BNL-229661-2026-TECH

CA/AP/744

Longitudinal Twiss parameters for tilted phase space ellipses

S. Peggs

April 2026

Collider Accelerator Department

Brookhaven National Laboratory

U.S. Department of Energy

USDOE Office of Science (SC), Nuclear Physics (NP)

DISCLAIMER

CA/AP/744

Longitudinal Twiss parameters for tilted phase space ellipses

Steve Peggs, Henry Lovelace III

April 16, 2026

Abstract

The longitudinal emittance ϵ_s is usually defined under the implicit assumption that the longitudinal phase space distribution is erect, an assumption that is significantly incorrect near transition. This note generalizes the definition of ϵ_s to include tilted distributions, by introducing longitudinal Twiss functions β_s, α_s and γ_s that are analogous to transverse Twiss functions. The evolution of these longitudinal parameters through RHIC transition is presented, for a typical set of simulation data.

Contents

List of Figures

List of Tables

1 Transverse parameters – reprise

1.1 Periodic linear motion

No particles: If transverse turn-by-turn motion around a ring is stable, the linearized (small amplitude) motion is represented by the one-turn matrix

$$M(s) \;=\; \begin{pmatrix} \cos(\mu) + \alpha\,\sin(\mu) & \beta\,\sin(\mu) \\ -\gamma\sin(\mu) & \cos(\mu) - \alpha\,\sin(\mu) \end{pmatrix} \tag{1}$$

where α, β and γ are *periodic* Twiss functions at a chosen azimuthal reference point. In order that M has a determinant of one the Twiss identity must be true:

$$\gamma \;\equiv\; \frac{1+\alpha^2}{\beta} \tag{2}$$

Defined in this way the (periodic) Twiss functions are a property of the ring – no particles are involved!

One particle: The transverse displacement x and angle x' of a single particle on turn n are then

$$\begin{aligned} x_n &= \sqrt{2J\beta}\;\;\sin(\phi) \\ x'_n &= \sqrt{2J/\beta}\;[\cos(\phi) - \alpha\sin(\phi)] \end{aligned} \tag{3}$$

where the phase on turn n is

$$\phi_n \;=\; \mu\, n + \phi_0 \tag{4}$$

Each single particle is labeled by its own action J, a constant that can be found through

$$J \;=\; \frac{1}{2}\left(\beta\, x'^2 \;+\; 2\alpha\, xx' \;+\; \gamma\, x^2\right) \tag{5}$$

if x and x' are known on *any* turn. Equation 5 describes the ellipse in (x, x') space traced by the particle as n increases turn-by-turn, if the motion is linear.

Many particles: The mean square transverse displacement of a *single* particle is proportional to its action since

$$\left\langle x^2 \right\rangle_1 \;=\; 2J\beta\left\langle \sin^2(\phi)\right\rangle \;=\; J\beta \tag{6}$$

where the subscripted angle brackets $\langle\rangle_1$ indicate an average taken over very many turns. Thus, the mean square size of a many-particle bunch with an action distribution $\rho(J)$ is

$$\left\langle x^2 \right\rangle \;=\; \frac{1}{N}\int_0^\infty \left\langle x^2\right\rangle_1\, \rho(J)\, dJ \;=\; \frac{\beta}{N}\int_0^\infty J\,\rho(J)\, dJ \tag{7}$$

where the population of the bunch is

$$N \;=\; \int_0^\infty \rho(J)\, dJ \tag{8}$$

Therefore the root mean square (RMS) bunch size is

$$\left\langle x^2\right\rangle^{1/2} \;=\; \sqrt{\beta\,\epsilon} \tag{9}$$

where the unnormalized geometric RMS emittance

$$\epsilon \;=\; \langle J \rangle \tag{10}$$

is simply the average action of the particles in the bunch!

The normalized RMS emittance ϵ_N is related to the geometric emittance through

$$\epsilon_N \;=\; \epsilon\, \beta_L \gamma_L \tag{11}$$

where β_L and γ_L are Lorentz relativistic factors. Ideally ϵ_N is constant in a hadron ring, even under acceleration, although realistic effects usually cause it to grow slowly with time. It only decreases when beam is lost from the distribution tails, or when the beam is cooled.

When the periodic Twiss functions are known from the one-turn matrix M then all 3 second order moments are

$$\begin{pmatrix} \langle x^2 \rangle \\ \langle x x' \rangle \\ \langle x'^2 \rangle \end{pmatrix} = \epsilon \begin{pmatrix} \beta \\ -\alpha \\ \gamma \end{pmatrix} \tag{12}$$

where un-subscripted angle brackets $\langle\rangle$ indicate an average over *all* particles in a bunch on one particular turn. It is assumed for simplicity that there are no closed orbit errors so that first order moments

$$\langle x \rangle = \langle x' \rangle = 0 \tag{13}$$

are zero. It is also assumed that the phases of the particles are completely uncorrelated – transients have settled down and the moments of the distribution do not change from turn to turn.

1.2 Single-pass or transient motion

In a single-pass transfer line or linac there is no one-turn matrix that can be used to define the *periodic* Twiss functions. In this case (or when the motion is significantly nonlinear) *transient* or *beam-based* Twiss functions are often derived from the simulated or observed beam distribution, through the equations

$$\begin{pmatrix} \beta \\ -\alpha \\ \gamma \end{pmatrix} = \frac{1}{\epsilon} \begin{pmatrix} \langle x^2 \rangle \\ \langle x x' \rangle \\ \langle x'^2 \rangle \end{pmatrix} \tag{14}$$

A perfect closed orbit is again assumed for simplicity, so that the first order moments $\langle x \rangle$ and $\langle x' \rangle$ are zero. The geometric RMS emittance is then found from the second order moments through

$$\epsilon = \sqrt{\langle x^2 \rangle \langle x'^2 \rangle - \langle x x' \rangle^2} \tag{15}$$

after invoking the Twiss identity of Equation 2. Equations 14 and 15 can be applied to *any* phase space distribution – non-Gaussian, filamented, distorted, or otherwise – so long as the second order distribution moments are known.

Beam-based Twiss parameters describe the shape of an ellipse that fits the distribution (well or badly), no matter how contorted or filamented the distribution may be. If the action that parametrizes the fitted ellipse is

$$J_{\text{ellipse}} = \epsilon \tag{16}$$

then the area of the one-sigma ellipse is

$$A = \pi \, \epsilon \tag{17}$$

The emittance *decreases* along a single-pass beamline in the occasional case when the contortions decrease, even without acceleration or cooling or beam loss [1, 2].

2 Longitudinal parameters $\epsilon_s, \beta_s, \alpha_s, \gamma_s$

What are the analogous longitudinal phase space parameters $\epsilon_s, \beta_s, \alpha_s$ and γ_s? How do they evolve in a circular ring like RHIC, for example during filamentation immediately after injection, or as transition is crossed?

Many of the definitions of longitudinal emittance in the literature differ by the selection of the two independent variables on the longitudinal phase space axes [3]. Here we choose s and δ, where s is the distance that a particle is *ahead* of the nominal bunch center, and δ is the off-momentum parameter

$$\delta = \Delta p/p \tag{18}$$

This choice establishes the closest analogy between longitudinal and transverse phase space parameters, because s and δ have the same dimensions as displacement x (length) and angle x' (dimensionless).

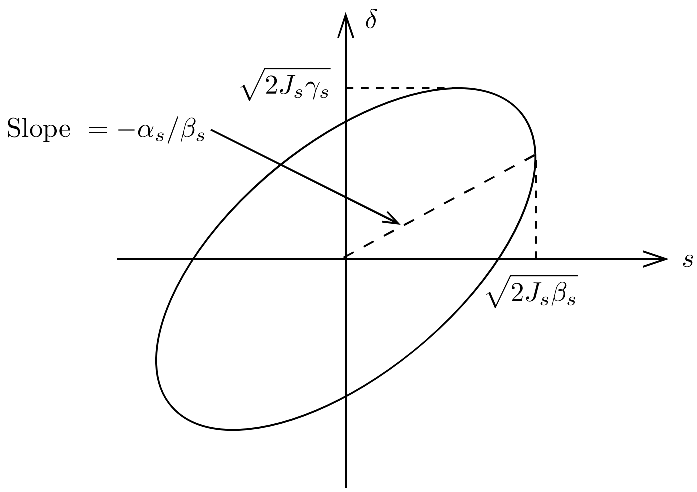

Figure 1: A tilted ellipse in longitudinal phase space (s, δ), where s is the distance in space ahead of a nominal (e.g. synchronous) reference particle, and δ is the off-momentum parameter. Significant tilts occur near transition.

One particle: Figure 1 shows the ellipse that a single particle of longitudinal action J_s traces if its longitudinal motion is purely linear, and if the "closed orbit" offsets $\langle s\rangle$ and $\langle\delta\rangle$ are zero. This motion is represented by the equations

$$\begin{aligned} s &= \sqrt{2J_s\beta_s}\,\sin(\phi_s) \\ \delta &= \sqrt{2J_s/\beta_s}\,(\cos(\phi_s) - \alpha_s\sin(\phi_s)) \end{aligned} \tag{19}$$

that introduce the single particle longitudinal phase ϕ_s. The maximum value of s occurs when

$$\begin{aligned} \sin(\phi_s) &= 1 \\ \cos(\phi_s) &= 0 \end{aligned} \tag{20}$$

showing that the ellipse has a tilt or slope of

$$\text{slope} = -\frac{\alpha_s}{\beta_s} \tag{21}$$

as illustrated in Figure 1.

2.1 Simple case – erect distributions

In the simplest case there are no offsets and the distribution is erect

$$\langle s\rangle \;=\; \langle\delta\rangle \;=\; \langle s\delta\rangle \;=\; 0 \tag{22}$$

and the longitudinal Twiss functions are defined by analogy to Equation 14

$$\begin{pmatrix} \beta_s \\ -\alpha_s \\ \gamma_s \end{pmatrix} \;=\; \frac{1}{\epsilon_s}\begin{pmatrix} \langle s^2\rangle \\ 0 \\ \langle\delta^2\rangle \end{pmatrix} \tag{23}$$

and the *geometric* RMS longitudinal emittance is

$$\epsilon_s \;=\; \sqrt{\langle s^2\rangle\,\langle\delta^2\rangle} \tag{24}$$

while the *normalized* longitudinal RMS emittance is

$$\epsilon_{s,N} \;=\; \epsilon_s\,\beta_L\gamma_L \tag{25}$$

The normalized emittance is more useful than the geometric when the case of interest involves acceleration, since then $\epsilon_{s,N}$ ideally remains constant.

When the ellipse is erect the RMS bunch length and relative momentum are

$$\begin{aligned} \sigma_s \;&=\; \left\langle s^2\right\rangle^{1/2} \;=\; \sqrt{\beta_s\epsilon_s} \\ \sigma_p \;&=\; \left\langle\delta^2\right\rangle^{1/2} \;=\; \sqrt{\frac{\epsilon_s}{\beta_s}} \end{aligned} \tag{26}$$

the geometric emittance is

$$\epsilon_s \;=\; \sigma_s\sigma_p \tag{27}$$

and the longitudinal beta function

$$\beta_s \;=\; \frac{\sigma_s}{\sigma_p} \tag{28}$$

is just the ratio of bunch length to relative momentum width. Remarkably, the beta function is also given by the simple expression

$$\beta_s \;=\; \frac{C}{2\pi}\frac{|\eta|}{Q_s} \tag{29}$$

where C is the accelerator circumference and Q_s is the synchrotron tune [3, 4]. The slip factor

$$\eta \;=\; \frac{1}{\gamma_T^2} \;-\; \frac{1}{\gamma_L^2} \tag{30}$$

changes significantly as the beam accelerates, unless the transition gamma γ_T is extremely large [2, 3, 4]. Dramatic things clearly happen when γ_L accelerates through γ_T since then $\eta \to 0$, $\beta_s \to 0$, $\sigma_s \to 0$, and $\sigma_p \to \infty$. In fact the ellipse tilts and α_s becomes significant close to transition.

2.2 General case – tilted distributions

In general the longitudinal Twiss parameters $(\beta_s, \alpha_s, \gamma_s)$ are found through the equations

$$\begin{pmatrix} \beta_s \\ -\alpha_s \\ \gamma_s \end{pmatrix} \;=\; \frac{1}{\epsilon_s}\begin{pmatrix} \langle s^2\rangle \\ \langle s\delta\rangle \\ \langle\delta^2\rangle \end{pmatrix} \tag{31}$$

and the geometric RMS longitudinal emittance is

$$\epsilon_s \;=\; \sqrt{\langle s^2\rangle\,\langle\delta^2\rangle - \langle s\delta\rangle^2} \tag{32}$$

by analogy with Equation 15. It continues to be assumed that the first order moments $\langle s \rangle$ and $\langle \delta \rangle$ are zero. In the particular case of a hollow elliptical distribution in which all particles have the same longitudinal action J_s then

$$\epsilon_s \;=\; J_s \tag{33}$$

otherwise

$$\epsilon_s \;=\; \langle J_s \rangle \tag{34}$$

since the geometric emittance is just the average action.

3 Deriving $(\epsilon_s, \beta_s, \alpha_s, \gamma_s)$ from simulation output

Longitudinal dynamics simulation codes rarely work in (s, δ) space. For example, `BLonD` [5] works in $(t, \Delta E)$ space, where t is the arrival time of a single ion relative to a nominal marker – positive if the ion is at the front of a bunch. Thus

$$s \;=\; \beta_L c\, t \tag{35}$$

where c is the speed of light. The energy ΔE is an offset relative to E_0, the nominal *total* energy of an ion

$$E_0 = A m_u c^2 \gamma_L \tag{36}$$

where A is the atomic mass number, γ_L is the Lorentz boost, and the atomic mass unit is

$$m_u = 0.931494 \quad [\mathrm{GeV/c^2}] \tag{37}$$

The relative momentum error then becomes

$$\delta \;=\; \frac{\Delta E}{E_0 \beta_L^2} \tag{38}$$

after using the identity

$$\frac{dE}{dp} \;=\; \beta_L^2 \,\frac{E}{p} \tag{39}$$

Using the transformations of Equations 35 and 38 the first moments of the distribution become

$$\langle s \rangle \;=\; \beta_L c \,\langle t \rangle \tag{40}$$

$$\langle \delta \rangle \;=\; \frac{1}{E_0 \beta_L^2} \,\langle \Delta E \rangle \tag{41}$$

while the second moments become

$$\left\langle s^2 \right\rangle \;=\; \beta_L^2 c^2 \left\langle t^2 \right\rangle \tag{42}$$

$$\langle s\delta \rangle \;=\; \frac{c}{E_0 \beta_L} \,\langle t\, \Delta E \rangle \tag{43}$$

$$\left\langle \delta^2 \right\rangle \;=\; \frac{1}{E_0^2 \beta_L^4} \left\langle \Delta E^2 \right\rangle \tag{44}$$

`BLonD` outputs various values every N turns, enabling the calculation of the moments $\langle s \rangle, \langle \delta \rangle, \left\langle s^2 \right\rangle, \langle s\delta \rangle$, and $\left\langle \delta^2 \right\rangle$ either inside `BLonD`, or in in post-BLonD analysis. Applying Equations 31 and 32 then enables the calculation of $(\epsilon_s, \beta_s, \alpha_s, \gamma_s)$ as a function of time.

3.1 Testing longitudinal values derived from simulation data

Figure 2 illustrates how the signal generator code `sig_gen_long.py` is used to independently test the analysis performed by the code `longitudinal_twiss.py` on data produced by `BLonD`. These two codes are listed in Appendices A and B. Table 1 illustrates a set of input parameters used by `sig_gen_long.py` to generate an initial particle distribution in the form of a hollow ellipse. The **public** parameters must be passed on for

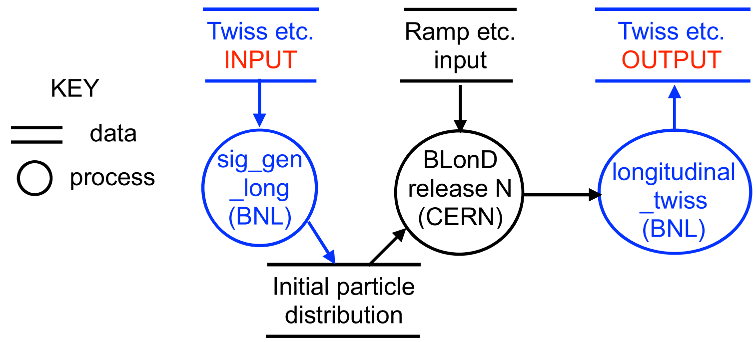

Figure 2: Using the signal generator `sig_gen_long.py` to test the derivation of Twiss values performed by `longitudinal_twiss.py`. Data and processes drawn in blue are developed at BNL, while the data formats and processes in black conform to a standard `BLonD` package released by CERN.

Parameter	Symbol	Units	Test value
Public			
Atomic number	Z	–	79
Atomic mass	A	–	196.97
Energy per nucleon	E_u	GeV/u	10.29
Total energy	E_0	GeV	2018.24
Private			
Geometric emittance	ϵ_s	mm	1.8
Twiss function	β_s	m	1200
Twiss function	α_s	–	-2.5

Table 1: Parameters and typical values used by `sig_gen_long.py` in its generation of an initial hollow elliptical particle distribution that is used by `BLonD`. Action and emittance are initially identical.

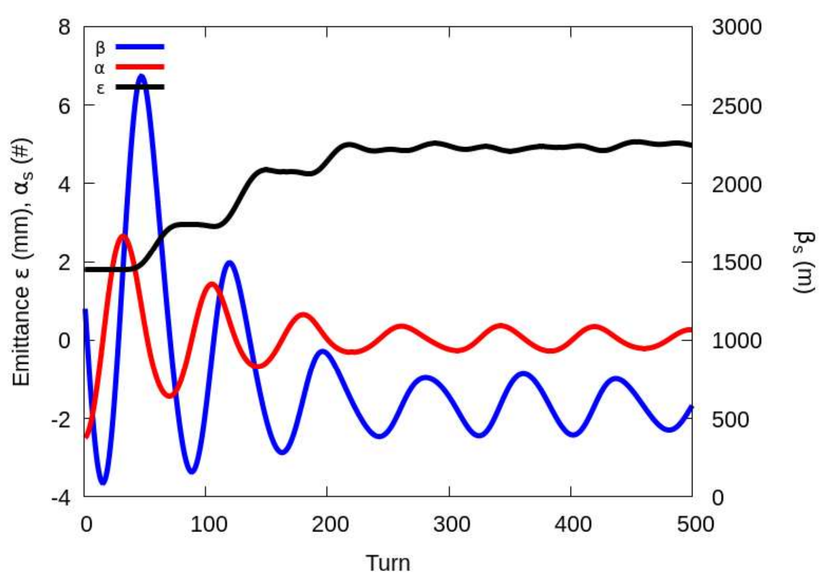

Figure 3: Longitudinal Twiss parameters β_s and α_s and geometric emittance ϵ_s versus turn number during blind testing. The values on turn 1 agree very well with the private parameters listed in Table 1.

BLond and longitudinal_twiss.py to work correctly. In contrast, the private parameters are not passed on, but should be returned correctly by the analysis code. This enables blind testing.

Figure 3 uses Table 1 values to launch an initial test ellipse that is *not* matched to the lattice being simulated. The Twiss parameters oscillate with the period of quadrupole mode oscillations,

$$T_{\rm QMO} \;\geq\; \frac{1}{2Q_s} \;=\; 59 \quad \text{turns} \tag{45}$$

where the (small amplitude) synchrotron tune is

$$Q_s \;=\; 0.0085 \tag{46}$$

Figure 4 compares phase space locations of 5,000 particles initially and after 100,000 turns, extending the results shown in Figure 3. The phase space distribution evolves and filaments as the number of turns increases, due to the mismatched initial conditions and to the decoherence caused by the spread in synchrotron tunes across the RF bucket.

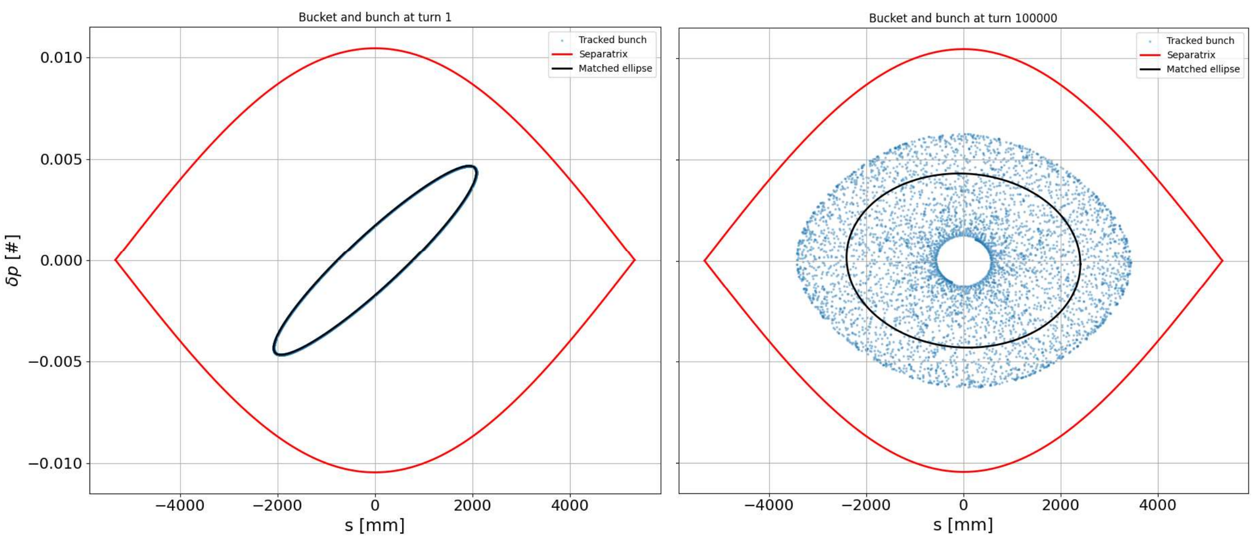

Figure 4: The evolution of an initially mismatched non-accelerating phase ellipse.
LEFT: The initial phase space ellipse on turn 1.
RIGHT: After complete filamentation, on turn 100,000. Each blue dot is a particle, the red curve is the separatrix, and the black curve is the matched one-sigma phase space ellipse.

3.2 Separatrix contour

The separatrix in Figure 4 is a contour of the longitudinal Hamiltonian of the system

$$\mathcal{H}(s,\delta) = \frac{h\,\eta\,\omega_0}{2\,\beta^2}\delta^2 \;+\; \frac{ZV}{2\pi h\,E_0\beta^2}\left[\cos\left(\phi_s + \frac{h\omega_0}{\beta c}\,s\right) - \cos(\phi_s) - \frac{h\omega_0}{\beta c}\,s\cdot\sin(\phi_s)\right] \tag{47}$$

where h is the harmonic number,

$$\omega_0 \;=\; (2\pi c)/C \tag{48}$$

is the revolution angular frequency, E_0 is the nominal total energy of a particle, V is the RF voltage, and the synchronous phase ϕ_s is non-zero if the system is accelerating. The substitution

$$\phi = \phi_s + (h\omega_0)/(\beta c)s \tag{49}$$

simplifies the Hamiltonian. The kinetic term $(h\,\eta\,\omega_0)/(2\,\beta^2)\delta^2$ changes sign with slip factor η as transition is crossed.

Particle motion is unbucketed outside the separatrix, which has a contour of height

$$\mathcal{H}(s,\delta) \;=\; \mathcal{H}(\phi_{\rm ufp},0) \;=\; \mathcal{H}(s_{\rm ufp},0) \tag{50}$$

where the unstable fixed point (ufp) is at

$$\phi_{\rm ufp} = \pi - \phi_s \tag{51}$$

or

$$s_{\rm ufp} = \frac{\beta C}{2\pi h}(\pi - 2\phi_s) \tag{52}$$

The location of the separatrix follows the curve

$$\delta_{sep} \;=\; \pm\sqrt{\frac{2ZVE_0\beta^2}{\pi h|\eta|}}\cdot\sqrt{\cos(\phi_s) - \cos(\phi) + (\phi-\phi_s)\cdot\sin(\phi_s)} \tag{53}$$

where δ_{sep} can have either sign. This is greatly simplified if the beam does not accelerate, in which case ϕ_s is zero and

$$\delta_{sep} \;=\; \pm\sqrt{\frac{2ZVE_0\beta^2}{\pi h|\eta|}}\cdot\sqrt{1-\cos\phi} \tag{54}$$

The small amplitude synchrotron tune is

$$Q_s \;=\; \sqrt{\frac{h|\eta|\cos(\phi_s)ZV}{2\pi\beta^2 E_0}} \tag{55}$$

where

$$0 < \phi_s < \pi/2 \tag{56}$$

if

$$\eta < 0 \tag{57}$$

otherwise the fixed point jumps to $\pi - \phi_s$. The synchrotron tune is

$$Q_s = 0.0085 \tag{58}$$

in the non-accelerating case shown in Figures 3 and 4, with a small amplitude period of about 118 turns.

4 Example: the evolution of $(\epsilon_s, \beta_s, \alpha_s)$ through RHIC transition

`BLonD` [5] is used in a 2-dimensional simulation of $^{197}\text{Au}^{+79}$ ions undergoing a RHIC acceleration ramp over one unit of γ across transition, using the parameters listed in Table 2. Longitudinal motion is non-adiabatic near transition when

$$\Omega \;\equiv\; \frac{1}{\omega_s^2}\left|\frac{d\omega_s}{dt}\right| \;\ll\; 1 \tag{59}$$

where t is time (not turn number), and ω_s is the synchrotron angular frequency (radians per second). The length of this non-adiabatic period – the characteristic time [6, 7] – is given by

$$T_c \;=\; \left(\frac{AE_T}{ZeV|\cos(\phi_s)|}\times\frac{\gamma_T^3}{h\gamma'}\times\frac{C_0^2}{4\pi c^2}\right)^{1/3} \tag{60}$$

where γ' is the acceleration rate

$$\gamma' = d\gamma/dt \tag{61}$$

Figure 5 plots values of α_s (left) and β_s (right) every 1000 turns. The values of α_s and β_s (and so also the bunch length) reach minimum values of about -0.56 and 70.3 m at transition on turn 105,930, at the center of the non-adiabatic span of $\pm$5,511 turns around transition.

Parameter	**Symbol**	**Units**	**Value**
Atomic weight	A		196.97
Atomic number	Z		79
Transition energy	E_T	GeV/u	21.76
RF voltage	V	kV	200
Stable phase	ϕ_s	deg.	3.038
Gamma transition	γ_T		23.367
Harmonic number	h		360
Ramp rate	γ'	s^{-1}	0.357
Circumference	C	km	3.833
Characteristic time	T_c	s	0.071
Initial bunch length RMS	t	ns	3.67

Table 2: Parameters used for a `BLonD` simulation of RHIC transition crossing without a γ_T-jump.

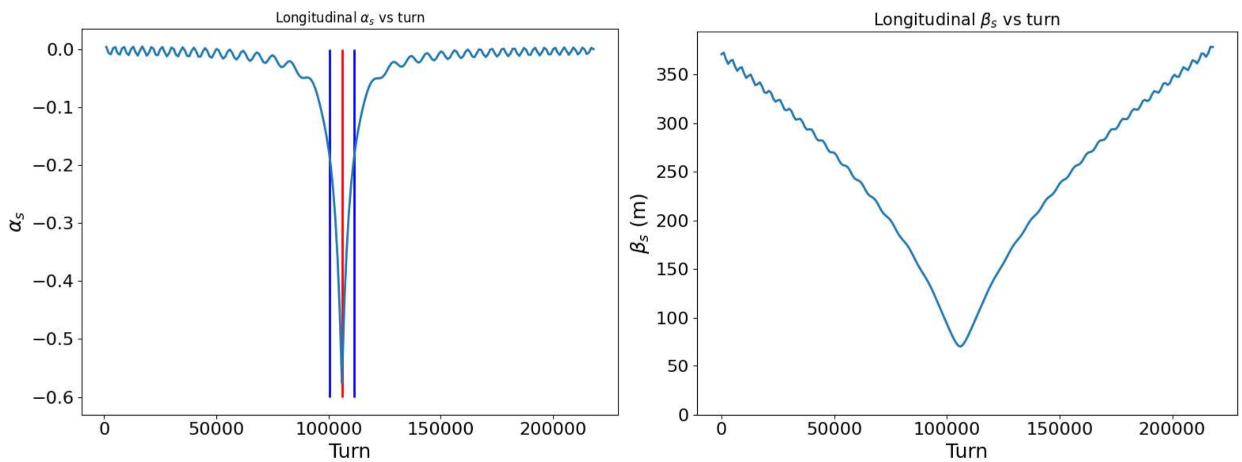

Figure 5: The evolution of α_s (left) and β_s (right) as transition is crossed in RHIC. The vertical red line in the left plot indicates transition, while the span of the vertical blue lines indicates the characteristic time $\pm T_c$ during which motion is non-adiabatic. The plot clearly shows the reduction of the α_s and β_s as transition is approached, and their return to pre-transition values.

Figure 6 shows the evolution of the *normalized* longitudinal emittance defined by Equations 25 and 32, across transition, with a value of

$$\epsilon_{s,N} \;=\; 4.46 \pm 1.00 \text{ mm} \tag{62}$$

while the *geometric* emittance is the mean action of the distribution of particles.

$$\epsilon_s \;=\; \langle J_s \rangle \tag{63}$$

Table 3 lists longitudinal emittances and actions.

Figure 7 illustrates the tilt around transition by plotting the 50,000 particle matched Gaussian distribution T_c before transition, at transition, and T_c after transition. The red ellipses have actions of $3\epsilon_s$. The upper plot shows the distribution one T_c before transition where a slight tilt is shown which corresponds to the change in α_s in the left plot of Figure 5. The tilt is at its maximum at transition, returning to the same tilt one T_c after transition.

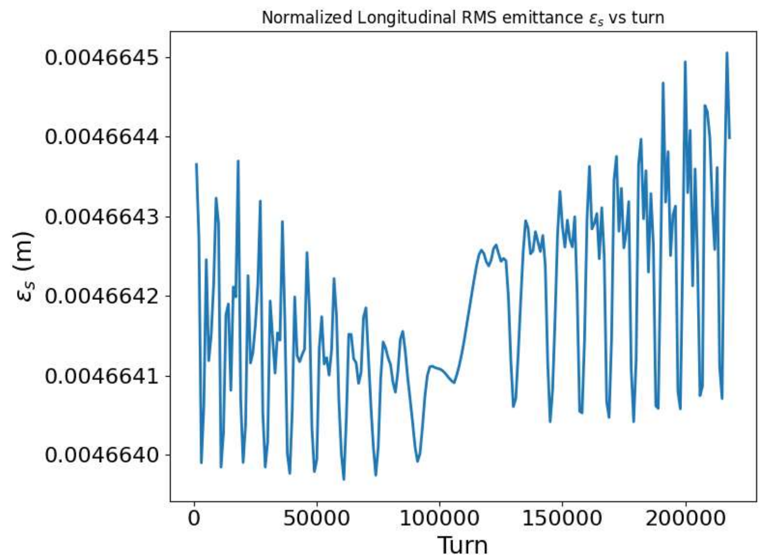

Figure 6: The simulated normalized emittance $\epsilon_{s,N}$ remains nearly invariant during transition crossing.

Attribute	**Symbol**	**Units**	**Value**
Normalized emittance	$\epsilon_{s,N}$	mm	4.68
Geometric emittance	ϵ_s	mm	0.20
Mean particle action	$\langle J \rangle$	mm	0.20
Maximum particle action	J_{max}	mm	2.21

Table 3: Longitudinal emittances and actions in the simulation of transition crossing in RHIC for the distribution shown in Figure 7.

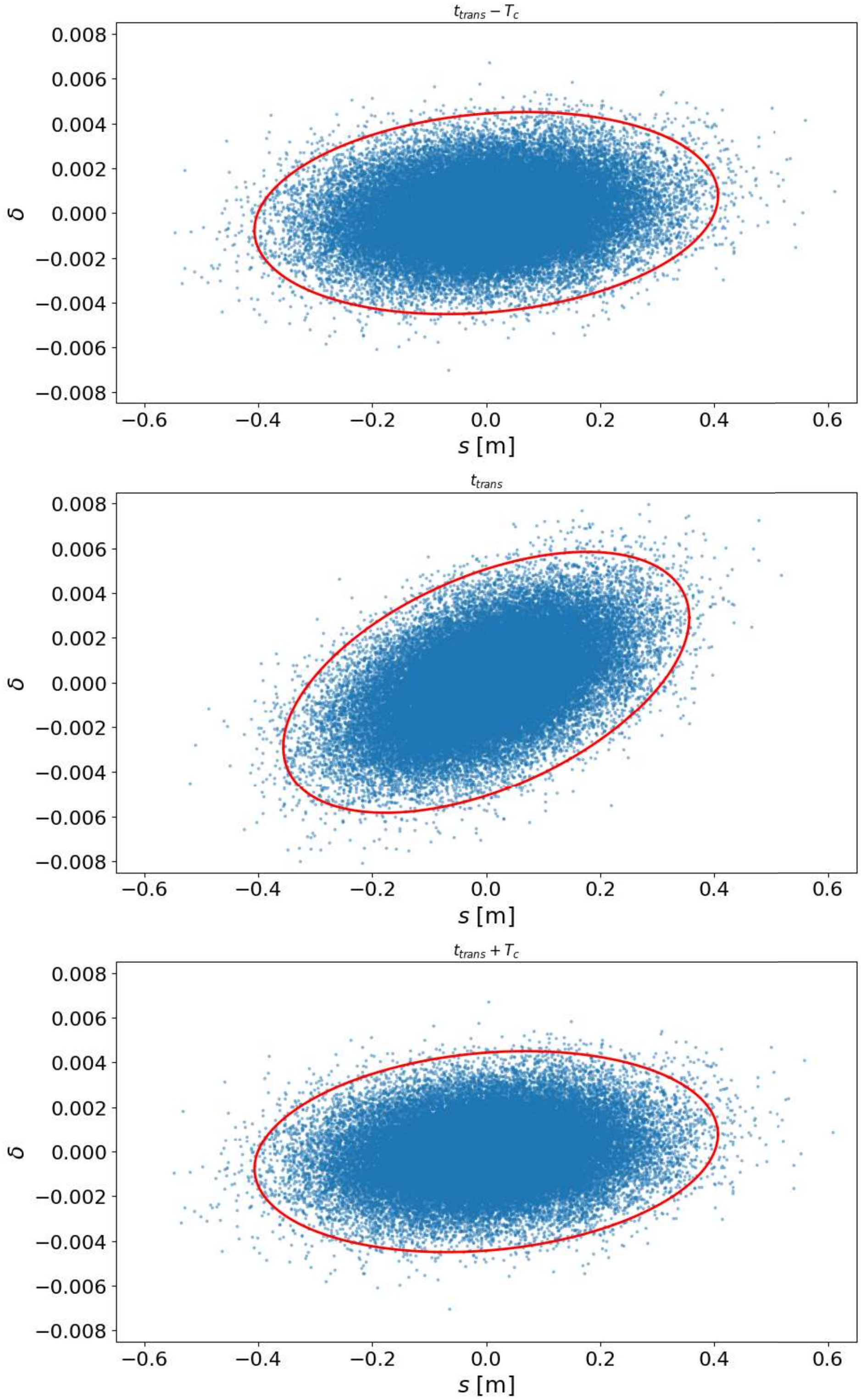

Figure 7: Longitudinal beam distribution evolution in RHIC over a time period of $\pm T_c$ across transition. The nonadiabatic characteristic time T_c is defined in Equation 60.
Top: T_c before transition. Middle: transition. Bottom: T_c after transition.
The red ellipse has an action of $J_s = 3\epsilon_s$ of the beam distribution.

5 Conclusions

Longitudinal Twiss parameters were defined, in direct analogy to transverse Twiss parameters. The longitudinal code `BLonD` was used in tandem with the custom-built analysis code `longitudinal_twiss.py` to simulate the evolution of longitudinal Twiss parameters through transition in RHIC. As expected, the simulation showed tilted ellipses in longitudinal phase space, with a maximum tilt at transition, consistent with previous results [8].

6 Acknowledgments

Many thanks go to Jie Wei, for his insightful discussions.

References

[1] T. Wangler. *RF Linear Accelerators.* Wiley-VCH, 2008.

[2] S. Peggs and T. Satogata. *Introduction to Accelerator Dynamics.* Cambridge University Press, 2017.

[3] S. Peggs and J. Wei. Longitudinal phase space parameters. Technical Report RHIC/AP/106, BNL, 1996.

[4] M. Conte and W.M. Mackay. *An Introduction to the Physics of Particle Accelerators.* World Scientific, 1991.

[5] H. Timko et al. Beam longitudinal dynamics simulation studies. *PRAB*, 26(114602), 2023.

[6] J.-P. Burnet et al. *Fifty years of the CERN Proton Synchrotron: Volume 1.* CERN Yellow Reports: Monographs. CERN, Geneva, 2011.

[7] J. Wei. *Longitudinal Dynamics of the Non-Adiabatic Regime of Alternating Gradient Synchrotrons.* PhD thesis, Stony Brook University, 1990.

[8] H. Lovelace III, K. Drees, S. Peggs, and V. Ptitsyn. Study of longitudinal effects during transition crossing of the EIC Hadron Storage Ring. In *IPAC 24*, number TUPS01, 2024.

Appendices

A The signal generator code sig_gen_long.py

```
#!/Library/Frameworks/Python.framework/Versions/3.10/bin/python3

import math
pi = math.pi
twopi = 2*pi
sqrt = math.sqrt
sin  = math.sin
cos  = math.cos

print(' ')
print('>> Usage: sig_gen_long.py')
print('Last edited by Steve Peggs on March 19, 2026')
print('See output files "long_ellipse.xmgr" and "BLonD_input.txt"')

# Constants ##################################
c     = 2.99792458e8       # m/s
e     = 1.60217662e-19     # C
amu   = 0.931494           # (GeV/c^2)/u
m_p   = 0.938272           #  GeV/c^2

# Ion selection ##############################
Z_ion = 79                 # -
A_ion = 196.97             # -
E_tot = 20.0               # TOTAL ion energy per amu [GeV/u]

# Twiss ellipse parameters #################
J     = 5e-4               # Longitudinal action [m]
beta  = 1000.0             #  [m]
alpha = -1.0               #  [-]
s_off = 0.0                #  [m]
d_off = 0.0                #  [-]

# Sets of complete input parameters
Z_ion = 79; A_ion = 196.97;
E_tot = 20.0; J = 3e-4; beta = 1500.0; alpha = -2.0; s_off = 0.0; d_off = 0.0 # 260318
E_tot = 18.0; J = 6e-4; beta = 1200.0; alpha = -2.5; s_off = 0.0; d_off = 0.0 # 260319
E_tot = 18.0; J = 6e-4; beta = 1200.0; alpha = -2.5; s_off = 0.3; d_off = 0.001 # 260319a

# Calculate
gamma_ion = E_tot/amu
beta_ion  = sqrt(1.0 - 1.0/(gamma_ion * gamma_ion))
bm1       = 1.0 - beta_ion

# Report to terminal
print(' ')
print('Z_ion  [-]    = %10d' % Z_ion)
print('A_ion  [-]    = %10.3f' % A_ion)
print('E_tot [GeV/u] = %10.5f TOTAL ion energy per amu' % E_tot)
print('J      [m]    = %10.3e' % J)
```

```
print('beta   [m]    = %10.5f' % beta)
print('alpha  [-]    = %10.5f' % alpha)
print('s_off  [m]    = %10.5f' % s_off)
print('d_off  [-]    = %10.5f' % d_off)
print(' ')
print('gamma_ion [-] = %9.5f' % gamma_ion)
print('beta_ion  [-] = 1-%7.5f' % bm1)

# Open and prepare xmgrace graphics file ##############################
dvss = open("long_ellipse.xmgr", "w")
dvss.write('@target G0.S0 \n')
dvss.write('@type xy\n')
dvss.write('@subtitle "Longitudinal phase space ellipse"\n')
dvss.write('@xaxis label "Longitudinal displacement,  s [m]"\n')
dvss.write('@yaxis label "Off-momentum parameter,  \\xd\\f{} = \\xD\\f{}p/p  [-]"\n')
dvss.write('@s0 symbol 1\n')
dvss.write('@s0 symbol size 0.1\n')
dvss.write('@s0 symbol fill pattern 1\n')
dvss.write('@s0 line type 0\n')
dvss.write('@s1 symbol 1\n')
dvss.write('@s1 symbol size 0.2\n')
dvss.write('@s1 symbol fill pattern 1\n')
dvss.write('@s1 line type 0\n')

# Open and prepare BLonD input file ##############################
blin = open("BLonD_input.txt", "w")
blin.write('This BLonD input file was generated by "sig_gen_long.py"\n')
blin.write('DE is per nucleon, NOT per ion\n')
blin.write('Z_ion   [-]    = %10d    Atomic number\n' % Z_ion)
blin.write('A_ion   [-]    = %10.3f    Atomic weight\n' % A_ion)
blin.write('E_tot  [GeV/u] = %10.5f    TOTAL energy per nucleon of an ion\n' % E_tot)
blin.write('gamma_ion [-]  = %10.5f    Relativistic gamma\n' % gamma_ion)
blin.write('beta_ion  [-]  =  1-%7.5f    Relativistic beta\n' % bm1)
blin.write('====================================\n')
blin.write('  Arrival time      Energy offset\n')
blin.write('  Delta_t [s]       Delta_E [eV/u]\n')
blin.write('====================================\n')

# Generate data ############################################
angle_count = 100
delta_angle = twopi/angle_count
s_sum  = 0.0
s2_sum = 0.0
d_sum  = 0.0
d2_sum = 0.0
sd_sum = 0.0
for i in range(0,angle_count):
    angle = i*delta_angle
    s = s_off + sqrt(2*J*beta) * sin(angle)
    d = d_off + sqrt(2*J/beta) * (cos(angle) - alpha*sin(angle))
    dvss.write('%10.6f %10.6f\n' % (s, d))
    Dt = s/(beta_ion*c)                    # Front of the bunch has positive s [m] & Dt [s]
    DE = d * (E_tot * beta_ion*beta_ion)  # Energy offset per nucleon [GeV/u]
    DEeV = DE * 1e9                        # Energy offset per nucleon [eV/u]
```

```
    blin.write('%18.15f %15.1f\n' % (Dt, DEeV))
    s_sum  += s
    s2_sum += s*s
    d_sum  += d
    d2_sum += d*d
    sd_sum += s*d
dvss.write('&\n')
dvss.close
blin.close

# Calculate distribution parameters and sizes #################
s_ave  = s_sum  / angle_count
s2_ave = s2_sum / angle_count
d_ave  = d_sum  / angle_count
d2_ave = d2_sum / angle_count
sd_ave = sd_sum / angle_count
sig2_11 = s2_ave - s_ave*s_ave
sig2_12 = sd_ave - s_ave*d_ave
sig2_22 = d2_ave - d_ave*d_ave
s_std = sqrt(sig2_11)
d_std = sqrt(sig2_22)
print(' ')
print('<s>    [m]  = %10.5f' % s_ave)
print('<d>    [-]  = %10.5f' % d_ave)
print('<s^2> [m^2] = %10.5f' % s2_ave)
print('<d^2>  [-]  = %10.5f' % d2_ave)
print('<sd>   [m]  = %10.5f' % sd_ave)
print('s      [m]  = %10.5f +/- %7.5f' % (s_ave, s_std))
print('delta  [-]  = %10.5f +/- %7.5f' % (d_ave, d_std))

# Calculate parameters and cross-check ###################
emit2 = sig2_11*sig2_22 - sig2_12*sig2_12
emit  = sqrt(emit2)
beta_out  =  sig2_11/emit
alpha_out = -sig2_12/emit
gamma_out =  sig2_22/emit
parity = beta_out*gamma_out - alpha_out*alpha_out
print(' ')
#print('emit^2    [m^2]  = %10.3e' % emit2)
print('emit unorm [m]   = %10.3e' % emit)
print('beta_out   [m]   = %10.5f' % beta_out)
print('alpha_out  [-]   = %10.5f' % alpha_out)
print('gamma_out [1/m]  = %10.5f' % gamma_out)
print('Twiss parity     = %10.5f' % parity)
print(' ')
```

B The longitudinal Twiss analysis code longitudinal_twiss.py

```
#!/usr/bin/env python3
from __future__ import division, print_function

import math
import numpy as np
from scipy import constants


def compute_longitudinal_twiss_from_particles(dt, dE, beta, gamma, energy_eV, c=constants.c):

    beta = float(beta)
    gamma = float(gamma)
    E0 = float(energy_eV)

    # ---------------------------------------------
    # Convert particle coordinates
    # ---------------------------------------------
    s = beta * c * dt
    delta = dE / (E0 * beta**2)

    # ---------------------------------------------
    # Compute means directly (CRITICAL FIX)
    # ---------------------------------------------
    s_mean = np.mean(s)
    d_mean = np.mean(delta)

    s_cent = s - s_mean
    d_cent = delta - d_mean

    # ---------------------------------------------
    # Covariance matrix
    # ---------------------------------------------
    sig11 = np.mean(s_cent**2)
    sig22 = np.mean(d_cent**2)
    sig12 = np.mean(s_cent * d_cent)

    # ---------------------------------------------
    # Determinant protection
    # ---------------------------------------------
    det = sig11*sig22 - sig12**2

    if det < 0 and det > -1e-30:
        det = 0.0

    if det < 0:
        return np.nan, np.nan, np.nan, np.nan, np.nan

    # ---------------------------------------------
    # Twiss parameters
    # ---------------------------------------------
    eps_s = beta * gamma * np.sqrt(det)
```

```
    if eps_s == 0:
        return 0, 0, 0, 0, 0

    scale = (beta * gamma) / eps_s

    beta_s  = scale * sig11
    alpha_s = -scale * sig12

    # --------------------------------------------
    # Beam sizes (original coordinates)
    # --------------------------------------------
    sigma_dE = np.std(dE)
    sigma_dt = np.std(dt)

    return eps_s, alpha_s, beta_s, s_mean, d_mean, sigma_dE, sigma_dt

def main():
    dt = []
    dE = []

    gamma = 19.32380
    beta =  math.sqrt(1.0 - 1.0 / (gamma**2))
    A = 196.970
    mass = 183.433337044e9
    energy = gamma *  mass/A
    with open("BLonD_input_260319a.txt", "r") as f:
        for line in f:
            line = line.strip()
            if not line or line.startswith("#"):
                continue
            parts = line.split()
            dt = np.append(dt, float(parts[0]))
            dE = np.append(dE, float(parts[1]))
    eps_s, alpha_s, beta_s, s_mean, d_mean, sigma_dE, sigma_dt = \
         compute_longitudinal_twiss_from_particles(
             dt,
             dE,
             beta,
             gamma,
             energy
         )

    print('Action J (m): ', eps_s/(beta*gamma))
    print('Alpha_s (#): ', alpha_s)
    print('Beta_s (m): ', beta_s)
    print('s_off (m): ', s_mean)
    print('dp/p offset (#): ', d_mean)
if __name__ == "__main__":
    main()
```