# ILC Simulations

Rebecca Ramjiawan

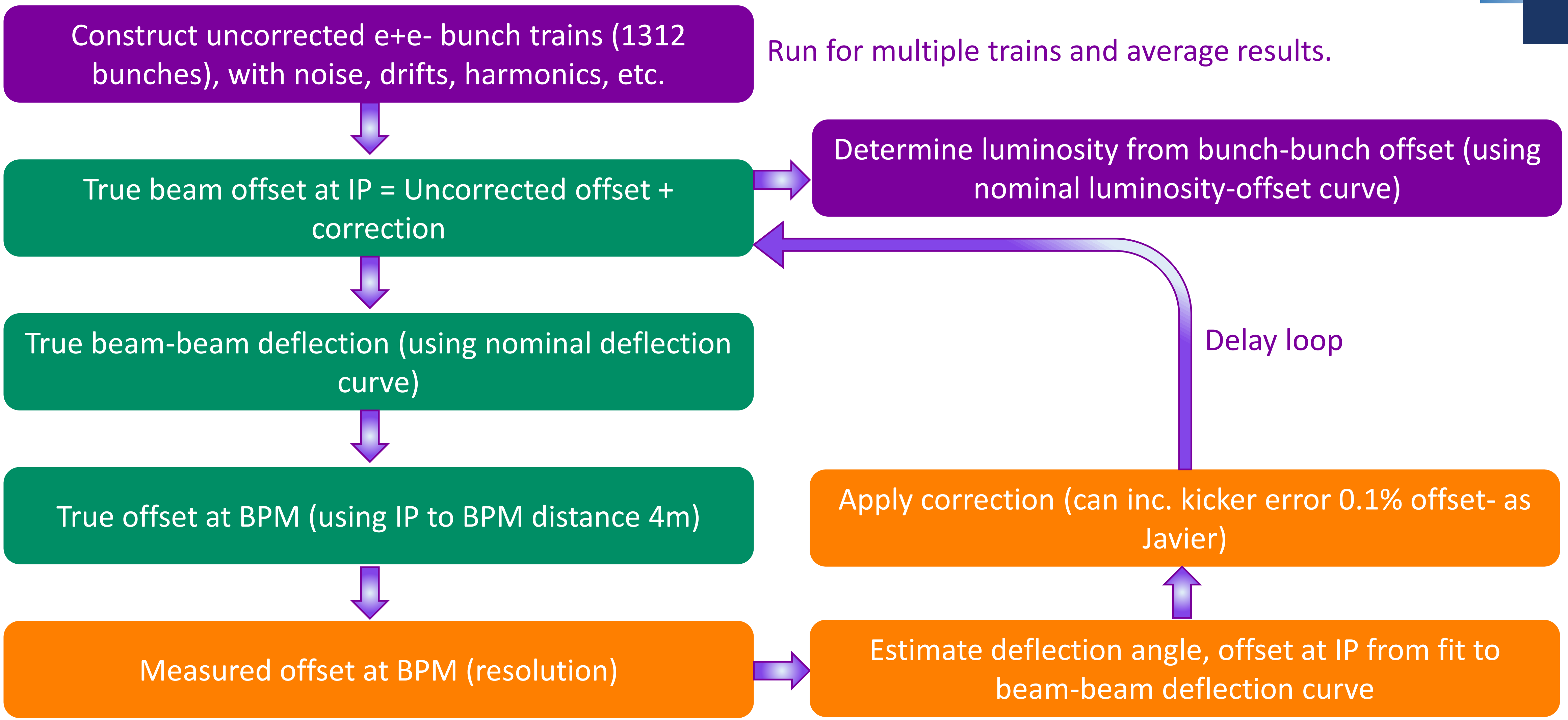
FONT meeting

*Friday, 10th August 2018*

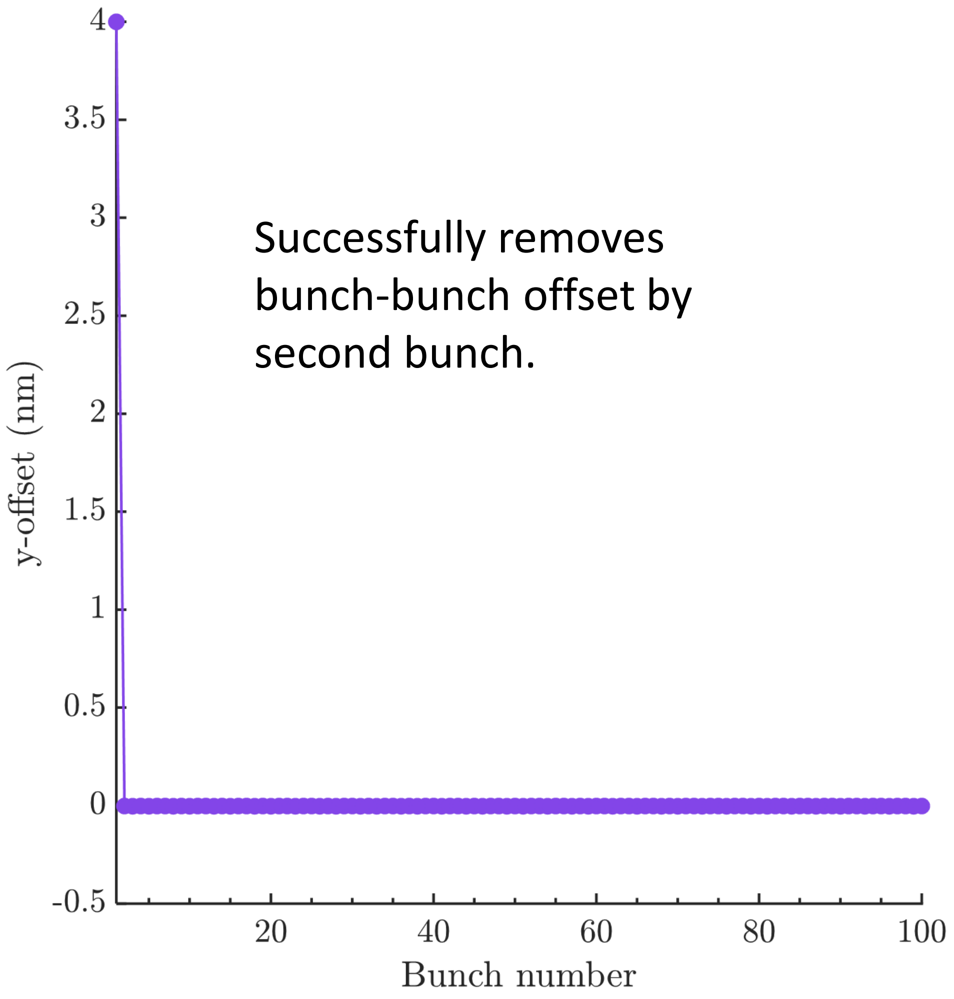
# Outline

- Basic model of FB system implemented in MATLAB, generate bunch trains and model BPM, kicker and lever-arm effect.
- MATLAB model 1312 bunch trains: bunch train structures with constant offsets, offsets between consecutive bunches, harmonics of a range of frequencies.
- MATLAB model FB system: BPM with resolution effects, kicker with kicker noise, proportional control, proportional and integral control, averaging over multiple bunches, weighted averaging over two bunches ($bunch_i$ and $bunch_{i-1}$).
- Lucretia: generate 1312 bunch train and track through the BDS to the IP, model bunch-bunch interaction at the IP using Guinea-Pig.
- Deflection angle curves modelled for various beam parameters showing dependence on: bunch size (x,y), bunch length, charge.

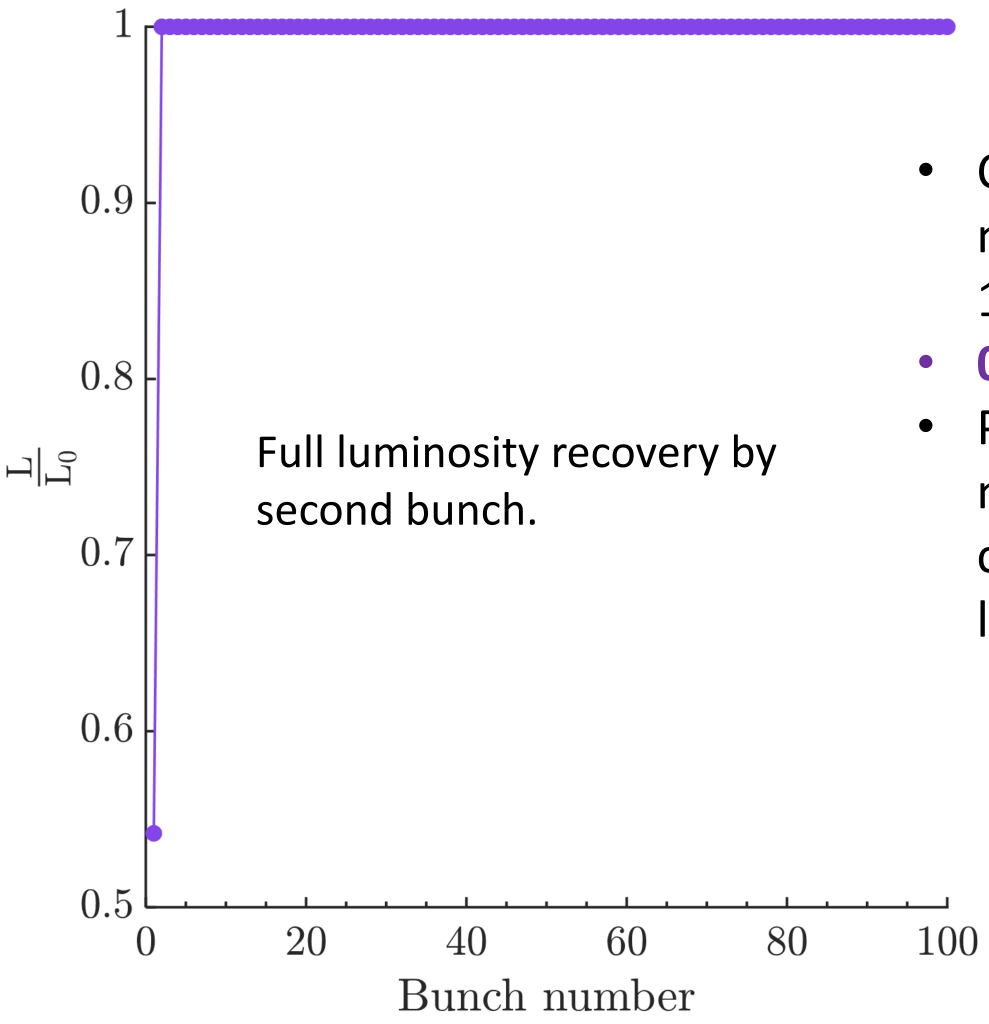
# Feedback Loop Simulation (MATLAB)



# Ideal Situation – Rigid Beam



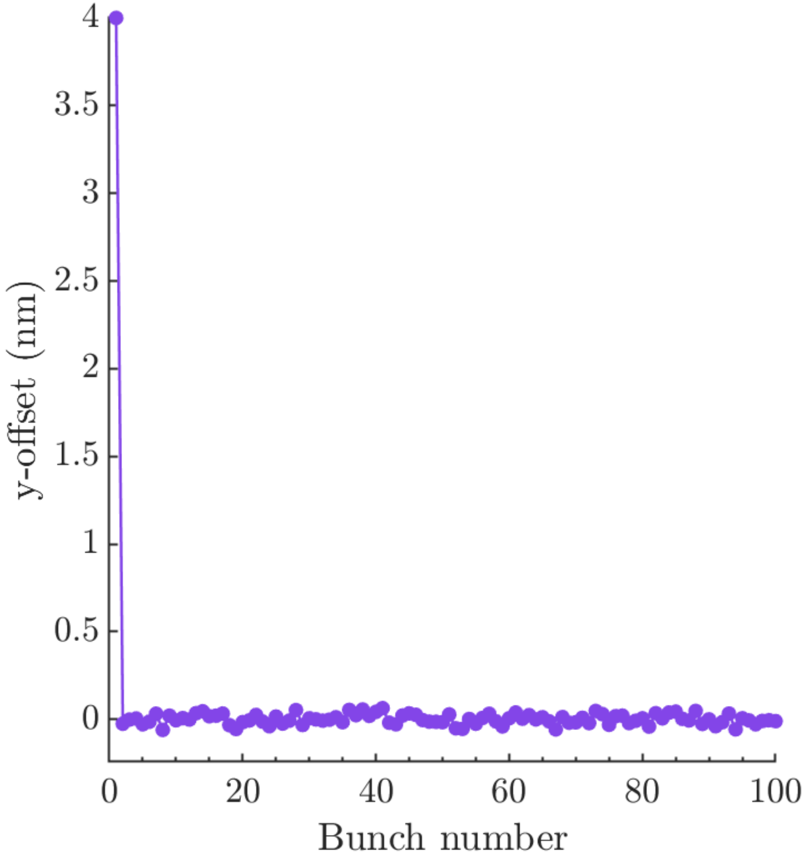

Successfully removes bunch-bunch offset by second bunch.

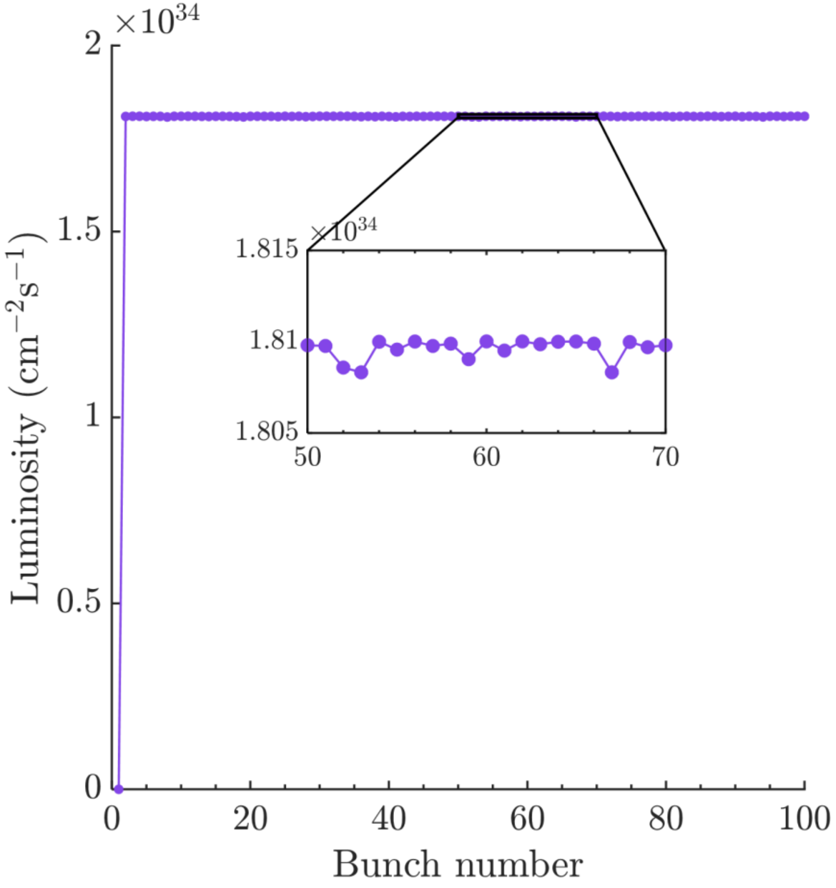



Full luminosity recovery by second bunch.

- Constant offset 70 nm throughout 1312 bunch train.
- **0 um resolution.**
- Perfect correction maintained correctly with delay loop ✓

# Resolution Effects



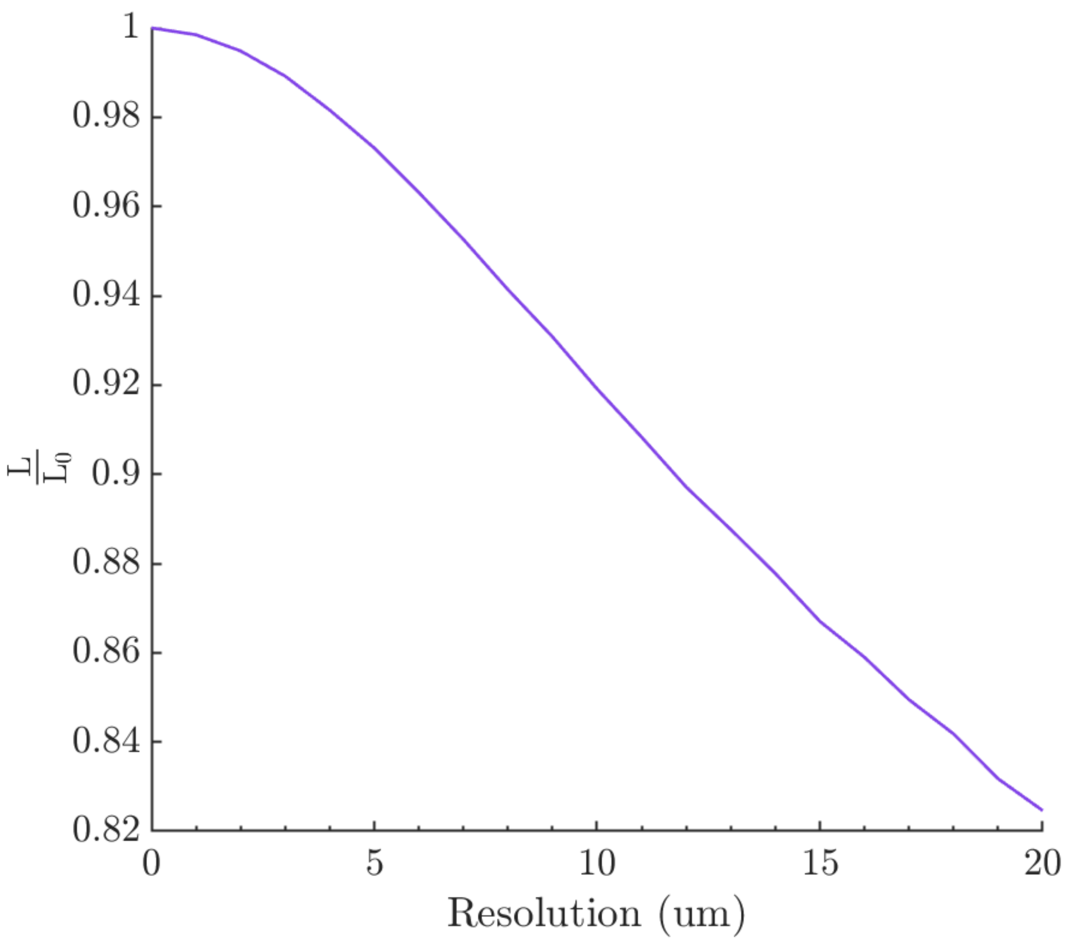

Rigid bunch trains, with 4 nm offset, with feedback with BPM resolution 1 um operating.

# Luminosity vs. Resolution

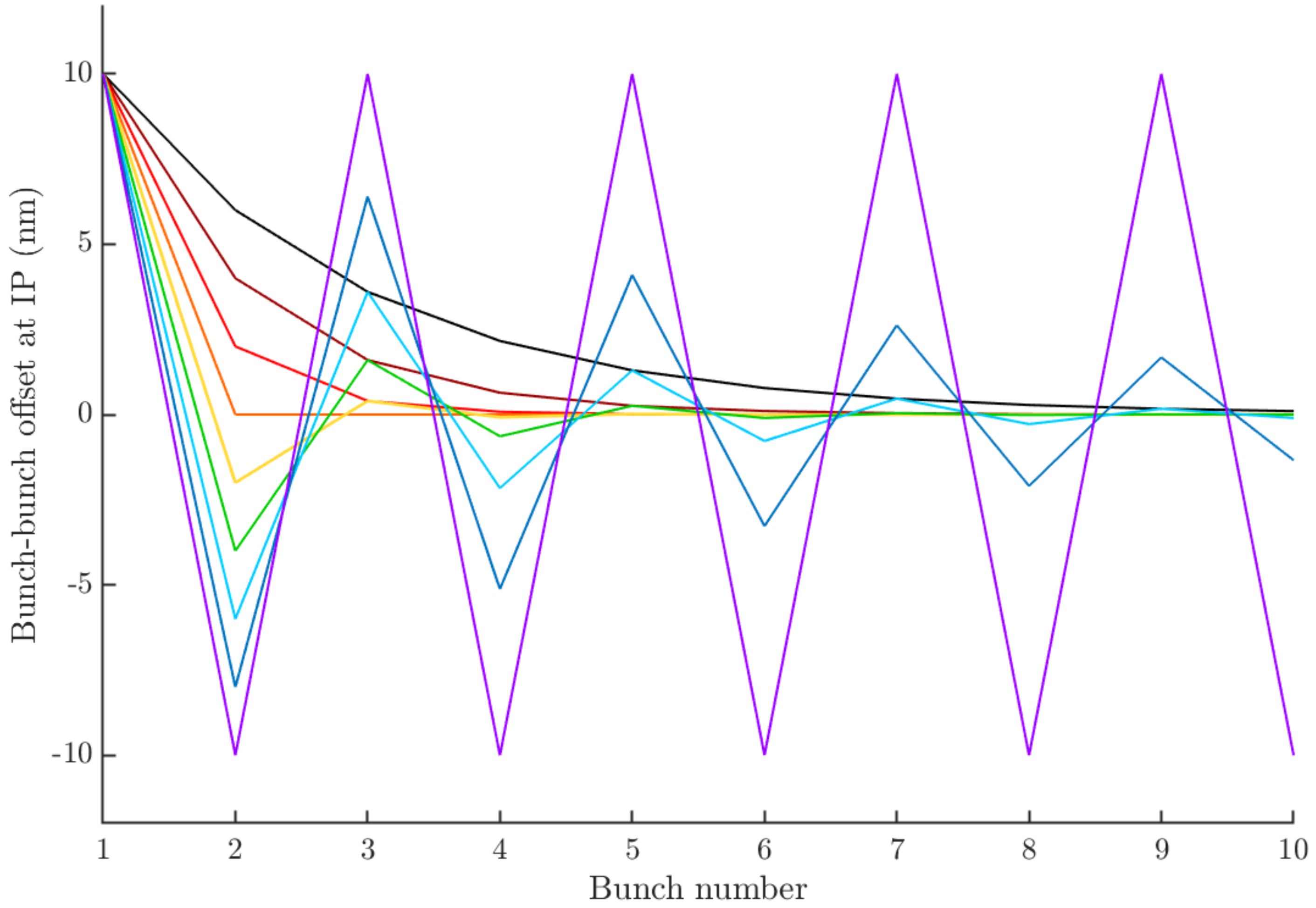
Two rigid bunch trains with zero offset (i.e requiring no correction), with a BPM with a range of resolutions (0 to 20 um).

As expected, very little degradation to luminosity from a BPM with up to 1 um resolution.

Fractional luminosity for FB system with 1 um resolution BPM and rigid beam: 0.9985 * $L_0$.



# Gain scan




Rigid bunch trains with 10 nm initial offset and proportional gain feedback. Scaling the feedback gain. (1 is nominal)

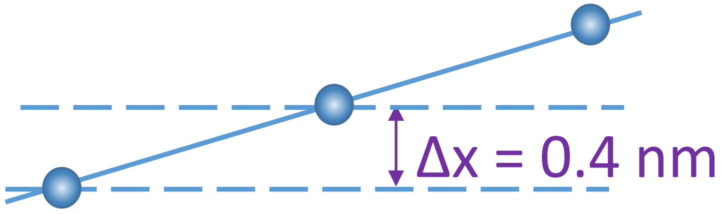
Gain:

0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8, 2

# Bunch train structures

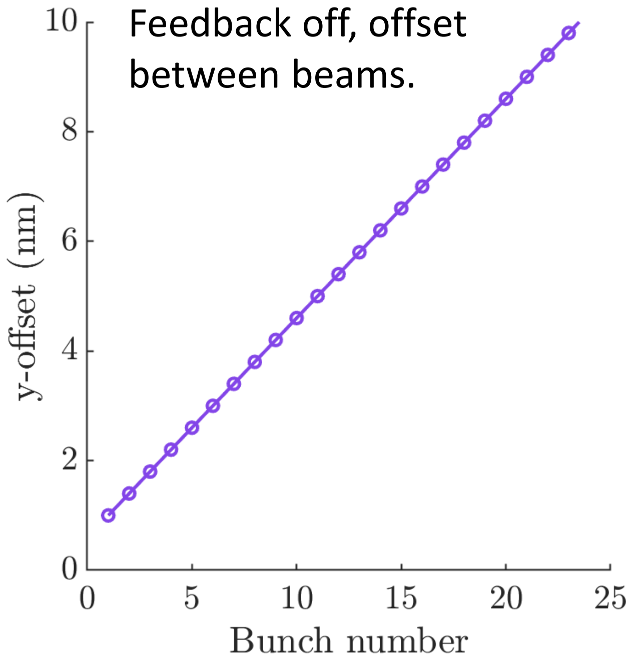
# Bunch train structure

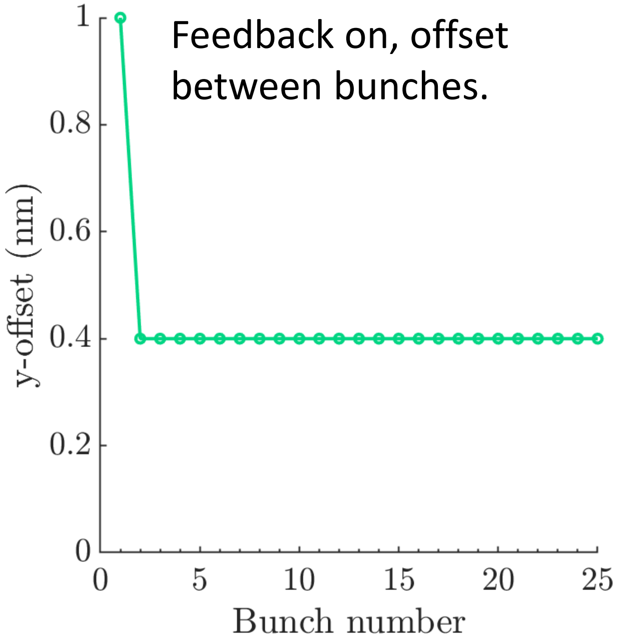
Bunch train: 1 nm initial offset with offset of 0.4 nm between consecutive bunches.

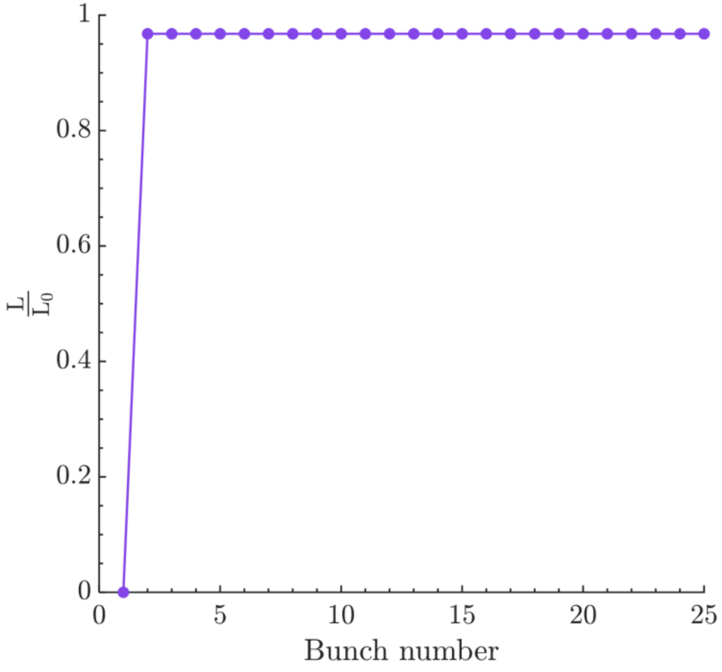



Proportional gain feedback, with 0 nm resolution. Would not take out the 0.4 nm difference between consecutive bunches.

Leading to a loss in luminosity.



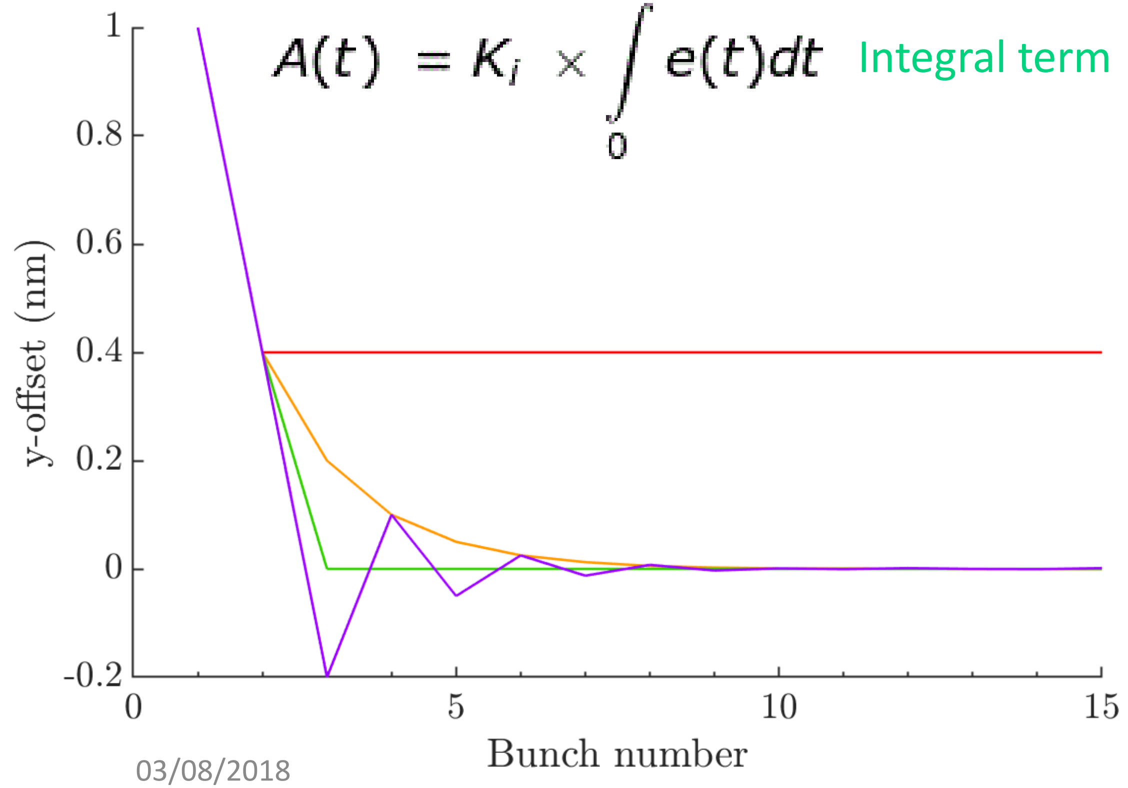



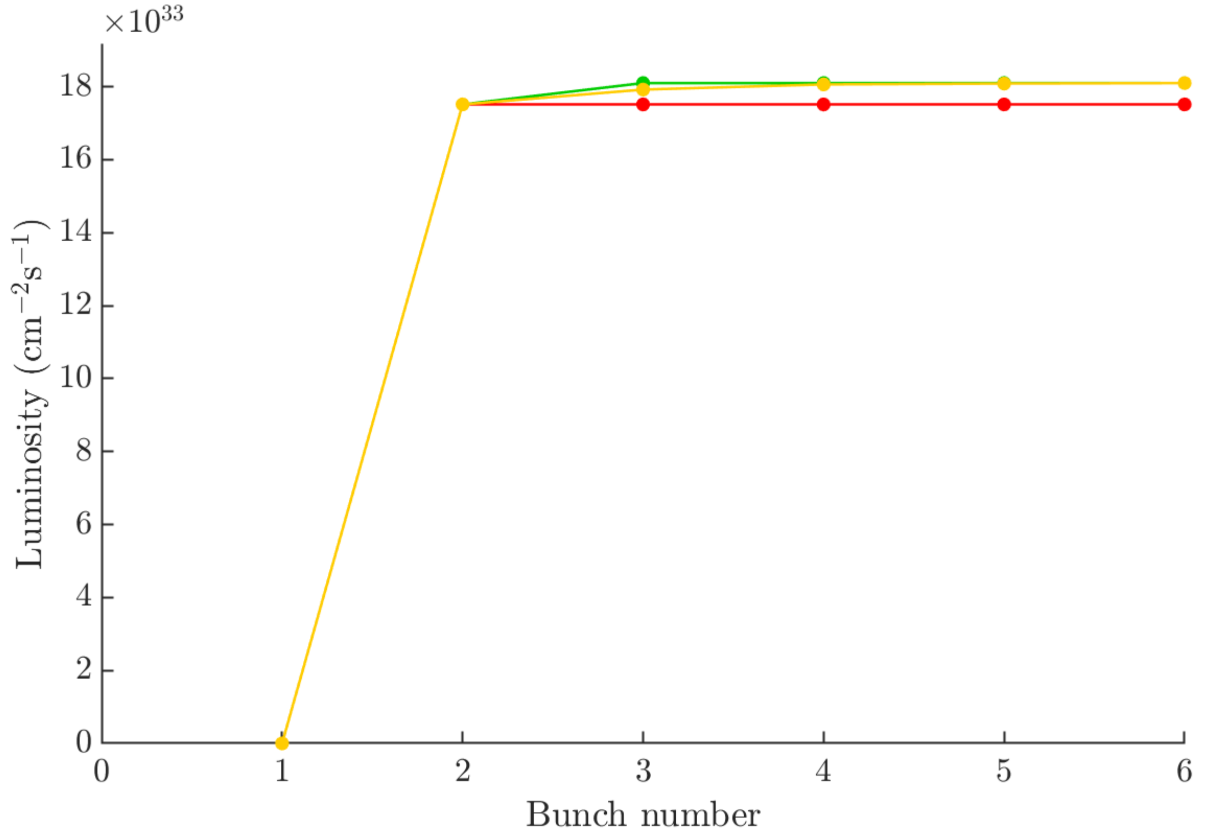
# Proportional Integral Control

0.4 nm offset between consecutive bunches. The 0.4 nm is not removed by just proportional feedback.
Nominal gains for proportional term,
Gains for integral term: 0, 0.5, 1, 1.5

$$A(t) = K_p \times e(t)$$ Proportional term

$$A(t) = K_i \times \int_0^t e(t)dt$$ Integral term



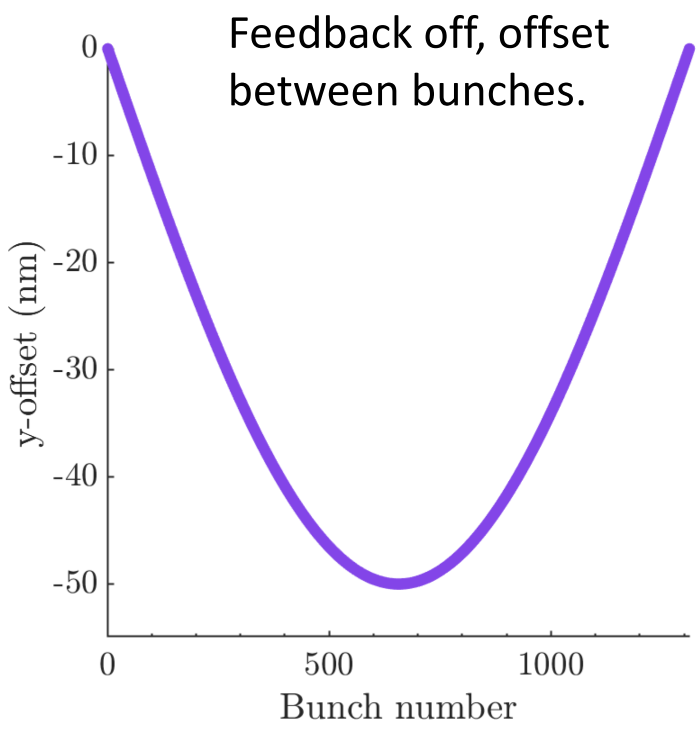



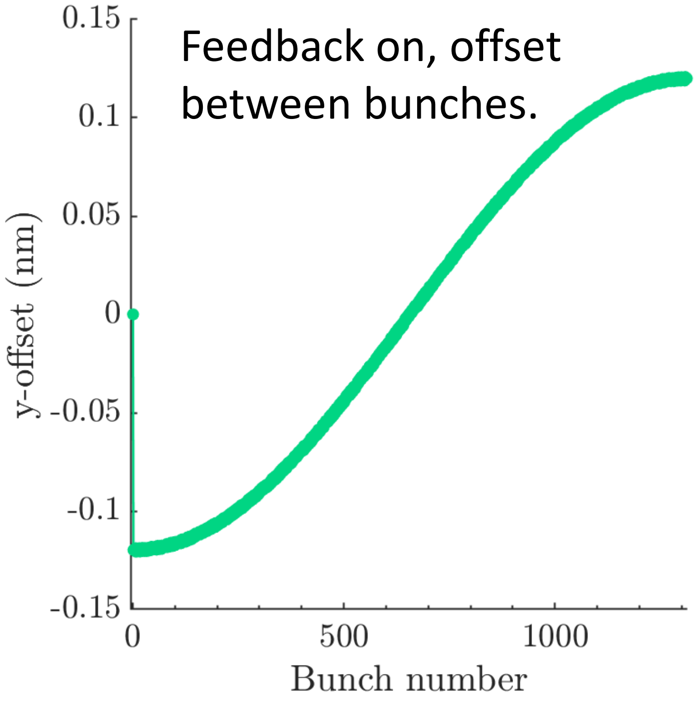
# Harmonic (across train)

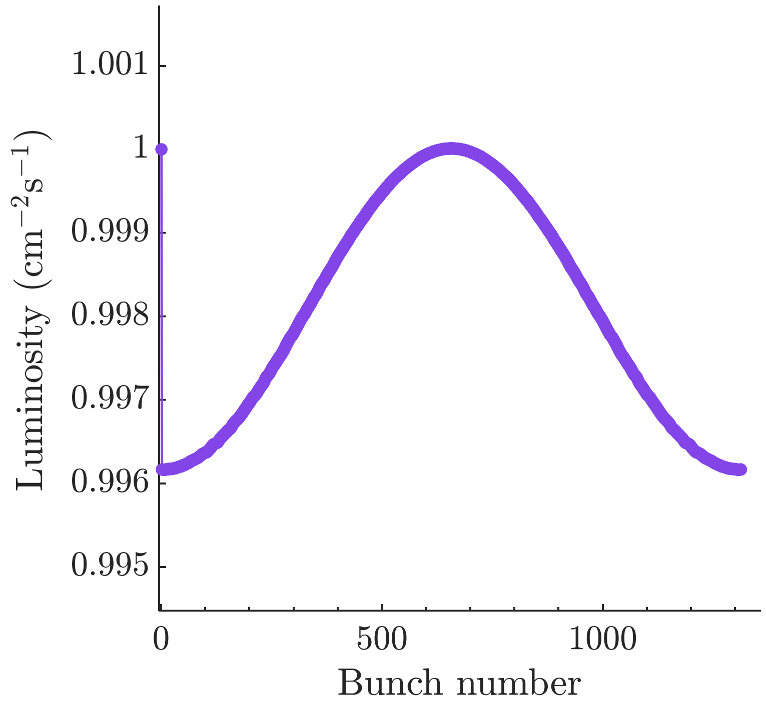
Bunch train: Banana shape (half cycle across bunch train).

**Proportional gain feedback**, with 0 nm resolution. Would not take out the difference between consecutive bunches. (Effectively differential of harmonic)

Leading to a loss in luminosity.

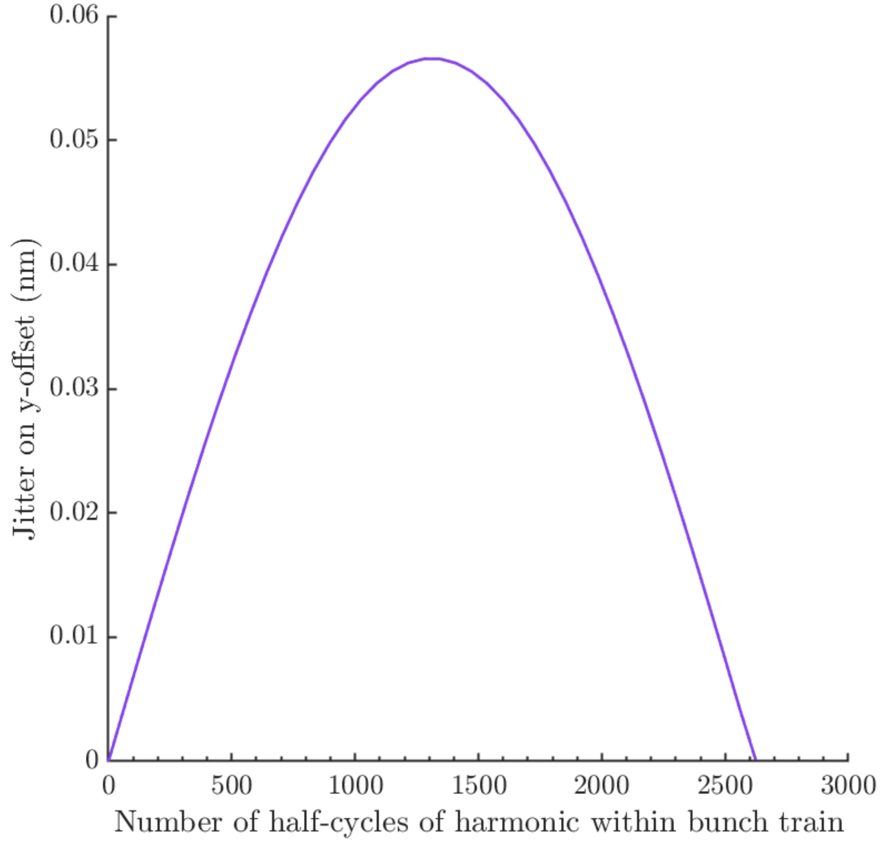


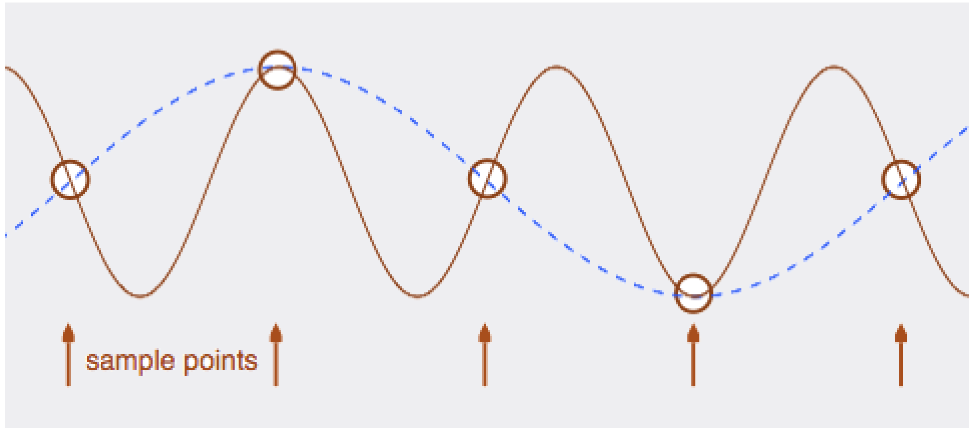



# Frequencies of harmonic

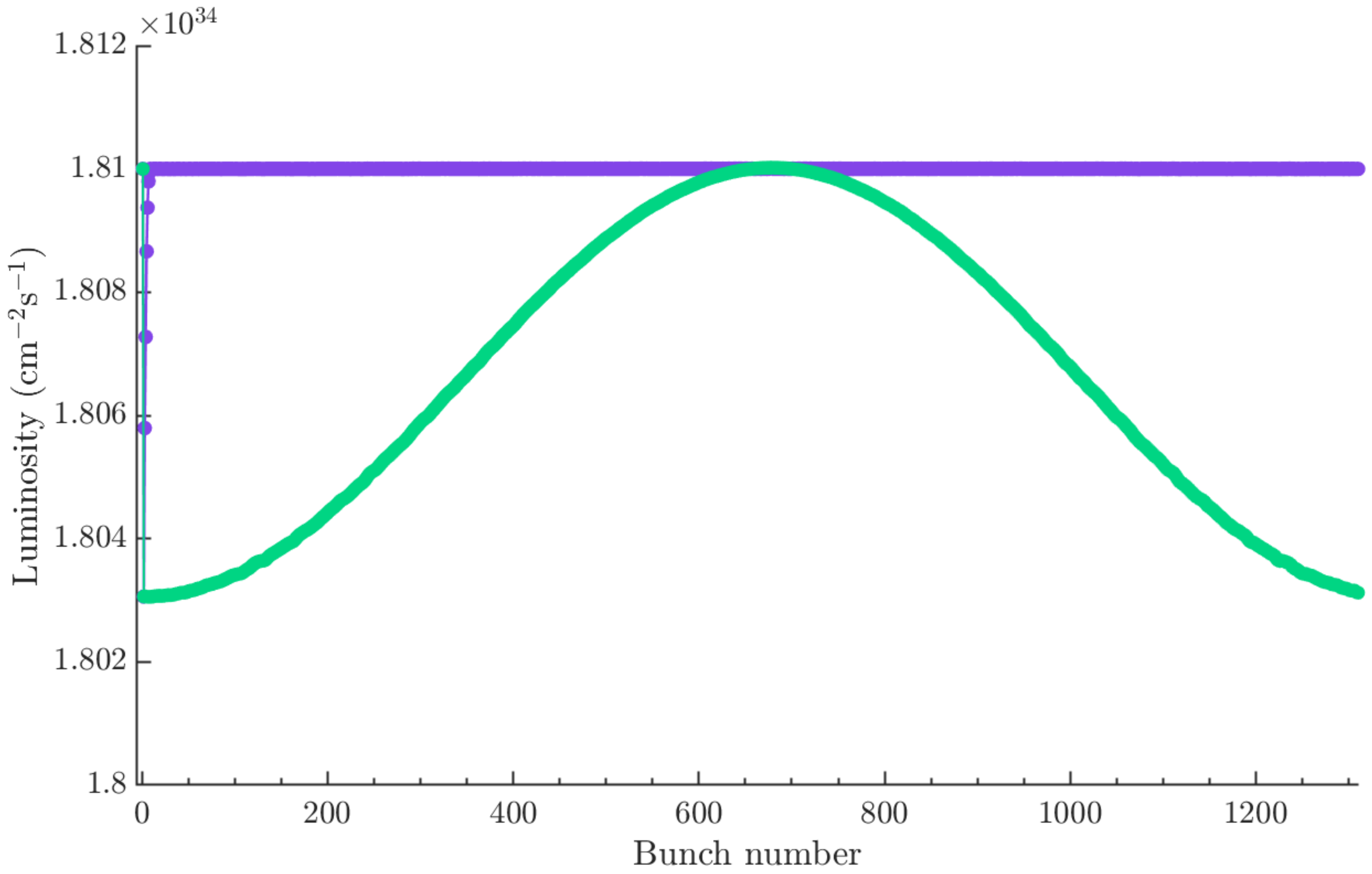
Jitter here defined as standard deviations of y-offsets (between beams at the IP) across a bunch train after feedback.




- Dependence of feedback performance on frequency of harmonic structure introduced to bunch train.
- Worst performance (peak of curve) occurs when frequency of harmonic =1312*pi, as this corresponds to bunch-to-bunch correlation of -1.
- Higher frequencies than 1312*pi when sampled at the bunch frequency are equivalent to lower frequency harmonics and consequently perform better.



# Proportional Integral Control: Harmonic




Proportional control vs. proportional integral control. For harmonic bunch train shape shown on slide 17.

Integral gain $K_i$: 0

Integral gain $K_i$: nominal

$$A(t) = K_p \times e(t)$$ Proportional term

$$A(t) = K_i \times \int_0^t e(t)dt$$ Integral term

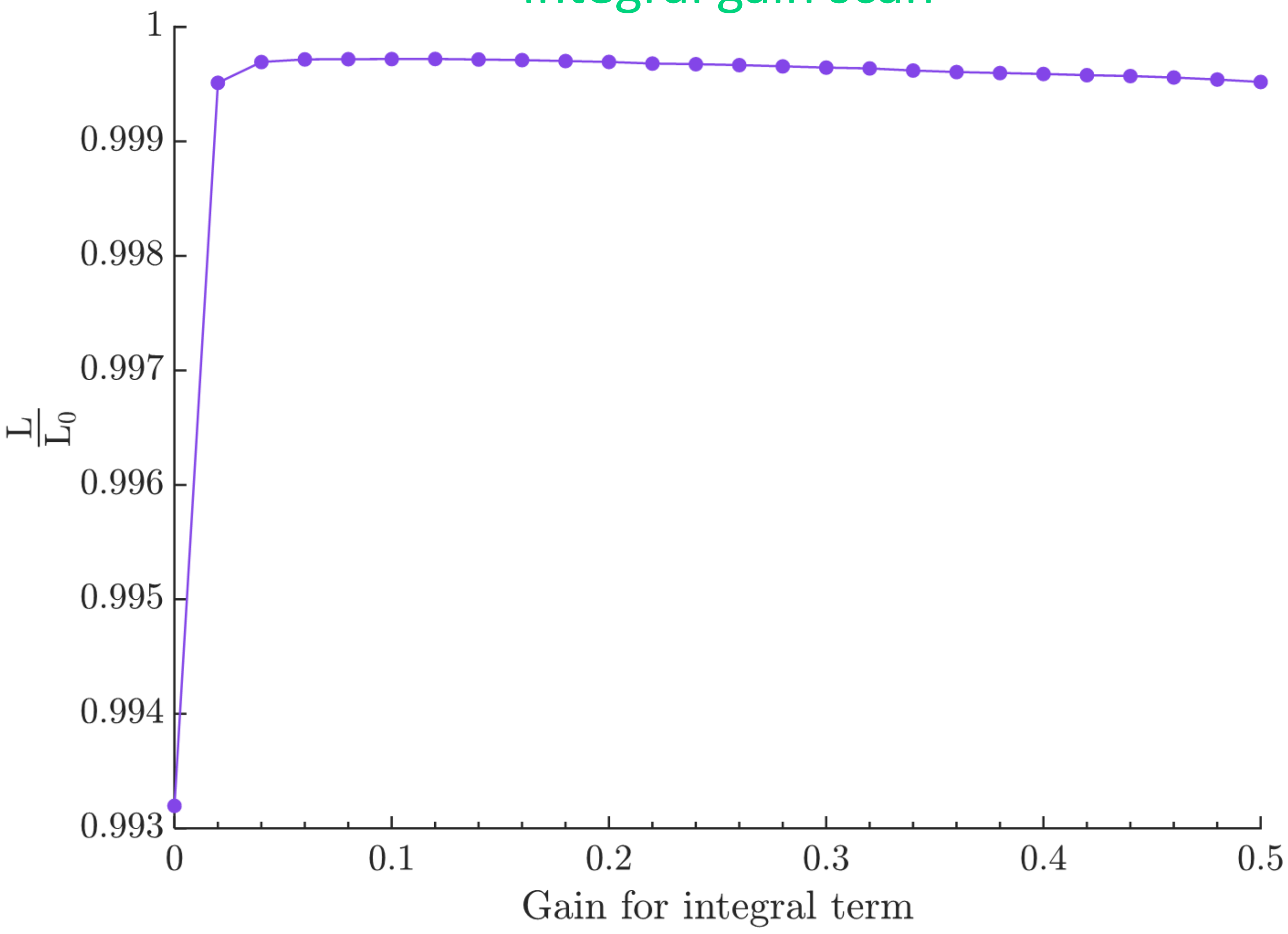
# Harmonic structure and BPM Resolution

- Proportional gain $K_p$ = 1;
- Integral gain $K_i$: shown on *x*-axis.
- BPM resolution: 1 um
- Harmonic term (um) =
  $0.1 \times \cos((t \times \pi)+\frac{\pi}{2})$

where t = 0:1/1311:1, half period across bunch train.

- If resolution is the dominant effect integration makes it worse. If harmonic is the dominant effect integration makes it better.

Integral gain scan

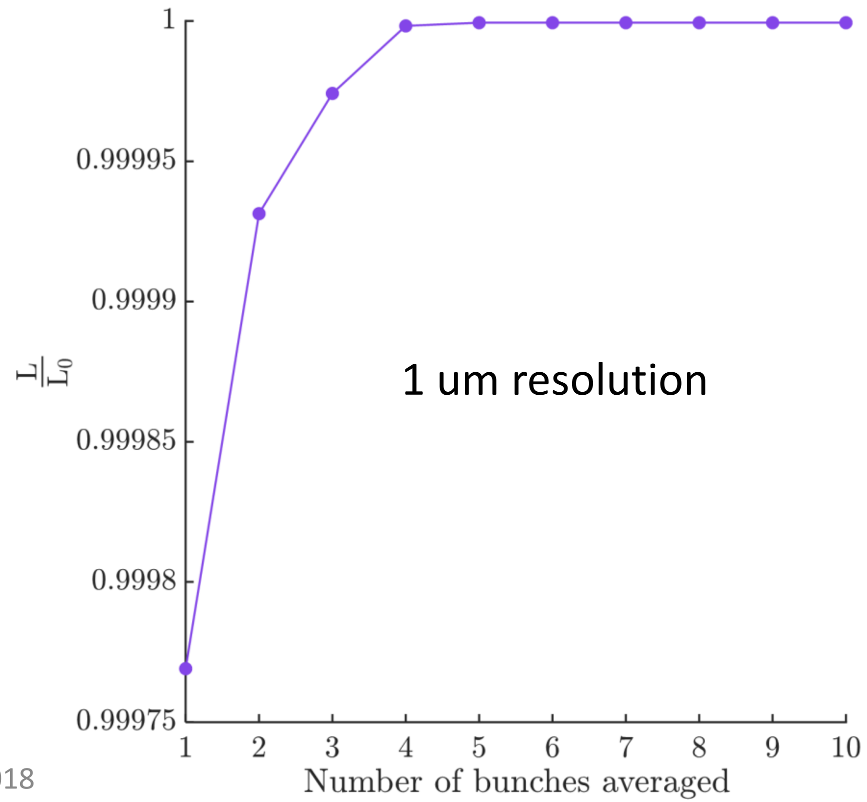


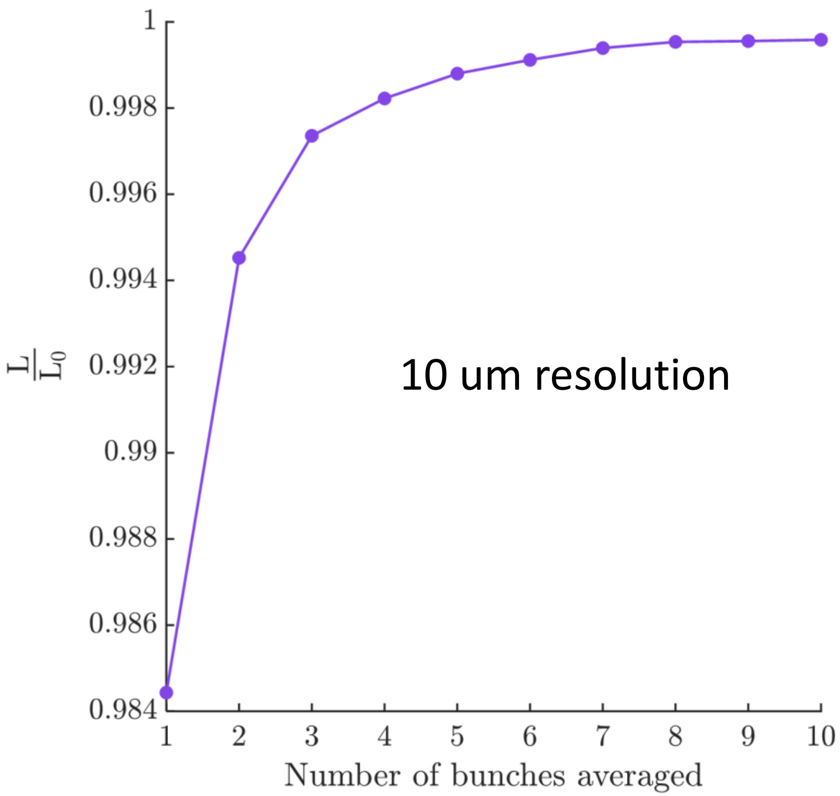
# Averaging Bunches to Improve Bunch Position Measurement

# Averaging over multiple bunches

If noise added is uncorrelated between consecutive bunches, e.g resolution effects then averaging offers some improvement. Very minor improvement for resolution of 1 um. Initially rigid trains with offset 70 nm.

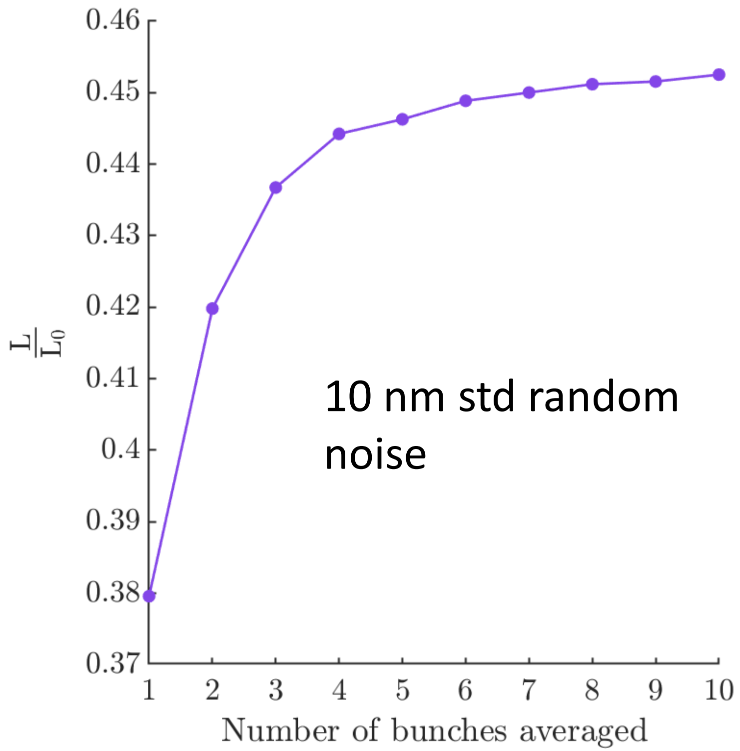
Would become important for beams with poorer bunch to bunch correlation.

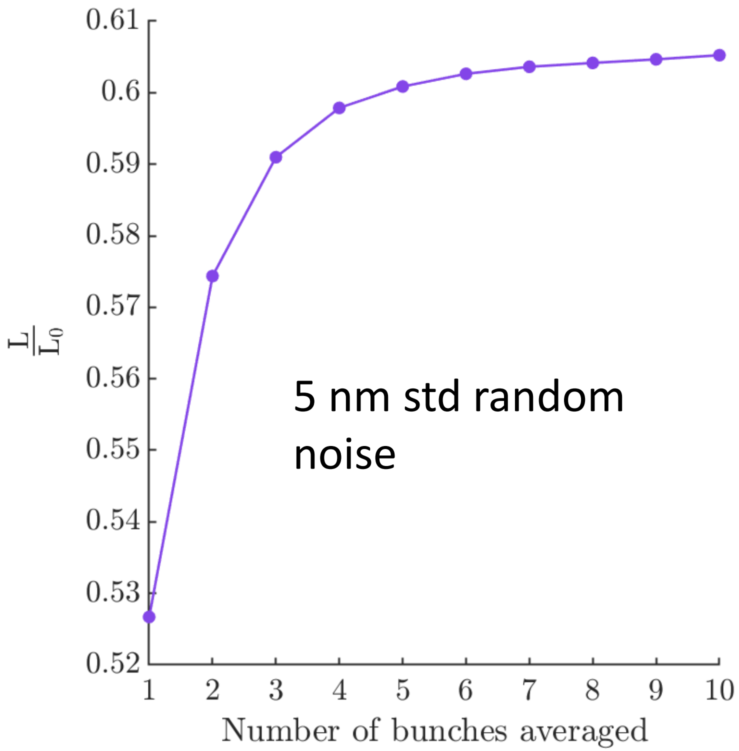


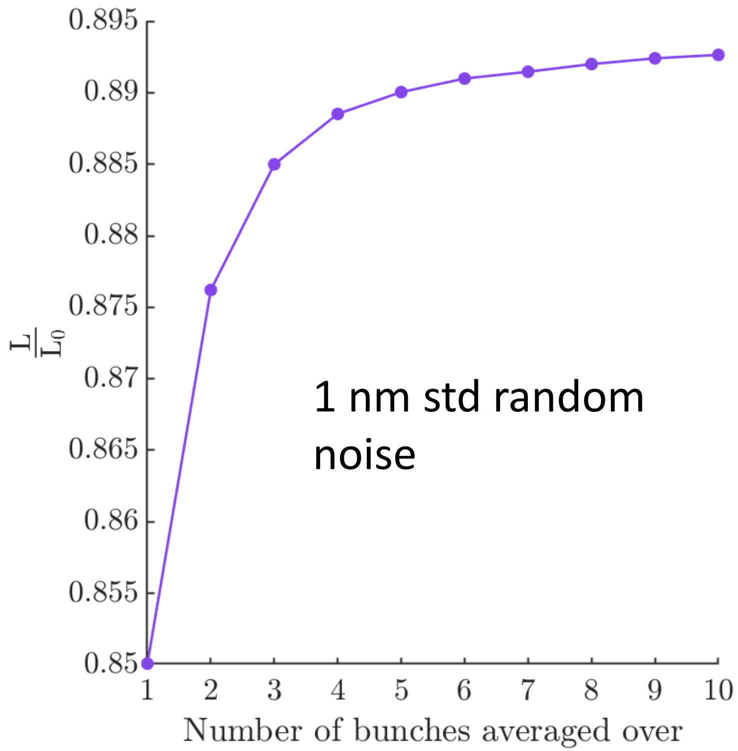



# Reduced bunch-to-bunch correlation

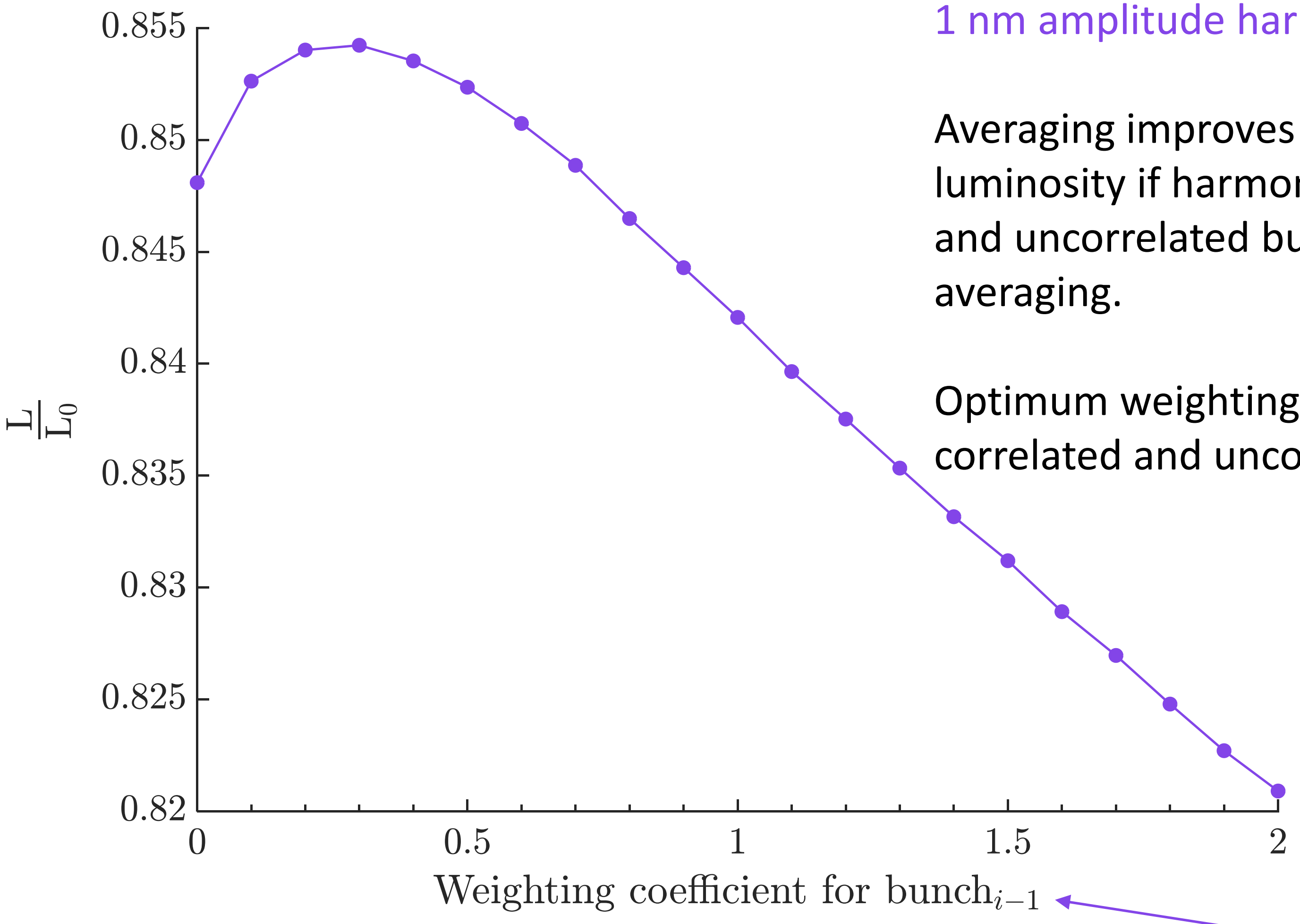
- Initial offset 70 nm with random noise added to each bunch.
- Random noise introduced bunch by bunch to reduce correlation and degrade feedback performance.
- Position measurements fed into feedback system are now averaged over some number of bunches (x-axis) and the average luminosity recovered shown (y-axis)



# Weighted averaging over **two** bunches




z

1 nm random noise added bunch by bunch **AND**
1 nm amplitude harmonic – half period across bunch train.

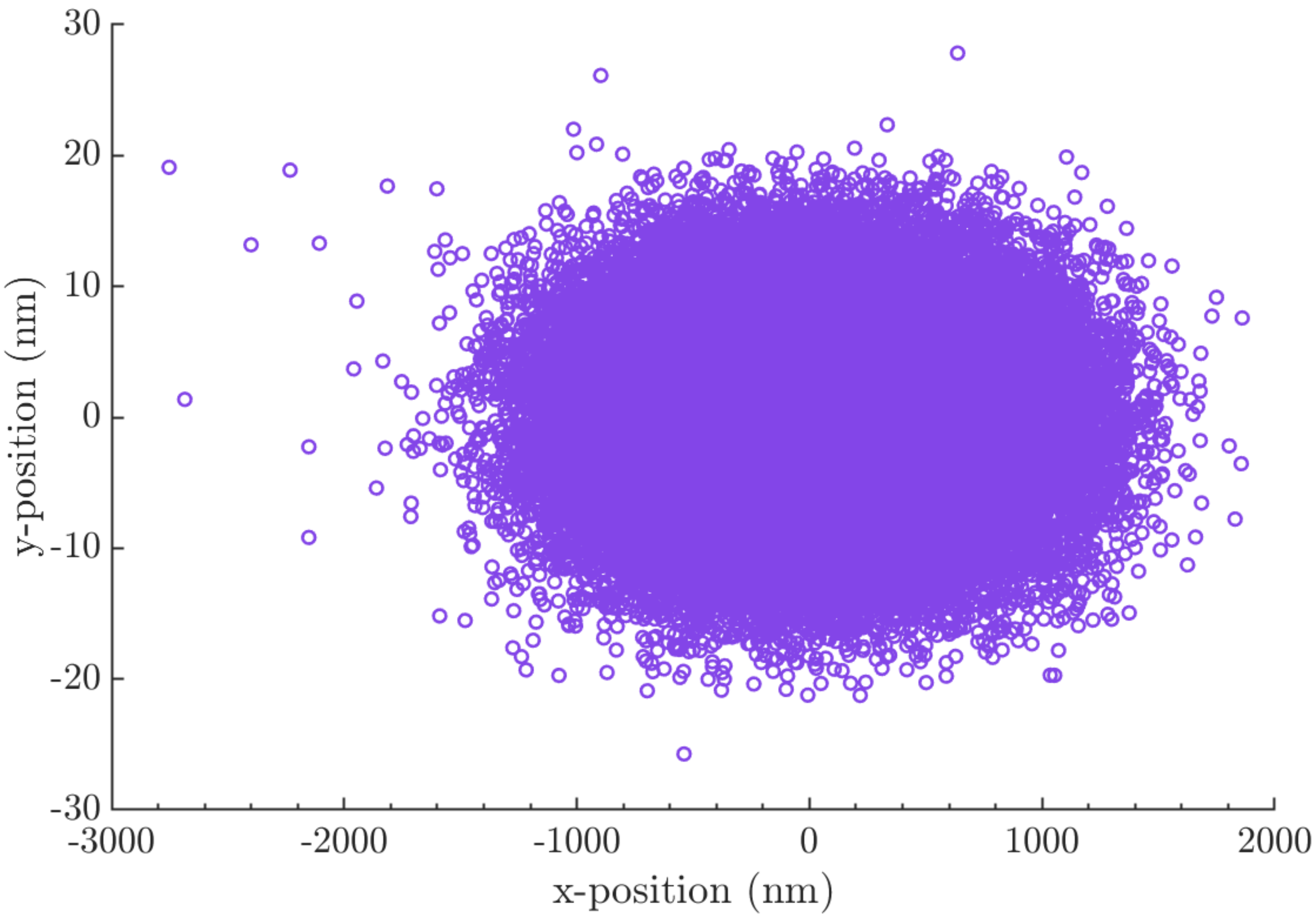
Averaging improves if random noise is the dominant effect but degrades luminosity if harmonic is the dominant effect. If combination of correlated and uncorrelated bunch-to-bunch effects then can use weighted averaging.

Optimum weighting coefficient depends on magnitude of effects which are correlated and uncorrelated between consecutive bunches.

$$\text{Bunch position}_i = \frac{\text{Bunch position}_i + z \times \text{bunch position}_{i-1}}{1+z}$$

# Tracking the bunches through the BDS

# Bunches at IP (Lucretia)



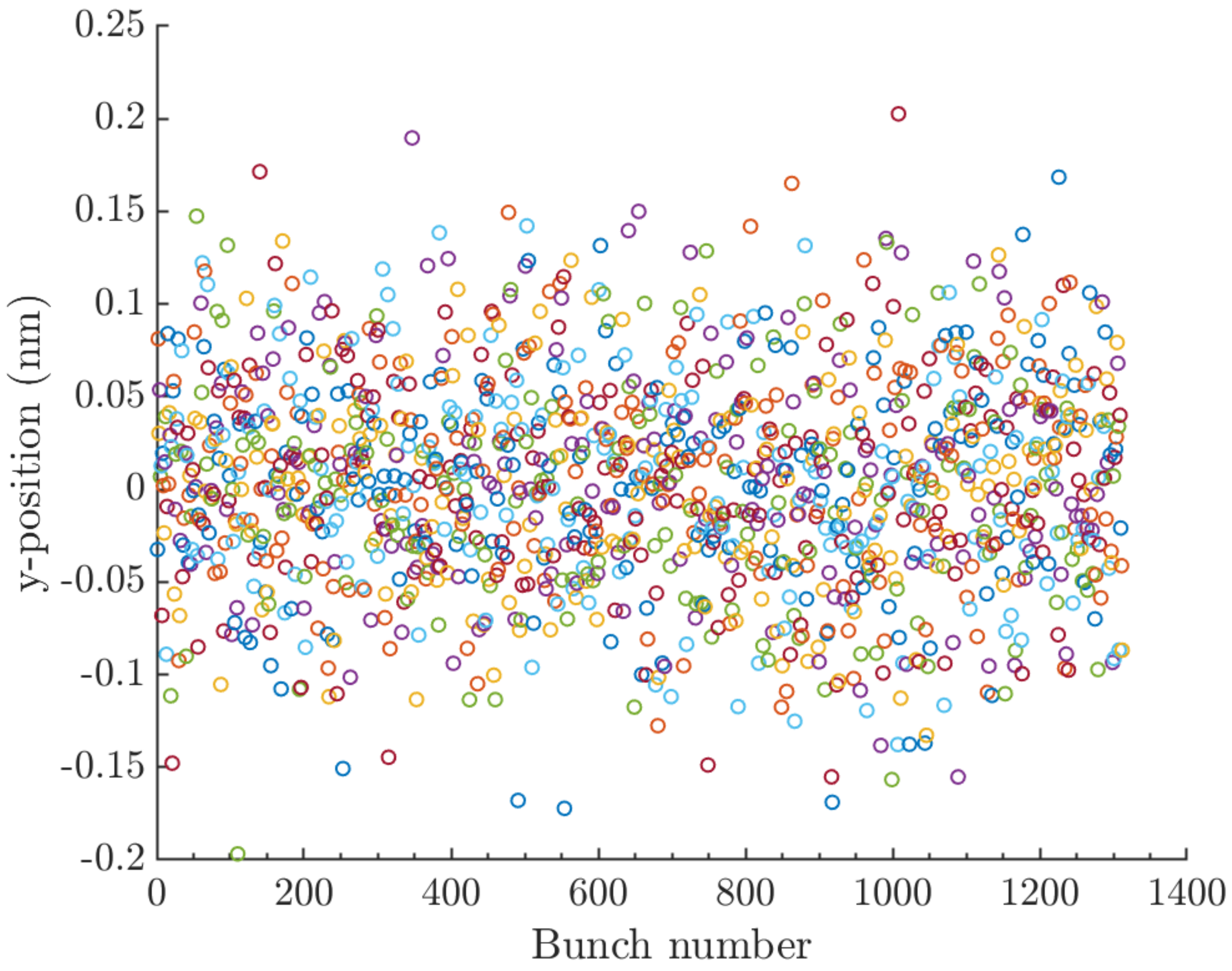

Horizontal beam size = 474 nm
Vertical beam size = 5.9 nm
80000 macroparticles.

I still need to figure out which/if any wakefields are implemented.

MATLAB structure called WF.ZSR, possibly short range wakefields?

First bunch in train shown.

# 1312 bunch train tracked through BDS

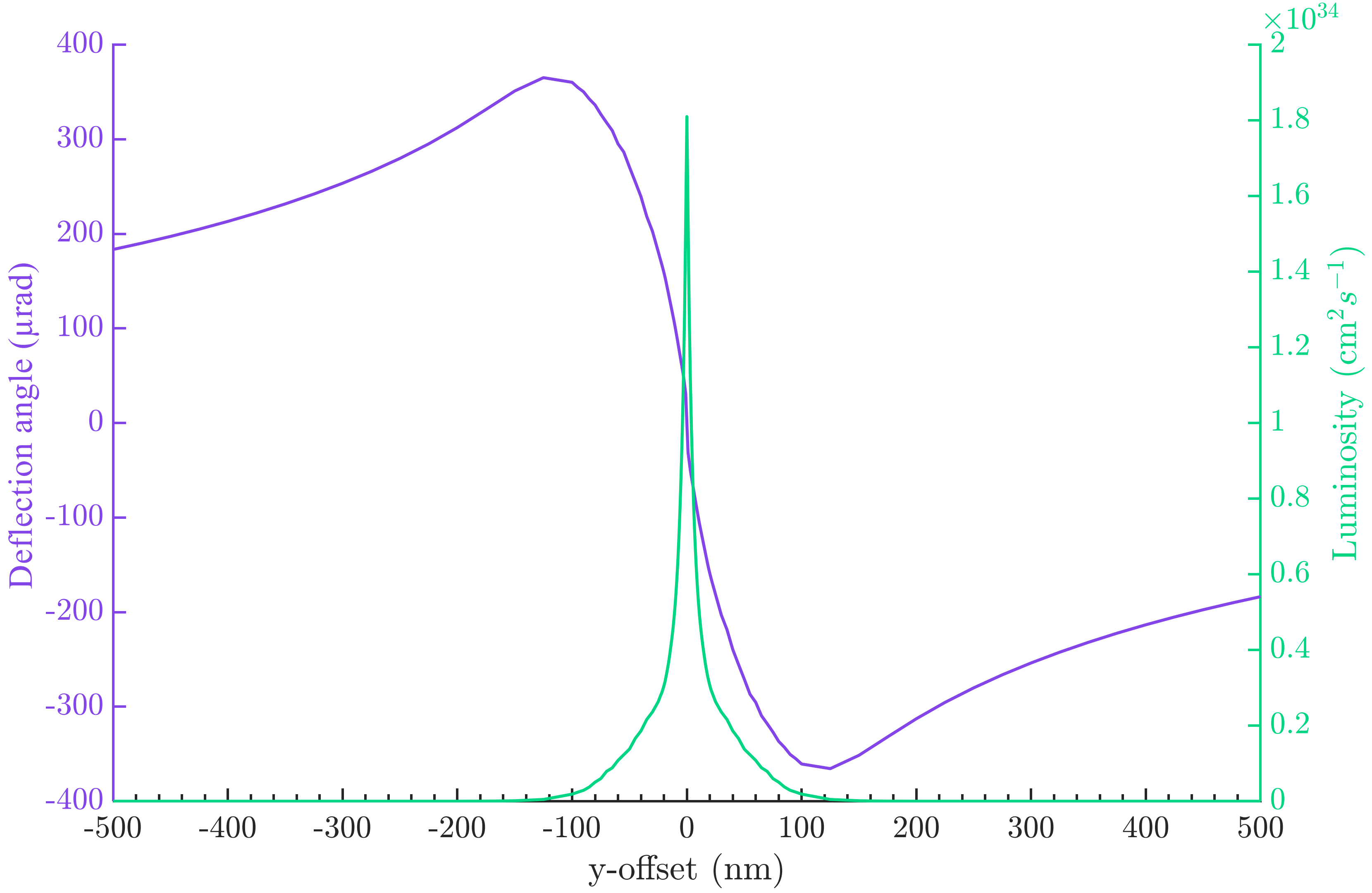



*y*-position of bunches at IP modelled in Lucretia for a train of 1312 bunches tracked through the BDS to the IP.

10000 macroparticles modelled per bunch, mean bunch position shown.

# Deflection Angle Curves

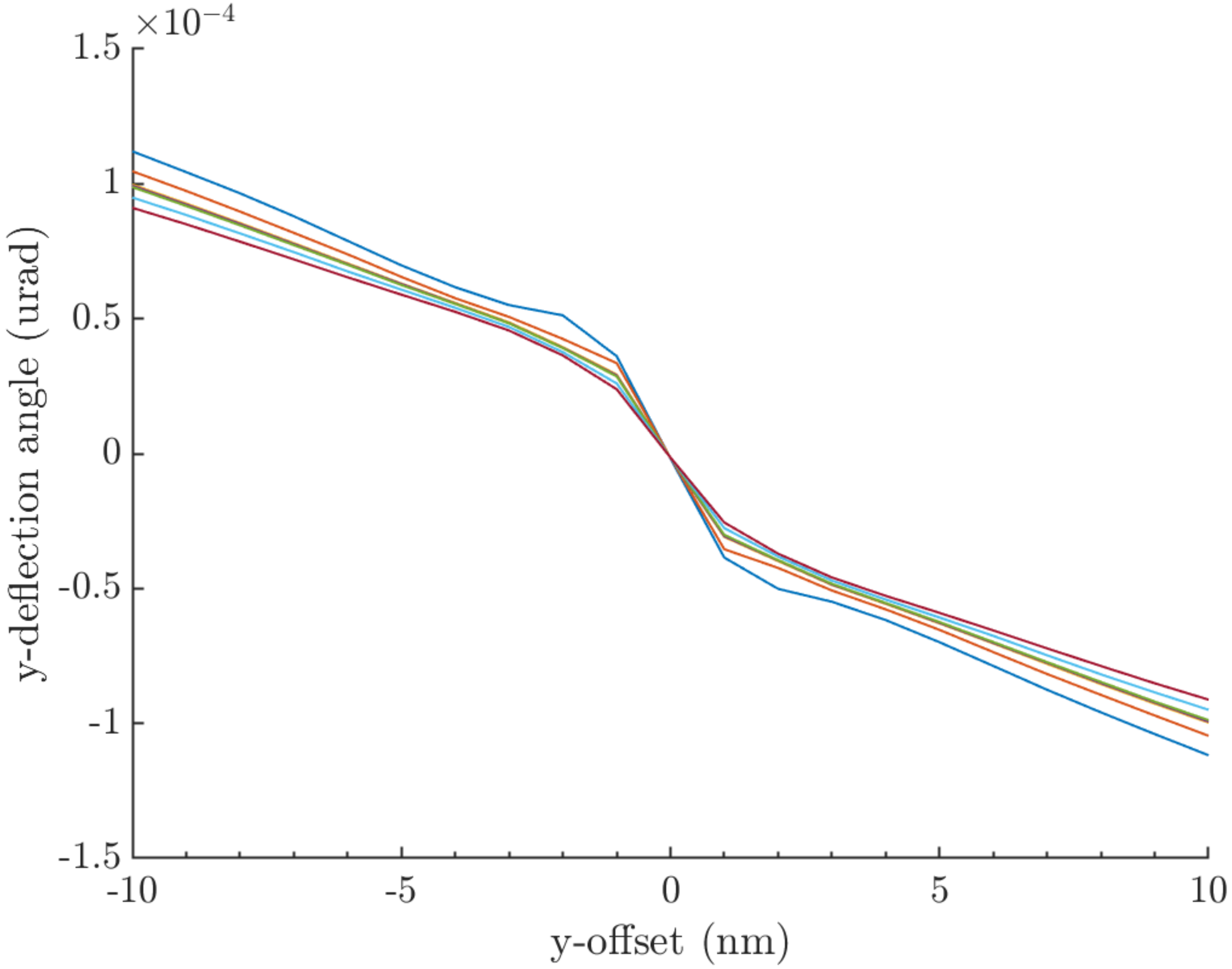
# Nominal Curves




Nominal beam-beam deflection angle curve and luminosity curve vs. beam-beam offset at the IP.

Maximum deflection: 365 urad
Maximum luminosity: $1.81\times 10^{34}$

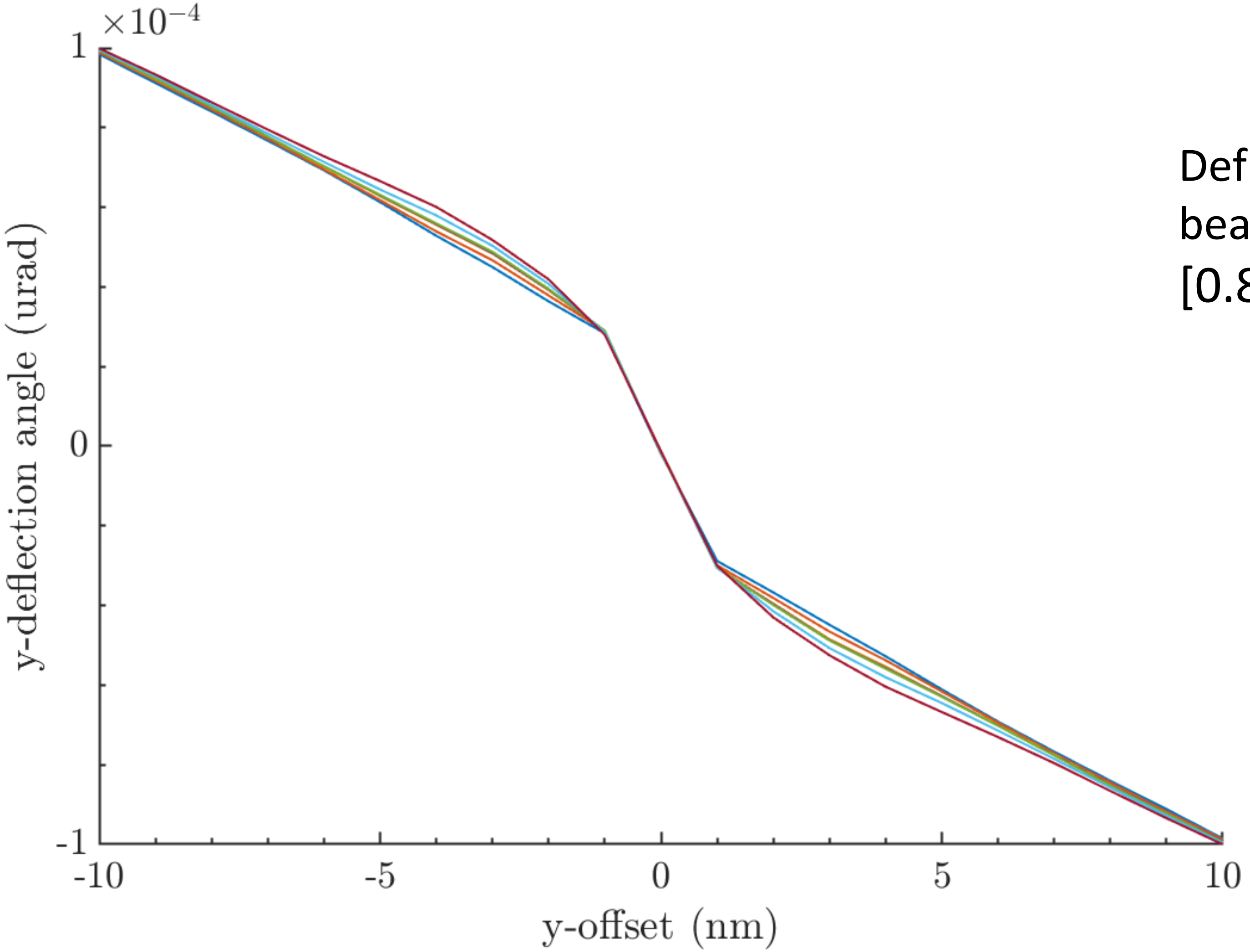
# Horizontal beam size (sigma_x)




Deflection-angle curves for a range of horizontal beam size values (sigma_x):

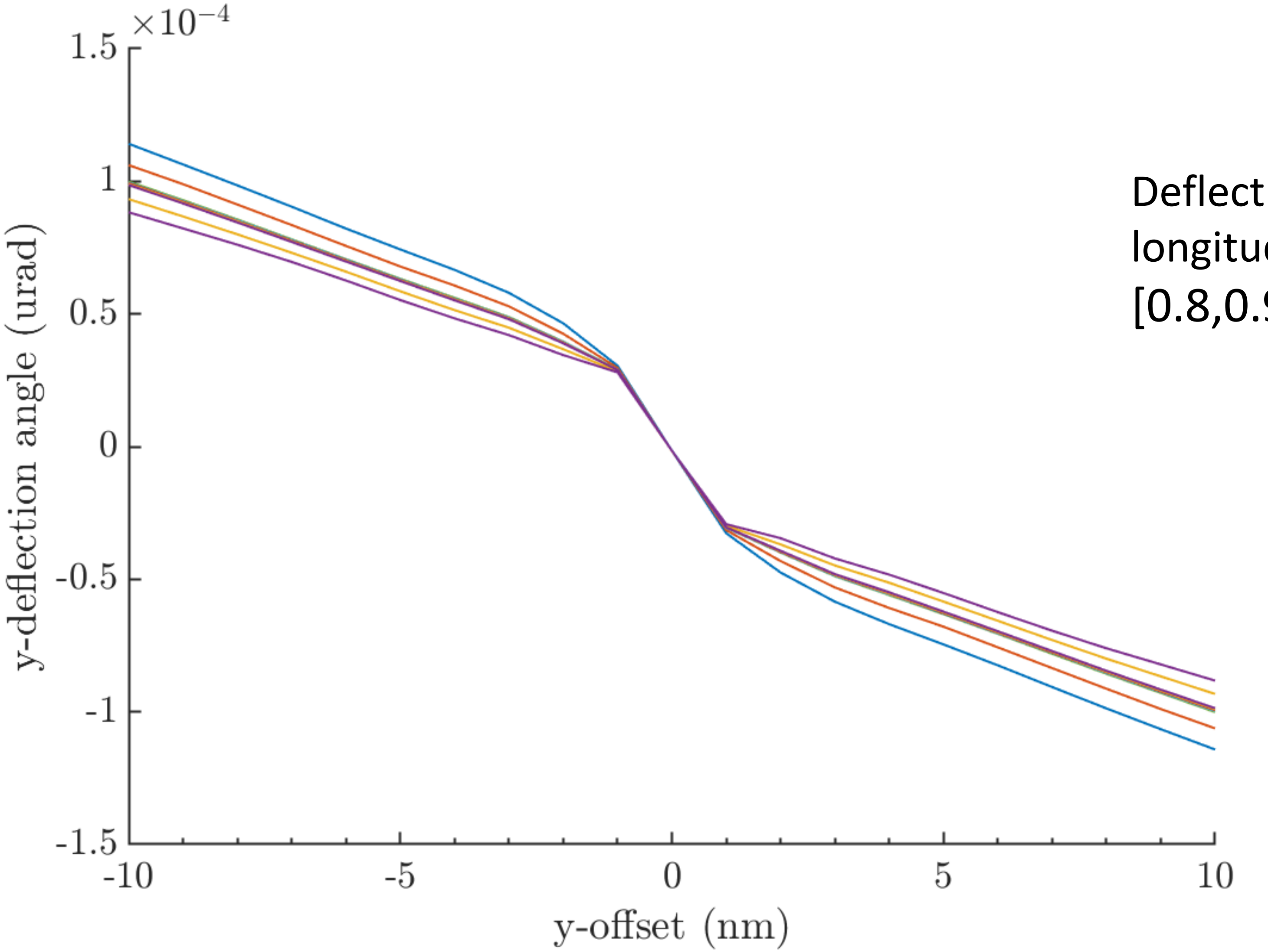
[0.8,0.9,0.99,1,1.01,1.1,1.2]*474 nm

# Vertical beam size (sigma_y)



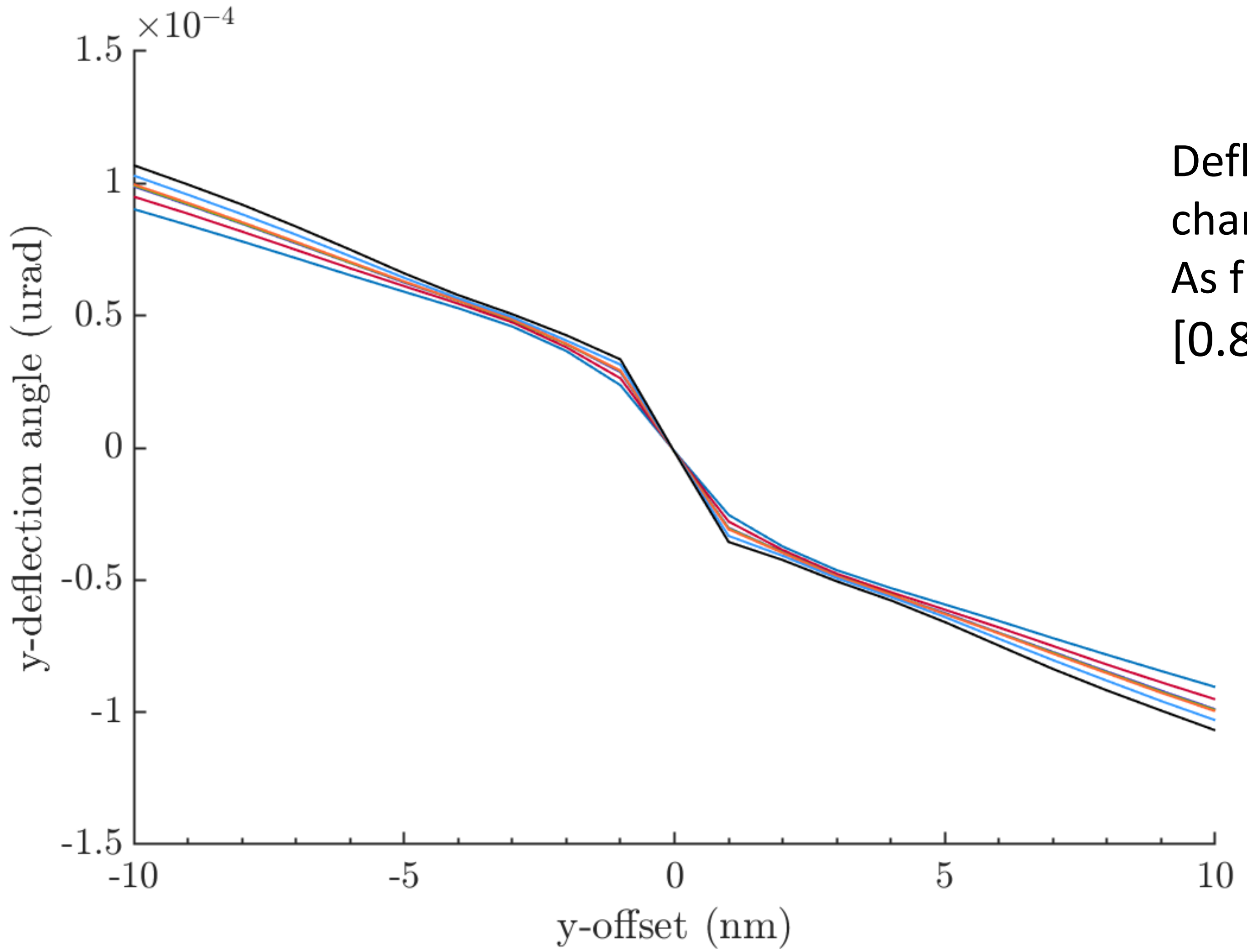

Deflection-angle curves for a range of vertical beam size values:
[0.8,0.9,0.99,1,1.01,1.1,1.2]*5.9 nm

# Bunch length (sigma_z)



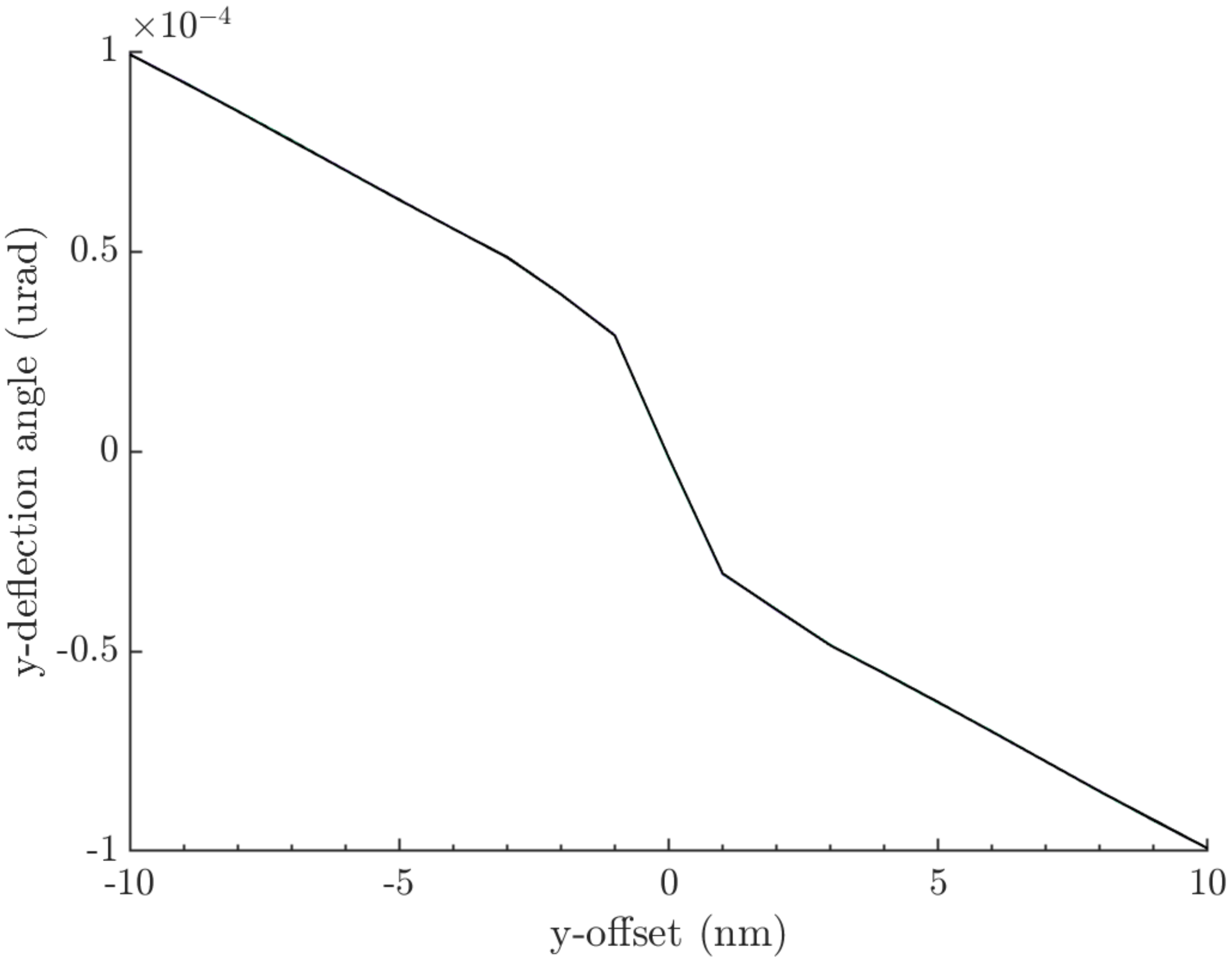

Deflection-angle curves for a range of longitudinal beam size values: [0.8,0.9,0.99,1,1.01,1.1,1.2]*300 um

# Charge (number of particles per bunch)




Deflection-angle curves for a range of bunch charges:
As function of number of particles per bunch [0.8,0.9,0.99,1,1.01,1.1,1.2]*2e10.

# Energy




Deflection-angle curves for a range of energies:
[248:0.1:252] GeV.

Shows no real variation with bunch energy ✓

# Summary

- Basic model of FB system implemented in MATLAB.
- Ability to model bunch trains: - bunch structures inc. constant offsets, offsets between consecutive bunches, harmonics of a range of frequencies.
- Ability to model FB system: proportional control, proportional and integral control, averaging over multiple bunches, weighted averaging.
- Generate 1312 bunch train in Lucretia and track through the BDS to the IP, model bunch-bunch interaction at the IP using Guinea-Pig.
- Deflection angle curves modelled for various beam parameters showing dependence on: bunch size (x,y), bunch length, charge.