Improving Beam Stability in the APS Booster Synchrotron

Chang Xu[1],* Joseph Calvey[2], Katherine Harkay[2], and Chih-Yuan Yao[2]
[1]*IIT Department of Physics,* [2]*APS Argonne National Laboratory*
(Dated: August 29, 2018)

The tune instability in booster synchrotron has been a problem since the start of the APS. The designed tune in x and y directions are respectively 11.75 and 9.8. The tune instability affects the emittance of the beam as well as the transport efficiency. Thus, it is critical to make the tunes as constant as possible throughout the acceleration in booster. Simulations of the beam behavior were performed, and a few experiments were done to verify the result. Several options for mitigating the tune swing were also investigated.

The transverse tune of a beam is defined to be the oscillation frequency divided by revolution frequency, it is strongly related to the structure of the accelerator lattice and magnet strengths in the accelerator. Improper tune could cause beam instability along the direction of oscillation when accelerated, In this paper the transverse tunes of the APS booster are studied in order to get better beam stability.

Both the horizontal tune and the vertical tune have an unexpected swing as the beam is accelerated in the booster, while they were designed to be constant. `Elegant`[1] was used to simulate the beam behavior, and appropriate parameters of the magnets were found to match the simulation with the measurements. On top of the simulation, a few ideas were applied and tried out to straighten the tunes and stabilize the beam.

I. CONTROLLING THE BEAM

The beam is controlled by dipole, quadrupole, and sextupole magnets. Since we assume the beam is accelerated linearly in the booster, the magnets should be linear over time as well, for dipole as an example,

$$(B\rho) = \frac{p}{q} = \frac{\beta E}{cq} \tag{1}$$

where B is the magnetic field, ρ is the bending radius of the accelerator, c is the speed of light, β is the velocity relative to the speed of light, q is the electron charge, and E is the energy. Using Ampere's Law,

$$\oint \vec{B}_y \cdot d\vec{l} = \mu_0 I \qquad B_y \approx BMSlope * I \tag{2}$$

magnetic field is proportional to the current, and `BMSlope` comes from measurement, and we control the magnets by linear current reference ramp. Similar linearities are true for the other multipole magnets as well[2]. As a result, we control the magnets by linear current reference ramp, as shown in FIG. 1.

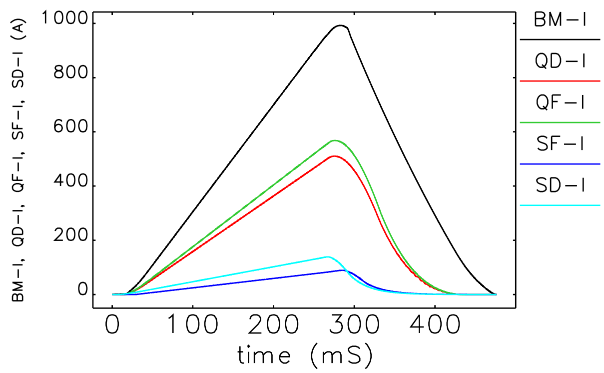

Figure 1. Reference ramp used to control different magnets. BM is "bending magnet", QD is "Quadrupole focus", QF is "quadrupole defocus", SF is "sextupole focus", and SD is "sextupole defocus".

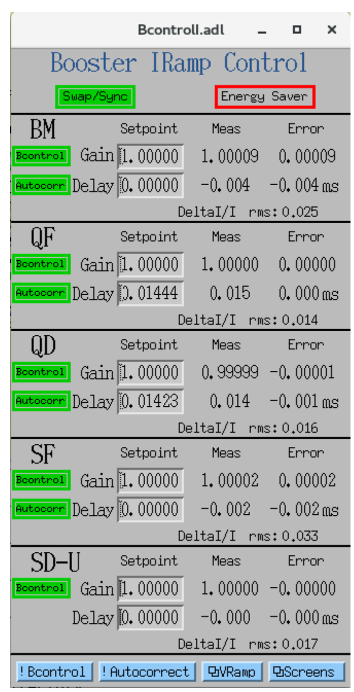

Figure 2. Screenshot of the control panel.

* cxu30@hawk.iit.edu

The injection of the beam is 35.65ms and it gets accelerated for 222.814ms, which is the ramp time, using the linear ascending part of the ramp.

As shown in FIG. 2, the gains and delays are used in a control panel[3] to control the ramp applied to the magnets. Gain is multiplied on top of the reference ramp which changes the slope, while delay is the offset of the reference ramp with respect to the injection time, for different magnets individually.

II. CURRENT TUNES AND CONCERNS

The tunes are measured by FFT every 1.5ms of accelerating in the booster[4], and only the fractional part is shown in the plots.

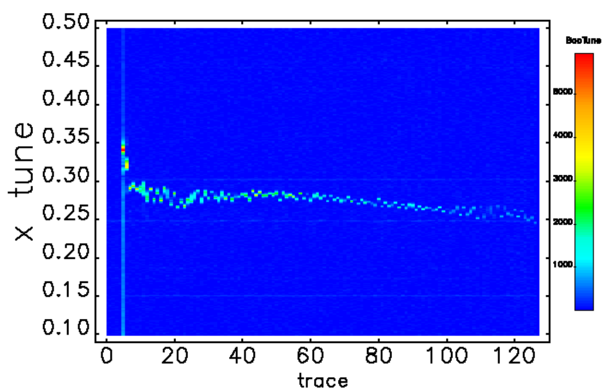

Figure 3. Measured tune vs. energy for x direction.

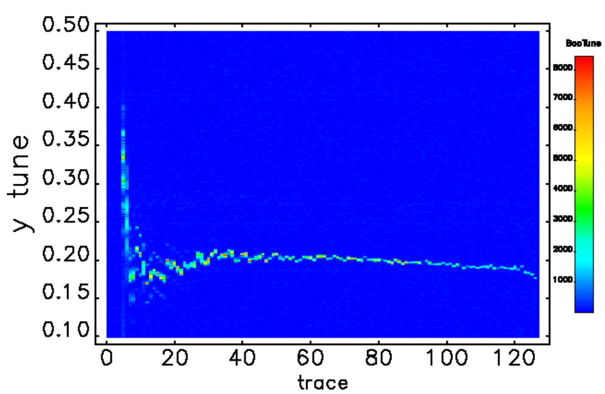

Figure 4. Measured tune vs. energy for y direction.

The raw data(FIG. 3. and FIG. 4.) are processed to FIG. 5. and FIG. 6. for better visualization.

The swings appearing at the lower energy part is undesired, and could cause beam loss if it crosses integer or half integer points. The purpose of this project is to attempt to better understand and control the booster tunes.

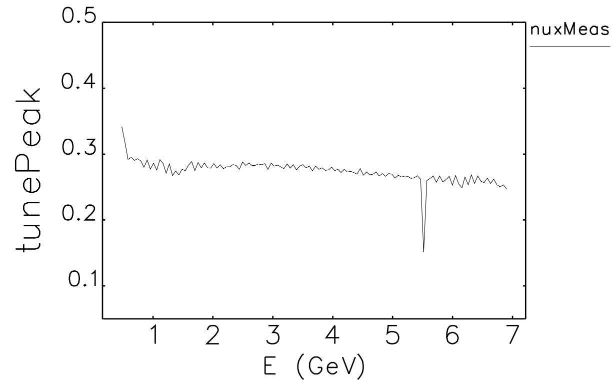

Figure 5. Measured tune peak vs. energy for x direction. The twitch at around 5.5GeV is error from data processing and can be ignored.

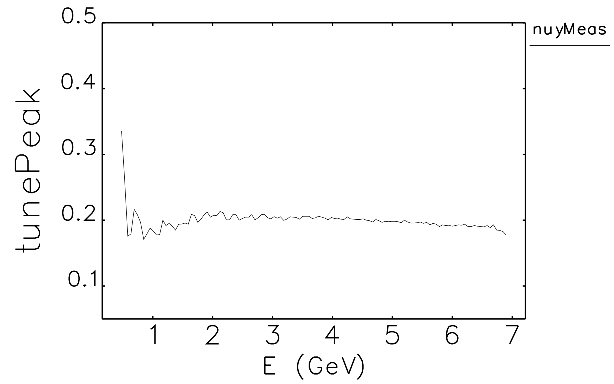

Figure 6. Measured tune peak vs. energy for y direction.

III. SIMULATION

Particle tracking simulation of the beam was done in `elegant` with 125 points interpolated from the reference ramp files. The number of 125 was chosen because of the 125 valid data points from tune measurements. The agreement of the simulation result with respect to the experiments is evaluated from the same number of data points.

When simulating, some parameters have to be corrected based on facts and calculations because of the general difference between coding and real machine. The method of finding the most influential parameters(injection time, ramp time, QD gain, QD delay, QF gain, and QF delay) on tunes has been developed and operated. A few important parameters are omitted (SD gain, SD delay, SF gain, and SF delay) due to the lack of time of this project, but their influences on the simulation was inspected and discussed. The constants used from Booster Parameters[5] operation page are listed below.

Table I. Parameter used in simulation from operation page.

General parameters	Notations	Values	Units
Injection energy	E_i	0.425	GeV
Extraction energy	E_e	7	GeV
Horizontal tune	ν_x	11.75	$\frac{1}{2\pi}$
Vertical tune	ν_y	9.8	$\frac{1}{2\pi}$
Ramp time	t_{ramp}	222.814	ms
slope of I vs. B_{BM}	$BMSlope$	0.0006887	T/A

III.1. Dipole parameters

First of all, to match the injection energy with the correct magnetic field strength, we calculate the correct injection time using equation (1) and (2) as follows with injection energy E_{inj},

$$\begin{aligned} B_{inj} &= \frac{\beta E_{inj}}{\rho c q} \\ &= \frac{1 * 0.425 * 10^9}{33.3009 * 3 * 10^8} T \\ &= 0.0425T \end{aligned} \tag{3}$$

$$\begin{aligned} I_{inj} &= \frac{B_{inj}}{BMSlope} \\ &= \frac{0.0425}{0.0006887} A \\ &= 56.56A \end{aligned} \tag{4}$$

and then from the reference ramp file(FIG. 1) with the current at injection time, $I_{inj} = 56.56A$, the correct injection time can be found to be around $t_{inj} = 36.55ms$. Thus, extraction time is found by $t_{ext} = t_{inj} + t_{ramp} = 259.36ms$. Reading from the same ramp file, we get $I_{ext} = 932.55A$ at $t_{ext} = 259.36ms$. However, using $BMSlope$ we can calculate the same I_{ext} for bending the beam correctly, using the extraction energy E_{ext},

$$\begin{aligned} B_{ext} &= \frac{\beta E_{ext}}{\rho c q} \\ &= \frac{1 * 7 * 10^9}{33.3009 * 3 * 10^8} T \\ &= 0.701T \end{aligned} \tag{5}$$

$$\begin{aligned} I_{ext} &= \frac{B_{ext}}{BMSlope} \\ &= \frac{0.701}{0.0006887} A \\ &= 1017.86A \end{aligned} \tag{6}$$

which is far off from the value in the reference ramp file ($I_{ext} = 932.55A$) and the magnetic field would not be strong enough to bend the beam around at corresponding energy.

In order to force the dipole magnetic field in simulation to be strong enough while not changing the reference ramp, we use a "fudge factor" to give B_y vs. I a bigger slope. So instead of using equation (2), we use,

$$B_y = fudgeFactor * (BMSlope * I + BMOffset) \tag{7}$$

Where the fudge factor is calculated by,

$$\begin{aligned} fudgeFactor * BMSlope &= \frac{B_{ext} - B_{inj}}{I_{ext} - I_{inj}} \\ fudgeFactor &= \frac{B_{ext} - B_{inj}}{BMSlope * (I_{ext} - I_{inj})} \\ &= \frac{0.701 - 0.0425}{0.0006887 * (932.55 - 56.56)} \\ &= 1.09 \end{aligned} \tag{8}$$

Since the slope is boosted, an offset `BMOffset` would appear in the equation. To simplify this case, we set `BMOffset` to be zero and modify injection time t_{inj} again equivalently instead to finally get the correct parameters for dipole. Plugging $B_{inj} = 0.0425T$ back into the solved equation (7), we can find the new $I_{inj} = 56.55$, and corresponding $t_{inj} = 36.59$ from reading the reference ramp file.

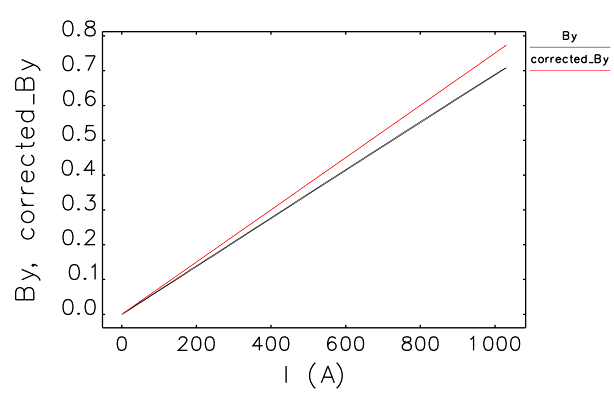

Figure 7. Measured and corrected By(T) vs. I(A).

III.2. Quadrupole parameters

With the parameters proved above, we are able to run `elegant` successfully, but not with a good result. In order to get a good agreement with measurements, four parameters(QD gain, QD shift, QF gain, and QF shift) of the quadrupole magnets have to be corrected. Since all four of them affect both x tune and y tune, a script was written to try and discover the right parameters, taking advantage of the very different roles these parameters play on the tune plots.

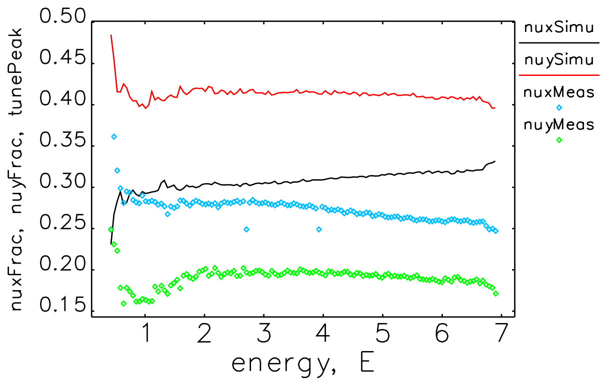

Figure 8. Simulation with no corrections on quadrupole parameters. Lines are simulations while dots are from measurements.

From inspections, QD gain changes mostly the offset of ν_y, but has a small effect on ν_x offset. It barely changes the shapes of the tunes. QD shift mostly tilts the lower energy tail, but has small effects on ν_y offset, ν_x offset, and lower energy tail of ν_x. It barely changes the shapes of the higher energy tail.

Similarly for QF gain and QF shift, swapping x and y.

By analyzing the effects of those quadrupole parameters, we are able to reflect their impacts directly onto four important individual features of the tune plots, that decide how close the simulation results and measurements are. QD gain decides the final value of ν_y, ν_{y_f}, and provided that ν_{y_f} is fixed, QD shift would directly affect how good the agreement is, which is measured by the sum of squared errors SSE_y for ν_y,

$$SSE = \sum_i [(\frac{energy_{meas} - energy_{simu}}{energy_{meas}})^2 + (\frac{\nu_{meas} - \nu_{simu}}{\nu_{meas}})^2] \quad (9)$$

which is also Similar for QF gain and QF shift.

A simple summary of the algorithm is shown below as TABLE II.

Following the steps, the parameters are found one by one in order dynamically. Baby steps are taken each time changing a parameter, and for every step it takes, it does not care for the parameters with "?", but it checks all the parameters with "✓"(where the quadrupole parameters result in correct tune features) and see if the new parameter it is trying out has gotten the old, already "checked" values, too far away from the measurements. If so, it would pause exploring new parameters and focus on fixing the old ones first. It moves on to the next state if and only if the current parameter reaches certain level of agreement.

Table II. Showing the states of parameters. "?" means the program hasn't cared about this yet, "(1)" refers to changing QDGain to move ν_{y_f} towards 9.8; "(2)" means trying increasing and decreasing QDShift and pick the one with the best SSE_y; "(3)" refers to changing QFGain to move ν_{x_f} towards 11.75; and "(4)" means trying increasing and decreasing QFShift and pick the one with the best SSE_x. And "✓" means the program successfully finds the correct quadrupole parameter to obtain the corresponding tune parameter at this state.

States	QDGain (ν_{y_f})	QDShift (SSE_y)	QFGain (ν_{x_f})	QFShift (SSE_x)
1.	(1)	?	?	?
2.	✓	(2)	?	?
3.	✓	✓	(3)	?
4.	✓	✓	✓	(4)
5.	✓	✓	✓	✓

The final result of the simulation is shown in FIG. 9. The simulation curve follows the measured data points very well, even at lower energy tail. The SSE values for FIG. 9 are $SSE_x = 0.828$ and $SSE_y = 0.315$, according to equation (9).

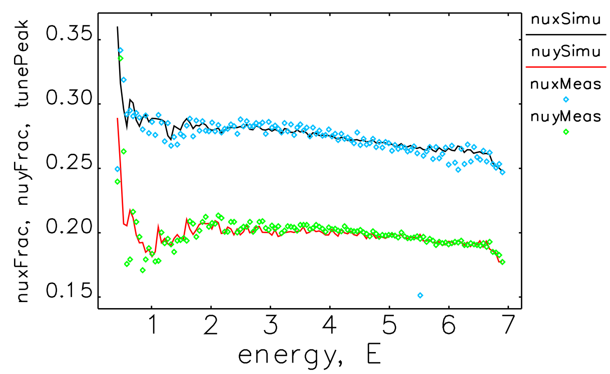

Figure 9. Final simulation result with corrected quadrupole parameters

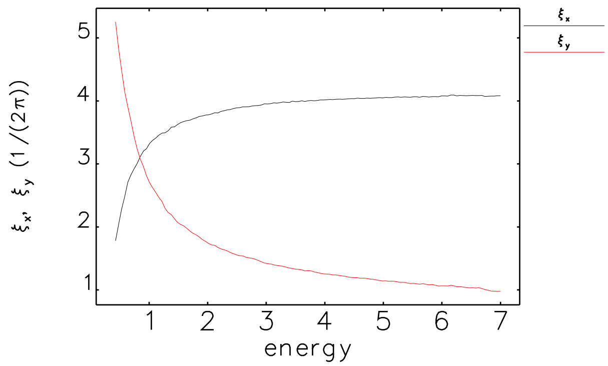

Figure 10. Chromaticity vs. energy for the final simulation result.

III.3. Sextupole parameters

To make the simulation complete, similarly to quadrupole parameters, sextupole parameters have to be corrected as well. The sextupole parameters can be determined by using a similar algorithm based on chromaticity vs. energy plots. The chromaticity was not measured in the experiments, so unfortunately the sextupole parameters were not corrected in the simulation.

III.4. Eddy Current Effect[2]

Since we are varying current in dipole, eddy current effect has to be considered. Old calculations were used for the sextupole constant in dipole field due to this effect, and the calculation was not completely verified because of the lack of time in this project.

III.5. Results

With corrected dipole parameters and quadrupole parameters, but not sextupole ones, we tested how well the result fits the measurements upon changing of the quadrupole parameters on top of the results. The agreement is pretty good and a few examples are shown below(FIG. 11 - 13), the small gap might come from the incorrect sextupole parameters sine they were left out as default.

IV. ATTEMPTS FOR BETTER BEAM STABILITY

With a good simulation, we tried to manipulate the tunes in different ways to get a better beam stability.

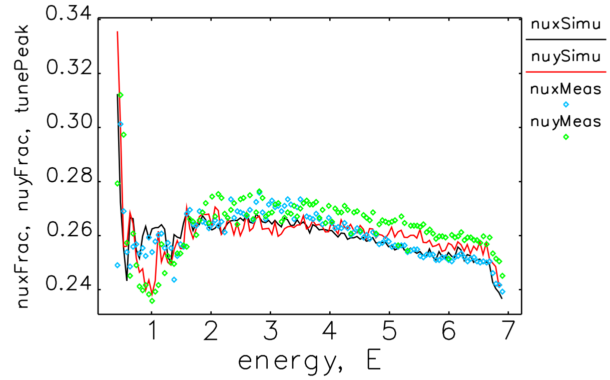

Figure 11. Data agreement for QDGain = 0.996.

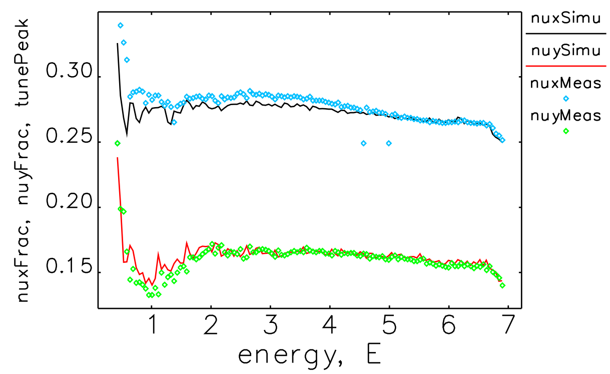

Figure 12. Data agreement for QDGain = 1.002.

IV.1. overlapping tunes for coupling

We have an assumption that when ν_x and ν_y are overlapped, we can control the coupling of the two tunes. With the setting of FIG. 11, we measured the

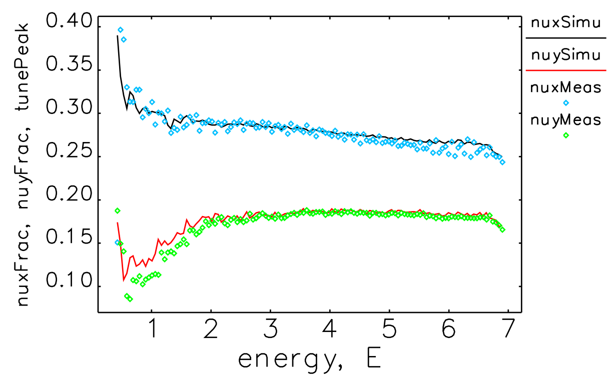

Figure 13. Data agreement for QDShift = 0.104.

beam profile and the coupling did appear. The profile is shorter in x direction when the tunes are coupled(FIG. 14) than separate(FIG. 15). This proves the feasibility of reducing x emittance by controlling the tune overlaps.

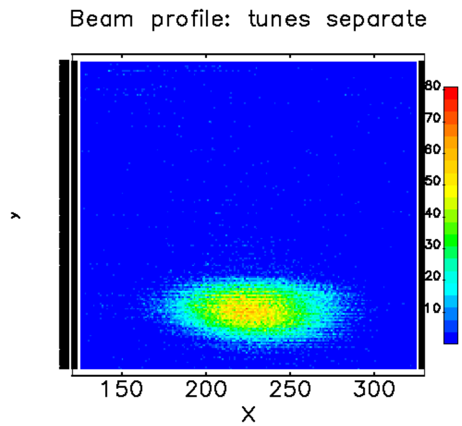

Figure 14. Beam profile for tunes separated.

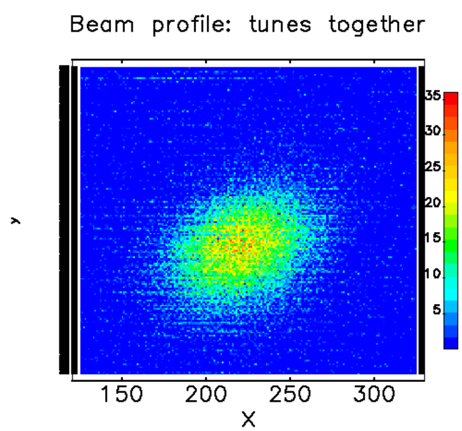

Figure 15. Beam profile for tunes together.

IV.2. Tilting Higher Energy Tail for Coupling

By changing ν_y from 9.8 to around 9.25, we can make the two tunes go towards each other at higher energy part(FIG. 16) This might be a better idea of controlling the coupling since the lower energy parts are separate. This possibility was only shown in simulation and the testing at the booster needs further inspections.

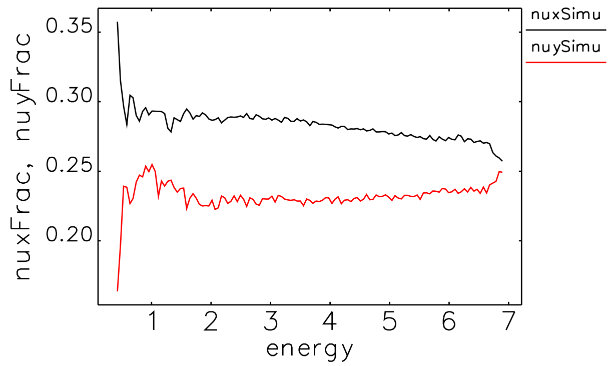

Figure 16. Simulation of tunes when changing ν_y from around 9.8 to around 9.25.

IV.3. Using Bumps to Smooth Out Tunes

Manipulating the reference ramp file could lead to changes at tune vs. energy plot. By adding small bumps(FIG. 17 - 19) onto the ramps, we can manipulate the ramps to try to cancel out the swings in tunes. The tests are only done in simulations.

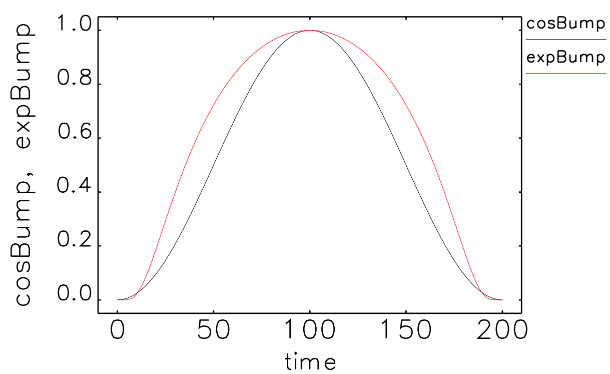

Figure 17. Sample bump files available to add/multiply on the reference ramp files.

Focusing on ν_y, we can straighten higher energy part(FIG. 20) or even tilt it up(FIG.21) to have a similar effect as decreasing ν_y to make the tunes go towards each other.

But for the lower energy part, as different bumps at different regions are applied to reduce the swing, a lot of zig-zags would appear in that region. Although the amplitude of the swing can be decreased by carefully applying multiple bumps, the zig-zags won't disappear, in fact, they could get worse. We suspect those zig-zags come from the linear ramp for dipole being not ideally straight, but need more evidence and more attempts on smoothing out the dipole reference ramp.

We can squeeze the ν_y to have a swing with smaller

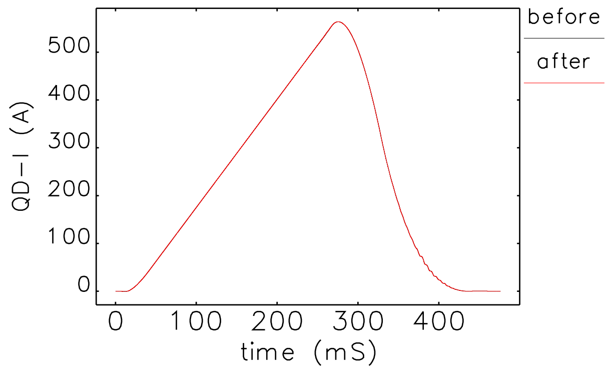

Figure 18. QD-I ramp file before and after applied a bump.

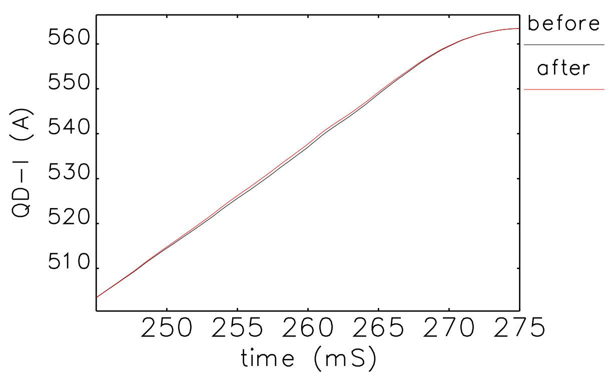

Figure 19. QD-I ramp file before and after applied a bump, zoomed in on the bumped region.

amplitude, but the zig-zags get worse.

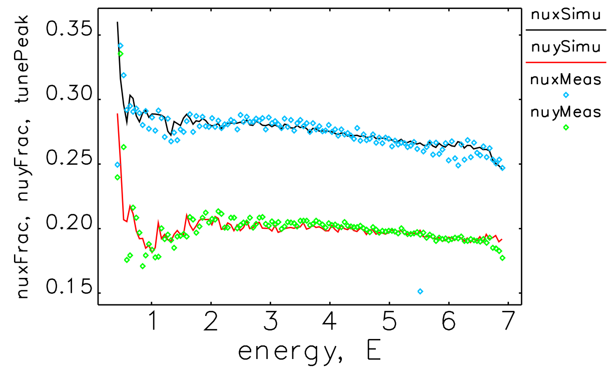

Figure 20. Flattening the higher energy tail using bump method.

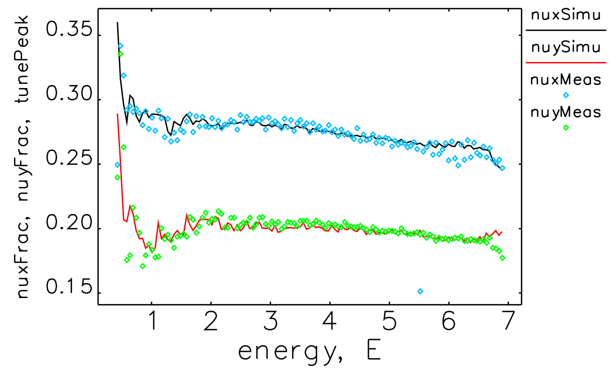

Figure 21. Tilting the higher energy tail up towards x tune using bump method.

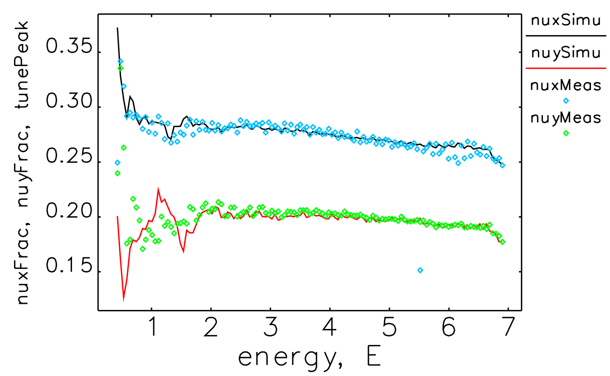

Figure 22. Smoothing out the lower energy tail using bump method.

V. CONCLUSIONS

A method of simulating the beam behavior with correct tune plots has been developed. A few things are yet to be done, including correcting sextupole parameters and verifying eddy current effect calculation. Machine learning[6] could also be a good method of finding the correct parameters to look into in the future.

A few ideas of making the beam more stable through tunes are proposed. Overlapping tunes and the bump method are proved to be useful, but still need future work. The swing at lower energy tail is suspected to be from non-linear part of dipole reference ramp, which could be fixed by going through negative current to calibrate the hysteresis effect on the By vs. I curve. Due to the inconvenience of our power supply for the dipole, this idea could be tested on the simulation first as a proof of principle. The zig-zags at low energy tail might be caused by the noise in the dipole reference ramp, and smoothing it out might help.

ACKNOWLEDGMENTS

I gratefully acknowledge Dr. Joseph Calvey, Dr. Katherine Harkay, and Dr. Chih-Yuan Yao for their helpful advice and insights on this project.

[1] M. Borland, Technical Report **LS-287**, 225 (2000).
[2] N. Sereno and S. Kim, Technical Report **LS-302** (2003).
[3] S. X. C. Yao, H. Shang and G. Fystro, Technical Report **OAG-TN-2014-033** (2014).
[4] C. Yao and H. Shang, In Proceedings of 2012 Beam Instrumentation Workshop **LS-287** (2012).
[5] N. S. Sereno, "Booster parameters," (2006).
[6] R. dubcakova, *Genetic Programming and Evolvable Machines*, (2010).