts, as while a ghost line goes through the betatron lines, it perturbs the measurements (and, quite possibly, the coherent signal from the beam itself) Correlation of such episodes with beam losses has been not (yet) been established. More work is needed…

Tracking the almost fixed noise line at 0.55…





This mysterious line at 0.55 (26.242 Khz, or an harmonic of this) is remarkably stable. It fluctuate in frequency tune space by ~ 10^-4, not too far from the spectrum Analyzer resolution, with an approximate period of 17.5 min.

The more tricky noise line at 0.565 deserve more attention. Further code needs to be written.

Coalesced Beam Studies Tune Tracking:

- Promising.. But we must:
- Clearly establish the coherent synchro-betatron signal by looking at the FFT phases. ➔ use a vector signal analyzer. Need hardware!
- Go back to the raw time-data, and split the 21 MHz (1.6 Ghz) signal into 72 "bunch by bunch", and do 72 FFT
  - A much bigger project, but if the Tev is indeed limited in great part by beam beam effect, why not invest into advanced D.A. and analysis software to track betraton tunes and chromaticity systematically ?
  - Also need instrumentation help!

Brief Technical Documentation

- *Hardware :*
- *Software :*
  - *The Java Side: ReadWaveForm*
  - *The root-based fitting package.*
    - *Main classes*
    - *Graphical Output*
    - *ASCII Ntuple output*
  - *This virtual front-end, seen from the ACNET world*
  - *The small GUI driving the implementation.*
    - *This one does not exist yet..*

Current Hardware

- *Hardware : We have only one scope (sorry, "box"), which could be connected to either X or Y planes. (see slide on ACNET for control).*
- *Reason for using a dedicated scope: we continuously read the device, preventing us to use the box as an interactive device: we can't change the marker position and do meaningful reading off the screen*
- *This "box" is a HP3561A Spectrum Analyser (FNAL # 54291) and is currently located in the North-East corner of MCR, below the old vsamcr tune display monitor.*
- *Takes its signal input from the same fan-out patch panel as the other Spectrum Analysers, used in setting the tunes.*
- *Setting; Using the file "Collider", characterized by a center frequency of 0.580 and a span of 0.065496 (in tune units). It takes 6 averages per reading. The vertical scale is from –90 db to –50 db, 5db per division, which is in fact completely irrelevant, the readout does not saturate for signal above -50 db*

The Java ReadWaveForm program

- *Goal : read a few ACNET devices to get the data, and compose a small straight ASCII file, to be read (asynchronously) by the TevTuneTracker RootFits package.*
- *ACNET devices being read in.*
  - *T:WFSA[], 400 element, 2 byte integer. The Wave Form, coming though the GPIB interface, converted to Ethernet, via a VxWork system.*
  - *T:ERING : the energy of the Tevatron, from the main dipole bus current. (we need this information to compute the expected synchro-betatron split)*
  - *T:IBEAM: if no current in the machine, no point burning CPU cycle to look for signals. Also, the line intensity could be related to the total beam intensity.*
  - *V:COALP: The M.I. Coalescing switch. Although not a guarantee that this state device reflects the Tev bunch configuration, it is the best indicator of what's injected (this device is used to control the next injection, therefore, someone could change it while beam is in TeV, thereby loosing the information about what's in the machine..).*
  - *C:CASE : The SDA Case number, which could be useful in later releases to adjust the algorithms based on running conditions.*

The Java ReadWaveForm program, II

- *More ACNET devices being read in:*
  - *T:MXA111 and T:MXA110 : the state of the switches controlling which signal (Horizontal or Vertical) the "box" receives. Note: this does not work yet.. To be debugged*
- *This java program is a standard Java Control/DAQ job. It "implements" "DaqJobCompletion" and "GenericCallback" interfaces.*
- *It also "extends" threads", and overwrites the same file over and over. Optionally, it can also write the same ASCII file under a unique name reflecting the state of the TeV and the time stamp. Such files can be used to "play back" some critical fits.*
- *Known Defect: A new job is created for each file (roughly every second or so), based on a "OnceImmediateEvent" event. A better implementation would be to specify the DaqJob based on "DeltaTimeEvent", to speed things up. The Callback for multiple device "Reading" and "JobCompletion" were a bit mysterious to us.*

The Root based C++ Fitter(s)

- *Overall organization: a bunch of loosely coupled classes, with multiple Main program to either run the code continuously re-reading the file created by ReadWaveForm, or from specific set of such files, or, conversely, coming from the T39 page (vsamcr data)*
- *Main Classes:*
  - *fitUncoalesced : for uncoalesced beams.*
  - *fitCoaleseced2 : for Coalesced beams, at any energies.*
  - *fitGhost1: Looking for the ghost line.*

    The classes have been written based a standalone fitter routine written by John Marraffino
- *Infrastructure Classes:*
  - *SpectrumAnalyser : a reader for the vsamcr, written by John Marraffino based on Dean Still code.*
  - *aFitResult: contain s and define the result of a fit.*
  - *CurrentFits: a container for the above results, to support "past history" knowledge.*
  - *eTevEnergy : for a given vsamcr data file and scan number, return TeV energy.*

The Main Programs

- *Read the input data file of interest and creates the raw histogram.*
- *The bin contents of this fixed bin size histogram are proportional to the "frequency probability" from the signal analyzer (scalar mode). Note: we convert here the log scale to a linear scale, such that, depending on condition, the noise signal is about 20 "units", and a strong signal is many tens of thousands of such units.*
- *Based on the state of the TeV, calls the ad-hoc fitter.*
- *For "offline analysis", performs 500 fits or only one, depending on Unix command line arguments (or small interactive dialog)*
- *For "online" (e.g., running for now on nova.fnal.gov):*
  - *Root graphical option can be turner off or on. (command line arguments)*
  - *If Energy of Tevatron < 100., wait for ~ 5 sec before reading the next file*
  - *If no beam, wait ~0.5 second.*

The fitUncoalesced class

- *First Smearing: Creates a Gaussian convoluted histogram (8 bins sigma), from the raw data.*
- *Fit with two Gaussians.. This defines the broad tunes in both planes, Vertical being always defined the lowest.*
    - *Seeded with either the history buffer (time extrapolated), if available, or, simply the default operating tunes (V=0.574, H = 0.584) . (currently hard-coded).*
  - *Note : the original implementation made multiple such convoluted 2-Gaussian fits, until the tune positions became stable under an increase smearing factor. For sake of expediency, this loop now runs only once, at 8 bins (or ~ .0013 in tune space).*
  - *The width of these Gaussian must less than .015 in tune space.*
  - *No cuts on tune position, nor amplitude. (this is probably an oversight, are should be revisited.)*
  - *No cuts on chi-square either. (a good thing, because we do not have a complete model of what this smeared histogram mustlook like, in details.)*
  - *If such fits are bad, not result is reported.*

The fitUncoalesced class, II

- *Once the "broad tunes" are found via this smeared histogram, narrow Synchro-betatron lines are searched for (narrow fits) :*
  - *Compute the expected (predicted) synchro-betatron line tune split ($\Delta_{sb)}$ based on the Tevatron energy, ranging from ~ .0016 to 0.00063*
  - *Each X, Y planes are treated independently from each other, even if the tunes are close to each others. Two distinct histograms are created, from the initial raw histogram, centered on the broad tune with a span = +- 2.5 times Δ_{sb}*
  - *Smooth these histogram, using the Root "Smooth" command, itself based on the old CERN Smoothing algorithm (# 355QH, presented by J. Friedman in the proceedings of the 1974 CERN school for computing, Norway, 11-24 August, 1974, Norway.) The number of consecutive smoothing is set to 3 (negotiable).*
  - *Fit these smoothed histograms with 5 Breit-Wigners surimposed on a flat background. The spacing between these lines, $\Delta_{sb,}$ is kept constant and is fitted for. The width of these lines are also a unique fitted quantity. Only the relative amplitudes are allowed to vary, and are fitted for. Such fit has therefore 9 parameters.*
  - *Validity cuts on such Δ_{sb} confirmed tunes:*
    - *The centroid*

The fitUncoalesced class, III

- *Narrow fits, cont'd:*
  - *Validity cuts on such Δ_{sb} confirmed tunes:*
    - *The centroids must be within 0.5 * width of broad Gaussian of the broad tune.*
    - *Significant relative amplitude with respect to background (> 5) (Negotiable, could be tuned better), for the central line and at least one of the satellite lines.*
    - *The fiited value for Δ_{sb} must be within 30% of the predicted value.*
    - *Again, no cuts on chi-square.*
- *If such a fit is successful, the final tune is set by the line position which is closest to the broad tune. Else, the final tune is set to the broad tune.*
- *Whether or not the narrow fits are successful, the best available value for the tune is send to ACNET. The error on the tune is either the width of the broad smeared Gaussian, or ¼ of the width of the narrow Breit-Wigner.*
- *The tune width variable is either set to $\Delta_{sb,}$ with a relative error of 0.25, or if the narrow fit is not successful, to the broad width with a relative error of also 0.25.*

The fitCoalesced2 class

- *First Smearing: Creates a set of 3 Gaussian-convoluted histogram, with 7.5, 10 ad 12.5 bins sigma.*
- *Fit each such histogram with 3 Gaussians.. The broad tunes are defined as the average of these Gaussian position over the set of the 3 convoluted histograms.*
- *As for the uncoalesced case, Vertical is always defined as the lowest tune, Horizontal is the middle one (The thought here that the highest tune is the horizontal pbar tune. Quite often, 2 out of the 3 Gaussians overlap greatly, as the pbar tune is invisible on the proton channel. Or, worse, the third Gaussian is the "ghost-line", in which case this analysis is most likely compromised.)*
  - *Seeded with either the history buffer (time extrapolated), if available, or, simply the default operating tunes (V=0.574, H = 0.584) . (currently hard-coded).*
  - *The only validity cuts on these broadtune fits is on the width of these Gaussian: less than .05 in tune space, for the 10 bins Gaussian convolution.*
  - *If such fits are bad, not result is reported.*

The fitCoalesced2 class, II

- *As for the uncoalesced case, we now search for narrow Δ_{sb} lines in the surrounding of the broad tune values. The difference is such tunes are now much broader. Thus, for each x and Y plane, (currently, the highest tune is ignored in this phase. Could be revisited..), we split these tunes X or Y "regions" into "zones", each zone covering a span of +- 2.5 times Δ_{sb}.*
- *Each zone, for each plane, are fitted independently from each other, with the 5 Breit-Wigner function. The shape of the background is no long constant, it follows the 3 Gaussian (based on the 10 bin convolution). Only it's amplitude is allowed to vary. The width the central line is also allowed to be different from it's satellite. This 5 B.W. fit has therefore 10 parameters.*
- *As in the uncoalesced case, the "tune value" is given by the line which is closest to the broad tune.*
  - *Details: the tune value for a given zone must be within that zone and zones can not extend beyond the span of the original histograms. Other self-consistency checks are applied.*

Note on Errors…

- *Standard Root/Minuit interface assume that the error on a bin content comes from Poisson statistics (for large N, sqrt(n)). Clearly wrong in our case.*
- *Various raw bin content error assignment have been tried:*
  - *Csnt (from noise in the absence of beam)*
  - *Cnst + (few%) * signal*
- *The stability of the fits does not depend critically on such an issue, provide we do not attempt to "pin-down" the flat background too hard, because the low level signal has a complicated shape (sometimes, negative interference occur, if so, there are too many line with too complicated shapes to follow..)*
- *However, the errors of the fitted tunes are not those reported by the fitting package. Instead, the tune position error is typically 0.25 * the width of the line, or if, a 5 BW fit succeeds, than it is 0.25 of the Δsb fitted quantity.*

Note on fit Seeding (initial parameter for minimization package.)

- *Critical to the success of the fits, even with the "smearing" action.*
- *For the multi-Gaussian fits, two disctinct strategies are used:*
  - *"Bump hunting" : look for extremum in the histogram, and order such bumps by amplitudes…*
  - *Use "previous history", via the use of the circular buffer. (each fitting class can be equipped wich such a buffer, which memorize past success.. Only confirmed 5 BW fits enter in this buffer. These buffer can extrapolate in the future, tentatively predict where the signal is.*
- *The 5 BW fits are always seeded by the Gaussian fits.*

The currentFits class.

- *Goal: support the history buffers for all core fitting classes.*
- *Based on a C++ STL queue, as a container. Used as a circular buffer, which an adjustable size. The queue contains pointers to aResultFits class.*
- *Basic methods :*
  - *Add a fit to the circular queue*
  - *Compute the average of tunes for a given plane.*
  - *Compute the anticipated value for a given aResultFit parameter, based on the time-stamped fit values, based on parabolic fit/numerical extrapolation.*

Possible Software Upgrades & Issues

- *A new minimization package, with more robust andmodern algorithms is being written by Mark Fischler, Dave Sachs and other. We could/should upgrade to it when available.*
- *Interactive use of this package: Checking the validity of the fits by looking at the debugging plots created by this Root application. Moreover, it would be nice to be able to control the history buffer for critical fitters, or simply have a manual mode in which each scan can be frozen while the user looks at it.*
- *Better organization of the code.. ? .. Open to the idea of extensize (Zoom-like) review..*
- *If deployed in a formal way, we need to consolidate the D.A. part, and secure the adhoc Root/C++ support in CD and/or BD .*

ACNET Devices

- T:MXA111 and T:MXA110 : Used to connect the HP3561 spectrum analyser to either the horizontal, or vertical strip line signal: If
  - T:MXA111=1, T:MXA110 =0, ➔ connected to vertical strips
  - T:MXA111=0, T:MXA110 =1, ➔ connected to horizontal strips
  - Other: unknown results.
- T:Tuu*v*s*s*: The tune data, generated by the C++/Root tune fitter program. Values are either coming from the broad tune fits, or the search for narrow synchro-betatron lines. These are array devices, of length 2, 32 bit floating point numbers.
  - First element : the value :
  - 2nd : the error estimated from fit results.
  - u* : either X or Y : the strip line to which the signal Analyser/ fitter is connected to
  - V* : either X or Y: refers to Horizontal or Vertical tune, respectively.

ACNET Devices, II

- T:TUu*v*BR: The tune values. Best available value returned by the fitter.
- T:TUu*v*WD : The tune width. Either the ($\Delta_{sb)}$ fitted values or the (if the search for such narrow lines was successful), or conversely, the broad tune width.
- T:TUu*u*GH: The position of the ghost line, for the X or Y signal.
- More information could be transferred…

Conclusions

- It is possible to express a measurement method into an algorithm that runs on a computer, as fast as one can read from a screen.
- Having the capability of tracking uncoalesced tunes ought to be good, and could be deployed. Required if we need feedback, itself required is we want to shorten shot setup!
- Coalesced beam need further study:
  - A bit of tickle without blowing the emittance ?
  - Noise line
  - Bunch by bunch
- From prototype to production version: a collaborative effort!

Acknowledgments

- Many thanks to member of the Tevatron dept, especially Dean Still and Jim Steimel.
- This project would not have been possible without the help of Beams Control. Thanks to Jim Patrick, Charlie Briegel, Ron Rechenmaker and others!