


Beam longitudinal dynamics simulation studies

H. Timko, S. Albright, T. Argyropoulos, H. Damerau[✉], K. Iliakis^{✉,*}, L. Intelisano,
 B. E. Karlsen-Baek^{✉,†}, I. Karpov[✉], A. Lasheen[✉], L. Medina[✉],
 D. Quartullo, J. Repond, and A. L. Vanel[✉]
CERN, Geneva 23, 1211, Switzerland

J. Esteban Müller[✉]
European Spallation Source, Lund, Sweden

M. Schwarz
Karlsruhe Institute of Technology, Karlsruhe, Germany

P. Tsapatsaris and G. Typaldos[✉]
National Technical University of Athens, Athens, Greece

 (Received 3 October 2022; revised 4 August 2023; accepted 9 October 2023; published 29 November 2023)

The beam longitudinal dynamics code `BLonD`, utilized tool, has been developed at CERN since 2014. It has emerged as a central tool for conducting longitudinal beam dynamics simulations. In this paper, we present this modular simulation suite and the various physics models that can be included and combined by the user. We detail the reference frame, the equations of motion, the plethora of options for radio-frequency parameters such as phase noise, fixed-field acceleration, and feedback models for the CERN accelerators, as well as the modeling of collective effects and synchrotron radiation. We also present various methods of generating multibunch distributions matched to a given impedance model. `BLonD` is furthermore a well tested and optimized simulation suite, which is demonstrated through examples, too.

DOI: [10.1103/PhysRevAccelBeams.26.114602](https://doi.org/10.1103/PhysRevAccelBeams.26.114602)

I. INTRODUCTION

For several decades, longitudinal beam dynamics simulations at CERN have been performed using Fermilab's widely used simulation suite called `ESME` [1]. With the upgrade projects of the CERN synchrotrons and studies of future machines, there was a growing need for precision simulations that could combine for a given study of all relevant physics with machine-specific features. At the same time, the simulation suite would have to be general enough to cover a wide range of applications, from low- to high-energy synchrotrons, from electrons over protons to ions, from space charge to synchrotron-radiation dominated regimes.

The Beam Longitudinal Dynamics simulation suite `BLonD` [2,3] has been developed at CERN since 2014.

*konstantinos.iliakis@cern.ch

†Also at the Department of Physics, Sapienza Università di Roma, Rome, Italy.

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

It is an open-source particle tracking code for the simulation of longitudinal motion in synchrotrons, written in `PYTHON` and `C++` languages. It relies on some of the most popular and efficient scientific libraries including `Numpy` [4], `Scipy` [5], and `CuPy` [6].

The code uses macroparticles with the same charge-to-mass ratio as the real particles. Real bunch intensities typically vary in the range of 10^8 – 10^{13} particles/bunch, while in simulations, an order of 10^4 – 10^7 macroparticles/bunch are used. `BLonD` has a modular structure that allows the user to model different effects according to their needs. The `BLonD` code's unique feature is that it allows both the coordinates of beam particles and the properties of the radio-frequency (rf) system, that is, its frequency and voltage vector, to evolve in space and time, by tracking them with respect to an external reference “clock,” just as in a real machine. This has the advantage of being able to include several beam- and/or cavity loops, collective effects, etc. when modeling the beam motion. Additional special features of the code are the generation of matched multibunch phase-space distributions with collective effects, rf phase noise and sinusoidal rf phase modulation, and its overall flexibility.

The trustworthiness of the `BLonD` suite has been strengthened through thorough benchmarking [7] and numerous applications in all the CERN existing and future

synchrotrons. The outcome of `BLonD` simulation studies has often guided the baseline choices for machine upgrades [8]. To mention a few challenging applications, the CERN Proton Synchrotron Booster (PSB), for instance, poses its challenges with an injection scheme using constant frequency during a magnetic ramp, with beam-based feedback systems on, strong space charge, and a beam that stretches over the entire ring before getting bunched. The proton synchrotron (PS) is equipped with numerous rf systems used to shape the beam via rf manipulations (e.g., splitting and merging bunches), so simulations including multi-rf systems and beam-loading effects are required to predict the best achievable beam quality when increasing beam intensity. Determining the beam and rf parameters for the post-2021 Super Proton Synchrotron (SPS) requires a meticulously accurate impedance model, which was determined from benchmarks of beam-based measurements against simulation results. In addition, apart from CERN, several other institutes have been using `BLonD` for longitudinal studies [9–12].

In this article, we present the unique features of the `BLonD` code and demonstrate them through application examples. We do this by first describing the methods and underlying equations before providing a concrete use case. The paper is structured as follows: Starting from an overview of the program structure (Sec. II), we provide the reference frame and beam-cavity interactions (Sec. III), we then describe the modulation of rf parameters (Sec. IV), the modeling of

impedance and collective effects (Sec. V), synchrotron radiation and quantum excitation (Sec. VI), global and local feedback models (Sec. VII), over the generation of particle distributions (Sec. VIII), to optimizations (Sec. IX), and finally benchmark techniques (Sec. X).

II. PROGRAM STRUCTURE

The main programming objects that comprise the building blocks of `BLonD` simulations are depicted in Fig. 1. The legend on the right explains the different shapes used. The objects marked as mandatory are present in every `BLonD` simulation, while the remaining ones are only initialized if considered relevant by the user. Objects connected with an arrow indicate that the source object is either input to the target object or updates data stored in the target object. The circled numbers on the top-right corner of each object suggest an initialization sequence. Some optional, data-analysis objects that are responsible for storing and plotting data are not shown in the figure.

The `Ring` object (1) holds all the immutable synchrotron properties, e.g., the circumference, the number of ring segments, etc., which are independent of the rf systems and the beam. The `rf station` object (2) contains the rf parameters for all the rf systems in one ring segment. The `beam` object (3) holds the longitudinal particle coordinates. The `distribution generators` (4)

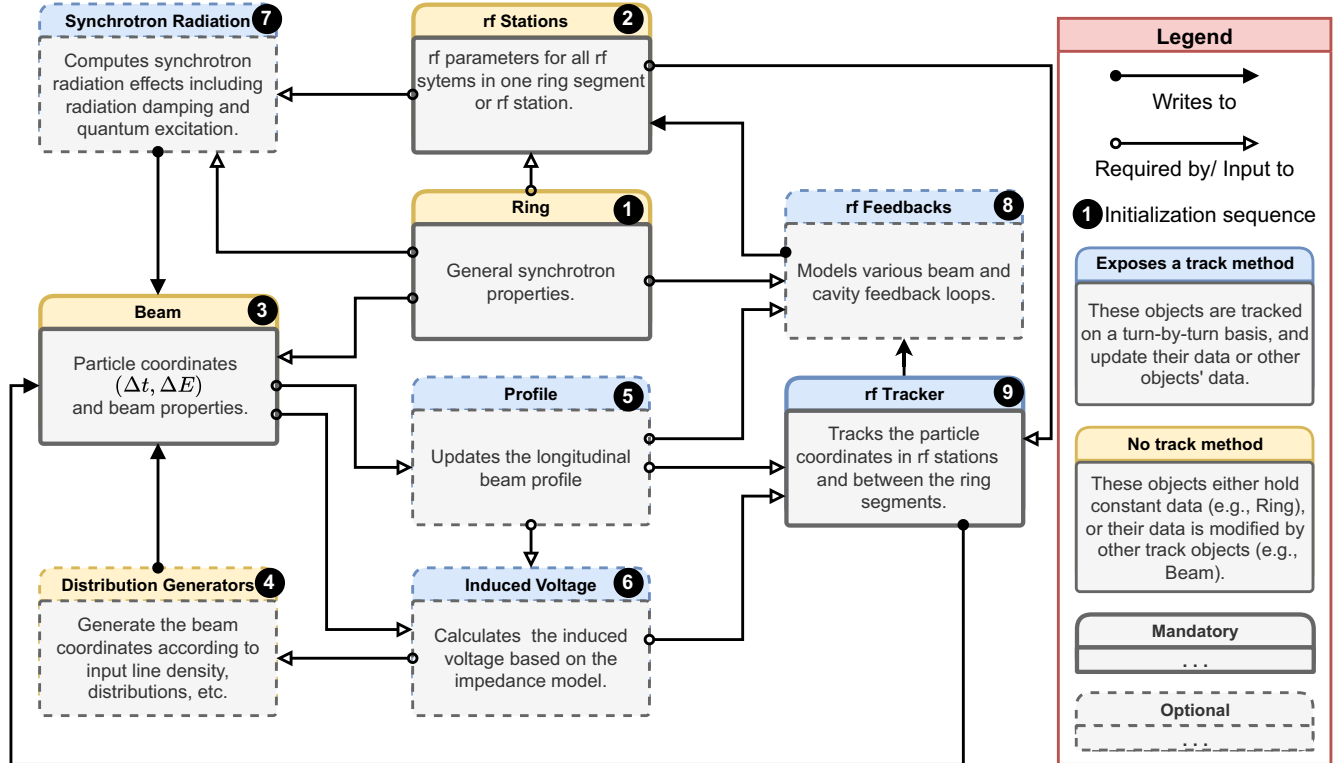


FIG. 1. `BLonD` program structure. This figure displays the main building blocks of `BLonD` simulations, their role, and interactions. The legend on the right explains the function of the different shapes.

initializing the phase-space distribution of the beam are described in detail in Sec. VIII. The `profile` object (5) is responsible for performing a binning [13] operation on the beam to calculate the longitudinal profile. The function of the optional induced voltage (6), synchrotron radiation (7), and rf feedbacks (8) objects is described in Secs. V–VII, respectively. Finally, the rf tracker (9) is responsible for updating the particle coordinates in every rf station and in between them by applying the equations of motion listed in Sec. III.

Following the initialization of the necessary objects, the simulation enters the main tracking loop. Each iteration of the main loop corresponds to one revolution of the beam in the synchrotron. The trackable objects, i.e., those that implement a `track()` method, are updated in every iteration, modifying private data such as the beam profile and the induced voltage, or data stored in remote objects, such as the beam coordinates.

Upon completion of the main tracking loop, statistical data or features are typically stored in files for postprocessing, and some plots are generated to visually inspect the simulation evolution. This marks the end of a BLoND simulation. A collection of illustrative simulation examples can be found in the public BLoND code repository [3].

III. REFERENCE FRAME AND BEAM-CAVITY INTERACTIONS

Just like beam- and cavity control work in a real synchrotron, BLoND describes the evolution of beam particle coordinates and the voltage, phase, and frequency in the radio-frequency accelerating cavities w.r.t. an external reference clock. The design clock time $t_{d,(n)}$ in a given turn n counts the total number of turns elapsed in the laboratory frame,

$$t_{d,(0)} \equiv 0 \quad \text{and} \quad t_{d,(n)} \equiv \sum_{k=1}^n T_{\text{rev.}(k)} \quad \text{for } n \geq 1. \quad (1)$$

The revolution periods $T_{\text{rev.}(n)}$ are defined by the design orbit of radius R_d and $\beta_{d,(n)}$, the relative speed of the design particle with respect to the speed of light c on that orbit,

$$T_{\text{rev.}(n)} = \frac{2\pi R_d}{\beta_{d,(n)} c}, \quad (2)$$

where the user can input $\beta_{d,(n)}$ implicitly via the corresponding design relativistic momentum $p_{d,(n)}$ or total energy $E_{d,(n)}$ evolution over time. The input also determines the design of magnetic field ramp $B_{d,(n)}$ through the relation

$$p_{d,(n)} = |q| \rho B_{d,(n)}, \quad (3)$$

where q is the charge of the particle and ρ the bending radius of the magnets.

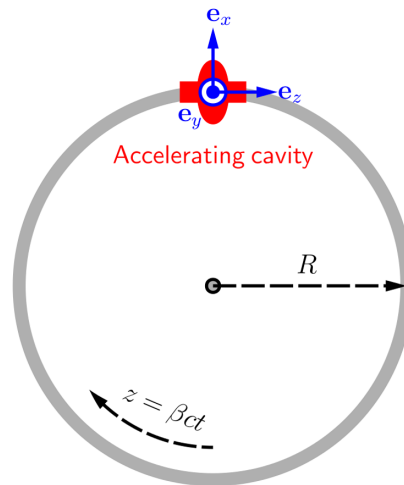


FIG. 2. Reference frame for the equations of motion and the reference time. A given particle on the orbit R is described with the energy E and the longitudinal coordinate t .

The user can choose to place several rf stations along the ring, in which case the magnetic field program of each section of the ring (from one rf station to another) has to be input. This subcyclcling of a turn can be useful, for instance, in the presence of strong synchrotron radiation. In each rf station, an arbitrary number of n_{rf} harmonic rf systems can be modeled; all rf systems in a given rf station will be treated as having the same longitudinal position. The origin of the coordinate system (e_x, e_y, e_z) is fixed to the longitudinal position of the first rf station, on the reference orbit, see Fig. 2.

The particles are described with the phase-space coordinates $(\Delta t_{(n)}, \Delta E_{(n)})$, which are the particle's arrival time and energy with respect to the integrated reference time $t_{d,(n)}$ and the design total energy $E_{d,(n)}$, respectively. For each section of the ring, first, the energy ΔE of a given particle is updated from time step n to $n+1$ based on the particle's arrival time and all the rf voltage kicks k received in the corresponding rf station,

$$\Delta E_{(n+1)} = \Delta E_{(n)} + \sum_{k=1}^{n_{\text{rf}}} q V_{k,(n)} \sin(\omega_{\text{rf},k,(n)} \Delta t_{(n)} + \varphi_{\text{rf},k,(n)}) - (E_{d,(n+1)} - E_{d,(n)}) + E_{\text{other.}(n)}, \quad (4)$$

where V_k is the voltage amplitude, $\omega_{\text{rf},k}$ the rf angular frequency, and $\varphi_{\text{rf},k}$ the phase of the rf system k , and $E_{d,(n+1)} - E_{d,(n)}$ the change of the design energy from one turn to another. The last term $E_{\text{other.}(n)}$ contains additional energy changes due to induced voltage, synchrotron radiation, etc.

The modular structure of the code allows that, on demand, the rf frequency and voltage vector need not be preprogrammed but can evolve dynamically. This evolution is dictated by global and local control loops, as described

later in Sec. VII. When such loops are acting, the rf voltage amplitude and phase can even be arrays instead of single numbers. The choice of the reference frame ensures that these arrays can be sampled at a desired frequency, and modulated to the reference frequency when input into Eq. (4), even if the rf frequency is not an integer multiple of the revolution frequency anymore.

The time coordinate of the particle is updated subsequently, using the already updated energy of the particle and the momentum compaction factor α of at least zeroth, and up to second order,

$$\Delta t_{(n+1)} = \Delta t_{(n)} + T_{\text{rev},(n+1)} \left[\left(1 + \alpha_{0,(n+1)} \delta_{(n+1)} + \alpha_{1,(n+1)} \delta_{(n+1)}^2 + \alpha_{2,(n+1)} \delta_{(n+1)}^3 \right) \frac{1 + \frac{\Delta E_{(n+1)}}{E_{d,(n+1)}}}{1 + \delta_{(n+1)}} - 1 \right], \quad (5)$$

where $\delta_{(n)} = \frac{\Delta p_{(n)}}{p_{d,(n)}} \simeq \frac{\Delta E_{(n)}}{\beta_d^2 E_{d,(n)}}$ is the relative momentum offset.

The collective effects contained in the $E_{\text{other},(n)}$ term of Eq. (4) can in general depend on the particle's coordinates $E_{\text{other},(n)}(\Delta t_{(n)}, \Delta E_{(n)})$ and the coordinates of other particles as well. Without this term, the one-turn mapping of the coordinate pair $\mathbf{x} \equiv (\Delta t_{(n)}, \Delta E_{(n)})$ to $\mathbf{y} \equiv (\Delta t_{(n+1)}, \Delta E_{(n+1)})$ can be described by rewriting Eqs. (4) and (5) as

$$\Delta t_{(n+1)} = \Delta t_{(n)} + f[\Delta E_{(n+1)}(\Delta t_{(n)}, \Delta E_{(n)})], \quad (6)$$

$$\Delta E_{(n+1)} = \Delta E_{(n)} + g(\Delta t_{(n)}). \quad (7)$$

Using the chain rule, the mapping or Jacobian matrix, defined as $\mathcal{M}_{ij} \equiv \partial y_i / \partial x_j$, becomes

$$\mathcal{M} = \begin{pmatrix} 1 + f'g & f' \\ g' & 1 \end{pmatrix}. \quad (8)$$

It is straightforward to show that \mathcal{M} fulfills the symplecticity condition [14]

$$\mathcal{M}^T \mathcal{S} \mathcal{M} = \mathcal{S}, \quad (9)$$

where $\mathcal{S} \equiv \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$. The equations of motion, without any collective effects, are therefore symplectic, and, as a consequence, preserve the phase-space area of the coordinates, despite the rather complex form of Eq. (5). Some solvers are not symplectic and lead therefore to unphysical blowup in simulations. In general, however, when other terms are introduced, like synchrotron radiation or multiparticle effects such as diffusion due to rf noise or collective effects, where

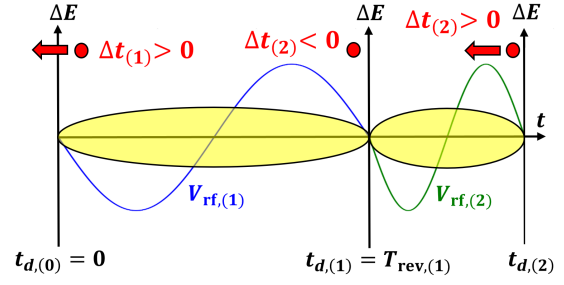


FIG. 3. Scheme illustrating how the periodicity algorithm operates for a particle that drifts away from the current time frame crossing the left border.

the entire N -particle system of equations needs to be considered, symplecticity can be broken.

A. Periodicity

The equations of motion provided in the preceding section lack periodicity in the time coordinate. Consequently, in scenarios where a particle remains outside the rf bucket without experiencing any acceleration, it may drift away toward infinitely large time coordinates which neglects the ring's geometry and is unrealistic. To address this limitation, the BLonD code supports adding the periodicity condition to the equations of motion.

To illustrate how the periodicity algorithm operates, let us suppose that a particle is outside the rf bucket at turn 1 and about to drift away from the present time frame, as shown in Fig. 3. This implies that $\Delta t_1 > 0$ and $\Delta t_2 < 0$, respectively, before and after the equations of motions are applied to the particle. Since $\Delta t_2 < 0$, the particle is ahead relative to the time frame of turn 2, therefore, the particle has to be kicked in energy and drifted one more time, using the rf wave (blue) of turn 1. In order to do that, we first need to represent the time coordinate of the particle relative to the previous time frame, i.e., $\overline{\Delta t_1} = \Delta t_2 + T_{\text{rev},(1)}$. Then, the equations of motion with the rf wave of turn 1 are again applied. As a result, the particle has now $\Delta t_2 > 0$ and it is inside the time frame of turn 2, as desired. The particle will again drift to the left, turn after turn, and the procedure will be repeated whenever $\Delta t < 0$.

On the contrary, if a particle drifts to the right and is about to cross the boundary at turn 1, then at turn 2, the particle will be late relative to the current time frame, therefore the equations of motion will not be applied at turn 2. They will be applied at turn 3, after having represented the time coordinate of the particle with respect to the time frame of turn 3, i.e., $\Delta t_3 = \Delta t_2 - T_{\text{rev},(2)}$.

There are specific cases where the ring periodicity has to be considered in order to describe accurately the dynamics of the particles. One example is a single bunch covering the entire accelerator circumference, with the head and the tail of the bunch crossing the boundaries between previous, present, and next turns. Such a scenario can occur for

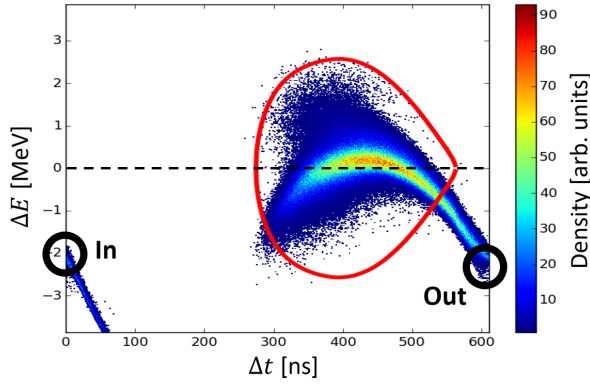


FIG. 4. Phase-space plot of an unstable bunch in an accelerating bucket ($h = 1$). The time coordinate on the horizontal axis spans from 0 to T_{rev} . With the periodic condition enabled, particles that exit the time frame from the right enter back into the frame from the left.

instance in the PSB operated at the $h = 1$ harmonic (Fig. 4), where the ring periodicity ensures that all the particles remain synchronized to the present time frame.

The periodicity algorithm is also useful when the beam losses have to be computed accurately or when an unbunched beam has to be captured inside an rf bucket. Another useful application is when parts of the rf bucket cross the lines $\Delta t = 0$ and $\Delta t = T_{\text{rev}}$ in phase space, for example, when the beam phase loop changes the design rf frequency and shifts the bucket in phase space, or when a second rf system with relatively high voltage is added in bunch-lengthening mode to the main-harmonic voltage during acceleration. In these cases, the bunch will be numerically split in phase space into two portions, which, however, in practice behave as a whole.

While the implemented periodicity feature has demonstrated notable improvements in accuracy for specific simulation scenarios, it is important to note that the algorithm for handling periodicity with full consistency regarding collective effects is presently unavailable in BLoND. This limitation may have a notable impact on simulations involving unbunched beams with collective effects.

IV. MODULATION OF RF PARAMETERS

BLoND provides the possibility of simulating any complex beam manipulation in the longitudinal plane since both the momentum and the rf programs $[\omega_{\text{rf},k}(t), V_k(t), \varphi_{\text{rf},k}(t)]$ can be given as an input. The choice of whether to modulate the rf phase or frequency is in theory equivalent, as the modulated phase offset $\Delta\varphi_{\text{rf}}$ can be related to the frequency offset $\Delta\omega_{\text{rf}}$ as follows:

$$\Delta\varphi_{\text{rf}} = 2\pi h \frac{\omega_{\text{rf}}}{\omega_{\text{rf},d}} = 2\pi h \frac{\omega_{\text{rf},d} + \Delta\omega_{\text{rf}}}{\omega_{\text{rf},d}}. \quad (10)$$

In practice, however, depending on the hardware in which the modulation is applied in, one or the other might be preferred. In addition to rf phase or frequency modulation, whenever $\omega_{\text{rf}} \neq h\omega_{\text{rev}}$, a similar phase shift applies w.r.t. the reference frame chosen:

$$\Delta\varphi_{\text{rf}} = 2\pi h \frac{\omega_{\text{rf}}}{h\omega_{\text{rev}}}. \quad (11)$$

An example of the latter is a magnetic ramp with fixed rf frequency, as it has been suggested for ion beam operation in the SPS [15].

Below, we show some examples of rf phase noise and modulation, as well as slip stacking. However, many other applications, such as cogging, synchronization, fixed-frequency acceleration, etc. are possible to simulate, too.

A. Rf phase noise and modulation

For controlled longitudinal emittance blowup [16,17], both band-limited rf phase noise of the main harmonic [18,19] and single-frequency modulation of a high harmonic [20] can be used. In BLoND, a turn-by-turn phase offset ($\Delta\varphi_{\text{rf},(n)}$) can be added to the programmed rf phase ($\varphi_{\text{rf},(n)}$) to achieve this. The modulation functions can be defined by the user directly or calculated with built-in functions. For instance, the generation method of the phase noise presently used in the CERN synchrotrons [21] can be applied.

In the case of time-dependent single-frequency $f_{\text{mod}}(t)$ modulation, $\Delta\varphi_{\text{rf},(n)}$ is computed as

$$\Delta\varphi_{\text{rf},(n)} = A \sin\left(2\pi \sum_{k=0}^n f_{\text{mod},(k)} T_{\text{rev},(k)}\right) + \varphi_{\text{off},(n)}, \quad (12)$$

where A is the modulation amplitude, $f_{\text{mod},(n)}$ is the modulation frequency, and $\varphi_{\text{off},(n)}$ is an offset about which the modulation is applied. To correctly simulate the phase modulation, the rf frequency has to be corrected by the equivalent frequency change w.r.t. the reference clock:

$$\Delta\omega_{\text{rf},(n)} = \frac{d\Delta\varphi_{\text{rf},(n)}}{dt} = \delta\Delta\varphi_{\text{rf},(n)} \frac{\omega_{\text{rf},d,(n)}}{2\pi h}, \quad (13)$$

where $\delta\Delta\varphi_{\text{rf},(n)}$ denotes the change of $\Delta\varphi_{\text{rf},(n)}$ from one turn to another.

B. Fixed-field manipulations: Slip stacking

Slip-stacking is an rf manipulation technique initially proposed at CERN for the PS and SPS accelerators [22,23], and has since then been studied and applied in various particle accelerator laboratories, including Fermilab [24,25]. Momentum slip-stacking (MSS) stands out as one of the most intricate rf manipulations, currently

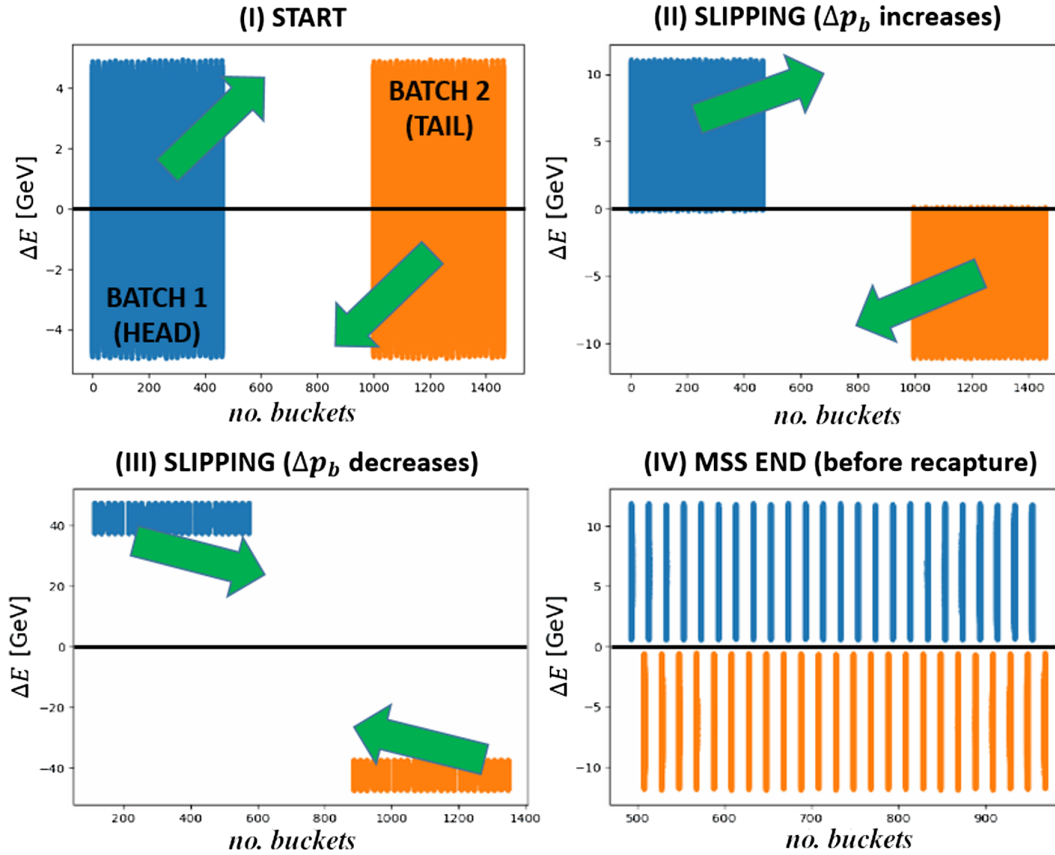


FIG. 5. Illustration of the MSS procedure simulated in BLonD. Two batches, starting from Phase I (top left) are separated in energy and move in the longitudinal phase space relative to each other. The black line marks $\Delta E = E - E_d = 0$. In Phase II (top right), the energy distance between the batches increases, while the opposite occurs in Phase III (bottom left). Recapture is done in Phase IV (bottom right).

undergoing rigorous investigation through simulations for ion beams in the SPS [8] using the BLonD code.

MSS allows two high-energy particle beams with slightly different momenta to slip azimuthally, relative to each other, within the same beam pipe. The two beams are captured by two rf systems with a small frequency difference between them. Each beam is synchronized with one rf system while being perturbed by the other. Upon reaching the desired azimuthal position, the full beam is recaptured with significantly higher rf voltage at the design rf frequency. In particular, for the SPS, two batches of 24 bunches, spaced by 100 ns, are interleaved on an intermediate momentum plateau to produce a single batch of 48 bunches with half the bunch distance (50 ns). The process as simulated in BLonD, is schematically illustrated in Fig. 5. Further details on how MSS is going to be applied in the SPS can be found in [26,27].

During MSS the magnetic field B_d will be constant, which means that, in a first-order approximation, the following relation holds

$$\frac{\Delta\omega_{\text{rf}}}{\omega_{\text{rf},d}} = -\eta_0 \frac{\Delta p}{p_d}, \quad (14)$$

where $\Delta\omega_{\text{rf}}$ and Δp are the changes in the rf angular frequency and beam momentum with respect to their design values. In a reference frame that is synchronized with the design revolution period $T_{\text{rev},d}$, a variation $\Delta\omega_{\text{rf}}$ implies a change in the rf phase according to

$$\Delta\varphi_{\text{rf}} = \frac{2\pi h \Delta\omega_{\text{rf}}}{\omega_{\text{rf},d}}. \quad (15)$$

Providing as an input to BLonD the aforementioned programs, as well as the rf voltage amplitude programs for the two cavity groups, the total rf voltage experienced by each beam is given by

$$V_{\text{rf}} = V_{\text{rf},1} \sin(\omega_{\text{rf},1}t + \varphi_{\text{rf},1}) + V_{\text{rf},2} \sin(\omega_{\text{rf},2}t - \varphi_{\text{rf},2}), \quad (16)$$

where the indices 1,2 indicate the first and second group of rf cavities accordingly.

C. Frequency modulation

In the LEIR accelerator, ion beams are injected and cooled on a long flat bottom. The cooled beams are then

adiabatically captured by the rf and accelerated. For high-intensity beams, frequency modulation is used during capture to increase the longitudinal emittance and improve reproducibility [28]. In this instance, the required $\Delta\omega_{\text{rf}}$ is computed and provided as input to BLonD, from which $\Delta\phi_{\text{rf}}$ is computed via Eq. (15).

V. MODELING IMPEDANCE AND COLLECTIVE EFFECTS

One of the main focuses of the BLonD code is to simulate collective effects due to beam-coupling impedance leading to beam instabilities [29]. The criteria for the development were to include the possibility of exploring stationary single bunch effects such as potential-well distortion, synchronous phase and synchrotron frequency shifts, as well as bunch lengthening [30]. For instance, a synchrotron frequency shift may affect the performance of a controlled longitudinal emittance blow-up [31]. On the other hand, another prerequisite was to include multibunch effects and long-range wakefields, to study coupled-bunch instabilities among other applications. The architecture of the code is built to account for all single and multibunch effects. Moreover, the choice of the coordinate system with the notion of “design” revolution period allows to perform consistent simulations between collective effects and the feedback systems described in Sec. VII.

The electromagnetic interaction is described by the wake function $W(t)$, which represents the electric field excited by a point charge as experienced by a test charge.

In BLonD, to reduce the computational complexity of the particle-particle interactions, a Particle-In-Cell (PIC) type solver is implemented [13]. A binning is applied to the particle distribution to compute the wake potential or induced voltage $V_{\text{ind}}(\Delta t)$ for each cell. This induced voltage is defined as the convolution of the line density $\lambda(t)$ of the beam, normalized to $\int_{-\infty}^{+\infty} \lambda(t) dt = 1$, and the wake function:

$$V_{\text{ind}}(\Delta t) = -qN_p \int_{-\infty}^{+\infty} \lambda(\tau) W(\Delta t - \tau) d\tau, \quad (17)$$

where N_p is the number of real particles in the beam.

The energy kick $E_{\text{ind}}(\Delta t) = qV_{\text{ind}}(\Delta t)$ due to the induced voltage entering the term E_{other} in Eq. (4), where the minus sign in Eq. (17) ensures an energy loss for a positive wake potential.

The induced voltage can also be computed in the frequency domain using the impedance $Z(\omega)$, defined as the Fourier transform of the wake function:

$$Z(\omega) = \int_{-\infty}^{+\infty} W(t) e^{-j\omega t} dt, \quad (18)$$

$$V_{\text{ind}}(\Delta t) = -\frac{qN_p}{2\pi} \int_{-\infty}^{+\infty} Z(\omega) \Lambda(\omega) e^{j\omega\Delta t} d\omega, \quad (19)$$

where $\Lambda(\omega)$ is the beam spectrum, obtained as the Fourier transform of the line density.

Although both time- and frequency-domain methods are equivalent, the discretization required for numerical simulations makes each method suitable for different situations, depending on the bandwidth of the impedance source.

Wakefields corresponding to narrow-band impedance sources require very high-frequency resolution to be able to resolve them correctly. However, in the time domain, we only need to describe the signal for a time window equivalent to the length of a bunch or a train of bunches. In that case, computations in the time domain will require significantly less resources. Alternatively, a broadband impedance will result in a wakefield that would require too fine resolution in the time domain.

For these reasons, BLonD implements both time and frequency-domain calculations, as well as different strategies to deal with the discretization of the line density.

A. Impedance sources

Wake functions and impedances can be calculated analytically for simple geometries (see, e.g., [32]) or using numerical codes for more complicated devices.

BLonD includes several analytical impedance models. One of them is the resonator model, with an impedance defined as

$$Z(\omega) = \frac{R_s}{1 + jQ\left(\frac{\omega}{\omega_r} - \frac{\omega_r}{\omega}\right)}, \quad (20)$$

where R_s is the shunt impedance, ω_r the angular resonant frequency, and Q the quality factor. Alternatively, the resonator wakefield can be used, which is defined as

$$\begin{aligned} W(t > 0) &= 2\alpha R_s e^{-\alpha t} \left(\cos \tilde{\omega} t - \frac{\alpha}{\tilde{\omega}} \sin \tilde{\omega} t \right), \\ W(0) &= \alpha R_s, \end{aligned} \quad (21)$$

where $\alpha = \frac{\omega_r}{2Q}$ and $\tilde{\omega} = \sqrt{\omega_r^2 - \alpha^2}$. The user can include one or more resonators to model the impedance of an element.

Furthermore, BLonD contains a model for the resistive wall impedance of the beam pipe, for the case of a cylindrical beam pipe, which in the frequency domain reads as [29]

$$Z(f) = \frac{Z_0 c L}{\pi [1 - i \operatorname{sgn} f] 2bc \sqrt{\frac{\sigma_c Z_0 c}{4\pi |f|} + i 2\pi b^2 f}}, \quad (22)$$

where Z_0 is the characteristic vacuum impedance, L the length, b the radius and σ_c the conductivity of the beam pipe.

Traveling wave cavities are implemented with the impedance of [33]

$$Z = Z_+ + Z_-, \quad \text{with}$$

$$Z_{\pm}(f) \equiv R_s \left[\left(\frac{\sin \frac{\tilde{\omega}_{\pm}\tau}{2}}{\frac{\tilde{\omega}_{\pm}\tau}{2}} \right)^2 \mp 2i \frac{\tilde{\omega}_{\pm}\tau - \sin \tilde{\omega}_{\pm}\tau}{(\tilde{\omega}_{\pm}\tau)^2} \right], \quad (23)$$

and the wakefield of

$$W(0 < t < a) = \frac{4R_s}{\tau} \left(1 - \frac{t}{\tau} \right) \cos \omega_r t,$$

$$W(0) = \frac{2R_s}{\tau}, \quad (24)$$

where τ is the cavity filling time and $\tilde{\omega}_{\pm} = \omega \pm \omega_r$. Coherent synchrotron radiation can be modeled as an impedance, too. If the user does not give a value for the height of the beam pipe, the impedance of the free-space model is used [34]

$$\frac{\Re Z}{Z_0} = \frac{\sqrt{3}\gamma}{4} \frac{f}{f_{\text{crit}}} \int_{f/f_{\text{crit}}}^{\infty} K_{5/3}(y) dy, \quad (25)$$

$$\frac{\Im Z}{Z_0} = \frac{-\gamma f}{f_{\text{crit}}} \left[\int_0^1 e^{-fy/f_{\text{crit}}} \left(\frac{-2\sqrt{1-y^2}}{4y\sqrt{1-y^2}} \right. \right.$$

$$\left. \left. + \frac{(iy + \sqrt{1-y^2})^{5/3} + (iy + \sqrt{1-y^2})^{-5/3}}{4y\sqrt{1-y^2}} \right) dy \right.$$

$$\left. - \int_1^{\infty} e^{-fy/f_{\text{crit}}} \left(\frac{8\sqrt{y^2-1}}{8y\sqrt{y^2-1}} \right. \right.$$

$$\left. \left. - \frac{(y + \sqrt{y^2-1})^{5/3} - (y + \sqrt{y^2-1})^{-5/3}}{8y\sqrt{y^2-1}} \right) dy \right], \quad (26)$$

where the critical frequency is given by

$$f_{\text{crit}} = \frac{3}{4\pi} \gamma^3 \frac{c}{R}, \quad (27)$$

and R denotes the bending radius of the dipole magnet. If the chamber height is given, the parallel-plates impedance is used [34]. Since BLonD uses the convention that particles leading the reference particle have $\Delta t < 0$, Eq. (26) has an additional minus sign w.r.t. [34].

For the special case of purely imaginary impedances, another method is available, in which the magnitude of the impedance is directly proportional to the frequency. In this

case, the impedance is described by a constant number, Z/n , where $n = \omega/\omega_{\text{rev}}$. The induced voltage computation is much simpler, as the DFTs are replaced by a derivative:

$$V_{\text{ind}}[n] = -\frac{qT_s}{2\pi T_s n} \frac{Z d\lambda[n]}{dn}, \quad (28)$$

where T_s is the binning size for the discretized line density. An interesting use of this method is the modeling of space charge as a reactive impedance (see, e.g., [35]).

Finally, BLonD allows to input also tables that consist of either time and wakefield values, or frequency and impedance values, which can be used to describe an arbitrary impedance model. All of the above-mentioned impedance models can be used in a combined manner as well.

B. Induced voltage calculation

BLonD uses the impedance sources described above to calculate the induced voltage that is then applied to the macroparticles in the same way as the voltage kick from the accelerating cavities. Both operations can be combined for optimization by summing the rf voltage to the induced voltage with the same time resolution. The total energy kick is then linearly interpolated and applied based on the particle position Δt .

BLonD implements two generic algorithms to compute the induced voltage that works with all the impedance sources described above; one is in the time domain and the other in the frequency domain.

In the frequency domain, the method consists of discretizing Eq. (19) as

$$V_{\text{ind}}[n] = -q N_p \text{IDFT}(Z[k]\Lambda[k]), \quad (29)$$

where IDFT is the Inverse Discrete Fourier Transform and $\Lambda[k]$ is the Discrete Fourier Transform (DFT) of the line density: $\Lambda[k] = \text{DFT}(\lambda[n])$.

In time domain, the induced voltage is calculated as a discrete convolution. However, it is in general more efficient to compute the discrete convolution using the DFT according to the circular convolution theorem,

$$V_{\text{ind}}[n] = -q N_p \text{IDFT}\{\text{DFT}(W[n])\text{DFT}(\lambda[n])\}. \quad (30)$$

It is important to carefully pad the two signals with zeros so that the result is the linear convolution. If the length of $W[n]$ is N and the length of $\lambda[n]$ is M , both signals need to be padded so that their length is at least $L = N + M - 1$; in practice, the next regular number is used for runtime efficiency. In this case, the complexity of the algorithm using Fast Fourier Transform (FFT) is $O(L \log L)$, to be compared to $O(NM)$ using the direct convolution algorithm in the time domain.

Given the similarities between Eqs. (29) and (30), BLonD has a single implementation for both methods, with the

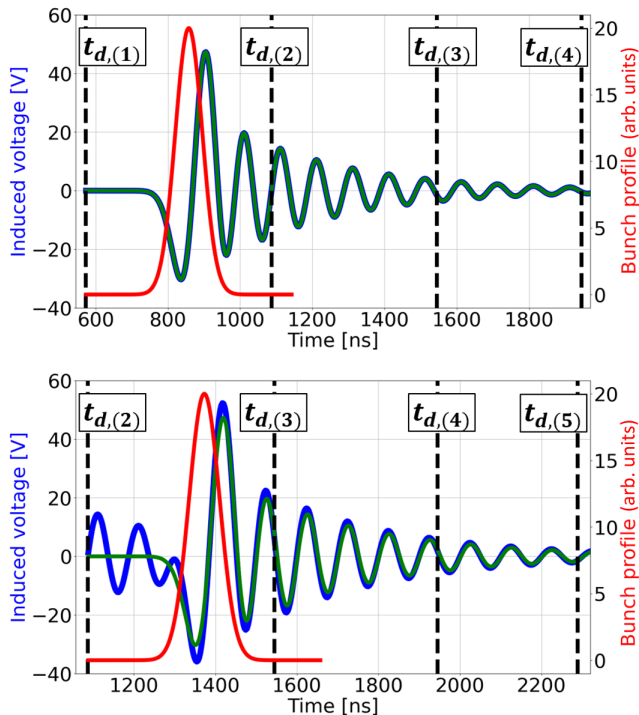


FIG. 6. Example of multiturn induced voltage at turn 1 (top) and turn 2 (bottom) for a Gaussian line density (red) and a resonator impedance. The total induced voltage (blue) is the sum of the induced voltage computed at the current turn (green) and the memory induced voltage coming from the previous turn (not shown). The vertical dashed lines mark the design clock times.

difference that when doing time-domain calculations, a pseudo-impedance is defined as $Z^*[k] = \text{DFT}(W^*[n])$, where the * indicates that the signal is zero padded. Similarly, the beam spectrum is defined as $\Lambda^*[k] = \text{DFT}(\lambda^*[n])$.

BLonD also implements two special algorithms that are limited to resonator impedances. The MUSIC algorithm [36] uses a propagation matrix to compute the induced voltage without binning. The second algorithm is an adaptation of a semianalytic method [37] to resonators and does not require uniform binning.

BLonD can also take into account wakefields lasting more than one turn, which can be an important contribution of narrow-band impedance sources in small rings. The code keeps in memory the induced voltage generated for a user-defined number of preceding turns, which are shifted in time after every turn and added to the induced voltage of the current turn. When the revolution frequency is not constant, a linear interpolation is done to compute the induced voltage with the right binning. An example of multiturn induced voltage at two consecutive turns is given in Fig. 6, where a fixed Gaussian line density interacting with a resonator impedance was assumed. In the first turn (top plot) the induced voltage is computed while the memory is initialized to zero. Therefore, the current induced voltage (green) coincides with the sum (blue) of the memory and current induced voltages, effectively

applied to the particles. After one turn (bottom plot), the memory array is shifted in time by the revolution period $T_{\text{rev},(1)} = t_{d,(2)} - t_{d,(1)}$. The memory of one of the previous turn in the interval $[t_{d,(2)}, t_{d,(3)}]$ is summed to the induced voltage from the current turn (green) to form the total induced voltage (blue). This operation is iterated turn after turn. By having a memory array sufficiently long to cover the decay of all impedance sources, multiturn including transient effects due to the motion of bunches are taken into account. Note that in this example T_{rev} decreases significantly from turn to turn to represent acceleration, which the algorithm takes into account. The memory induced-voltage is shifted by the current T_{rev} at each turn before being added to the current induced-voltage.

All these algorithms can be combined to represent a full machine impedance model, using each of them for different impedance sources depending on their characteristics. BLonD finally calculates the induced voltage as the sum of the induced voltage from each impedance source, which is then applied to the macroparticles.

VI. SYNCHROTRON RADIATION AND QUANTUM EXCITATION

BLonD is not only used for studying the beam dynamics of existing accelerators but also for designing future machines. The Future Circular Collider (FCC) project considers two main options that can be installed in a tunnel of about 100 km: a lepton machine (FCC-ee) operating with up to 365 GeV collision energy [38], and a 100 TeV hadron machine (FCC-hh) [39]. For both of them, synchrotron radiation becomes an essential part of the beam dynamics and needs to be included in macroparticle simulations. This is also true for light sources, which use electron beams to provide x rays to user experiments.

Synchrotron radiation is implemented by adding three additional terms to the energy equation Eq. (4) in the term $E_{\text{other},(n)}$, namely,

$$E_{\text{other},(n)} = -U_0 - \frac{2}{\tau_z} \Delta E_{(n)} + 2 \frac{\sigma_{\Delta E}}{\sqrt{\tau_z}} E_{d,(n)} r, \quad (31)$$

where the first term U_0 accounts for the average energy loss per particle over a turn, the second term describes the difference in energy loss for each particle, and the last term models quantum excitation [40]. Here, τ_z denotes the damping time (in units of turns), $\sigma_{\Delta E}$ the equilibrium energy spread, and r is a number drawn from a normal distribution of zero mean and unit variance. The energy loss U_0 is proportional to the fourth power of the design energy. For the FCC-ee \bar{t} operation, this corresponds to losing more than 5% of the beam energy at every turn (for each beam $E_d = 182.5$ GeV and $U_0 = 9.2$ GeV). In reality, the energy loss occurs progressively along the ring. However in

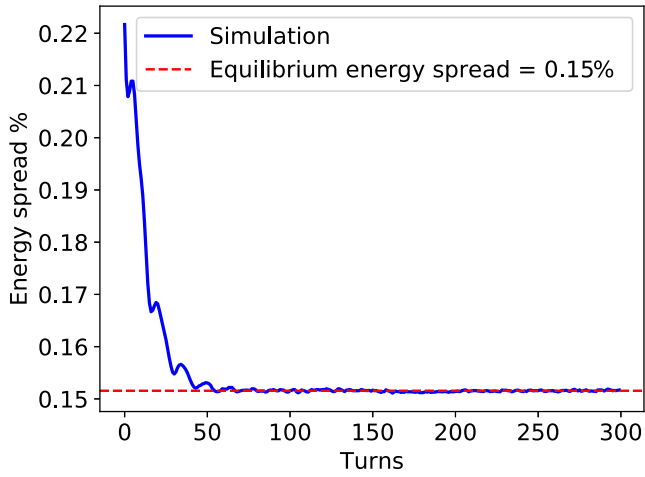


FIG. 7. Evolution of the normalized rms energy spread in FCC-ee t_1 with 365 GeV collision energy. Simulation and machine parameters: the two rf stations contain a 400 MHz rf system with 4 GV rf voltage and an 800-MHz rf system with 6.9 GV rf voltage in total; six rf stations are empty.

BLonD, when only a single rf station is present, synchrotron radiation will be applied at a single location, and the particles are prone to escaping the bucket due to this substantial energy kick they experience. *Empty rf stations* were implemented to remedy this issue. They are stations that apply all the kicks except for the rf kick (second term in Eq. (4)), allowing the energy to decrease as continuously as necessary. Stations are declared to be *empty* by setting the voltage to zero. Another use of *empty rf stations* is for rf systems that are not distributed symmetrically on the ring. In the FCC-ee design report [38], the ring is divided into eight sections, and the 400 and 800 MHz rf stations are placed on one half of it. The BLonD simulation will have eight rf stations, six of which will be *empty*.

The importance of a correct distribution of kicks can be understood through a simulation example shown in Fig. 7 in which a single bunch is generated with a normalized relative rms energy spread of about 2.2×10^{-3} , and then tracked for 300 turns ($10\tau_z$). In similar simulations with only two rf stations, and without additional empty rf stations, particles are lost from the bucket due to large discrete kicks given by synchrotron radiation and quantum excitation.

VII. GLOBAL AND LOCAL FEEDBACK MODELS

In this Section, we will describe global and local feedback models. By global feedback models, we mean models that are acting on the entire rf system, i.e., either the rf frequency or phase. By local feedback models, we mean models that act on one single rf system, typically the rf voltage amplitude and phase of a group of cavities. Often, the feedback models also involve measuring beam signals and feeding the signals back or forward to regulate the desired quantity.

A. Global feedback models

For accurate and realistic modeling of beam motion in a synchrotron, the presence of beam-based feedback systems cannot always be neglected. For example, quantitatively reproducing measured capture losses in the SPS requires the inclusion of the SPS beam phase loop in the simulation. Also, rf phase noise or modulation is in practice often injected through the set point of a beam phase loop, instead of being injected directly into the cavity set point.

The BLonD equations of motion allow the actual turn-by-turn rf phase $\varphi_{\text{rf},(n)}$ and frequency $\omega_{\text{rf},(n)}$ to deviate from the originally designed phase $\varphi_{\text{rf,d},(n)}$ and frequency $\omega_{\text{rf,d},(n)}$ programs. This feature enables the user to dynamically change the rf parameters throughout the simulation, or simply to program an rf frequency that is not an integer multiple of the revolution frequency of the clock. The deviations of the rf frequency from the exact multiple of the design revolution frequency will result in an accumulated phase deviation,

$$\Delta\varphi_{\text{rf},(n)} = \sum_{k=1}^n \frac{\omega_{\text{rf},(k)}}{h_{(k)}\omega_{\text{rev},(k)}} 2\pi h_{(k)}. \quad (32)$$

In such a case, the user will see the bunch and the rf bucket drift with respect to the reference frame. Frequency loops can be applied to minimize this phase drift and to ensure that in the long run the rf frequency (and with it, the beam orbit and energy) is maintained at its design value.

The exact implementations of the beam-based feedback models are machine-specific. For the CERN synchrotrons, the exact implementations of frequency, synchronization, radial, and beam phase loop are available for use. They can be a good starting point also for feedback loops in other machines, while also custom-made feedback models can be easily built and used with the BLonD architecture. This is ensured by the modular and object-oriented architecture of BLonD, which allows interfacing a user-determined function with the tracking code that updates particle coordinates and with the rf object that updates rf parameters. The PYTHON language used on the top level furthermore ensures fast prototyping and testing of new code.

B. Cavity feedback models

In the presence of beam and due to beam loading [41], both the amplitude and phase of the rf voltage deviate from their design values that are usually constant over a tracking turn. Given that the rf cavities are often the largest contributor to the machine impedance, the correction needed to bring these parameters back to their design values can be computed as in the real machine, by a cavity-based feedback system. Depending on the system, the correction is applied within the same turn for fast feedback systems or with a one-turn delay.

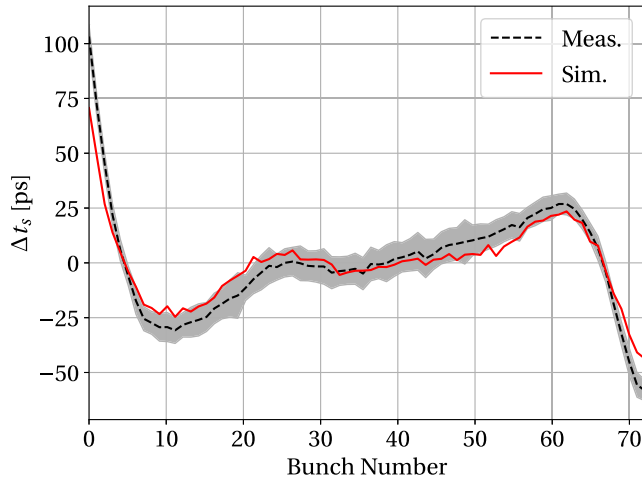


FIG. 8. Comparison between bunch-by-bunch spacing from measurements done at SPS flattop and a simulation recreating the measurements with the SPS one-turn delay feedback model in BLoND.

The accurate modeling of the feedback system of an accelerator is crucial for the realistic simulation of beam evolution, stability, and losses, as well as the assessment of the rf requirements of the machine. For example, the reduction of the bucket size resulting from the deviation of the design rf parameters, together with the modulation of the bunch-by-bunch positions resulting from the feedback loops used for beam-loading compensation can yield larger losses and reduce the machine performance. Figure 8 illustrates the modulation of the bunch-by-bunch position from a simulation in BLoND using the SPS one-turn delay feedback model compared with measurements done in the real machine [42]. This becomes especially important for the bunch-to-bucket transfer between consecutive stages of an accelerator chain, between the

PS-SPS and SPS-LHC [43–45]. In the case of the High-Luminosity LHC (HL-LHC), the presence of large power transients at the head and tail of the high-intensity beam batches could result in a power demand beyond the capacity of the rf system; to understand their magnitude and dynamics, a realistic model of the full LHC rf cavity controller is needed [46]. The exact implementation of the cavity controller is again machine dependent, and models for the SPS (see Fig. 9) and LHC controllers are presently available in BLoND.

Generally, a one-turn delay feedback system (OTFB) measures the cavity (antenna) voltage and applies the necessary correction in the following turn. In the SPS model, for instance, the antenna voltage $\vec{V}_{\text{ind},(n)}$ is the sum of the beam- and generator-induced contributions ($\vec{V}_{\text{ind},b,(n)}$, $\vec{V}_{\text{ind},g,(n)}$, respectively) [33]:

$$\vec{V}_{\text{ind},(n)}(t) = \vec{V}_{\text{ind},b,(n)}(t) + \vec{V}_{\text{ind},g,(n)}(t), \quad (33)$$

where t is resolved typically on a bucket-by-bucket basis. The additional contribution from the reflected current is added, if applicable for the system. The vector notation refers to the use of complex voltage vectors. In its BLoND implementation, all these signals span two full turns and are discretized at the rf frequency. As described in Sec. V, the beam-induced voltage is the result of the cavity response to the rf beam current $\vec{I}_{b,(n)}(t)$ which is obtained from the beam profile $\lambda_{(n)}(t)$ at f_{rf} . Likewise, the generator component in Eq. (33) is the result of the cavity response to the generator current $\vec{I}_{g,(n)}(t)$ according to the corresponding generator model (and its time evolution [47]). To compute the bunch-by-bunch correction to the rf voltage, the feedback system first computes the difference between the antenna voltage and the required design (or *set point*) voltage $\vec{V}_{d,(n)}$,

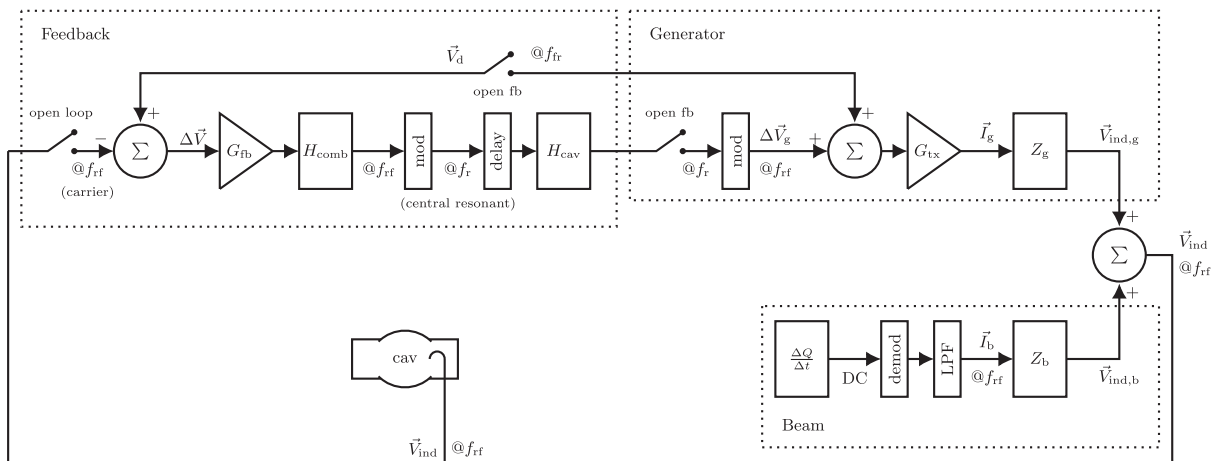


FIG. 9. Schematic of the SPS one-turn delay feedback implementation in BLoND. The correction to the rf voltage along each turn is calculated by the cavity control from the difference of the cavity voltage (antenna), i.e., the sum of the beam- and generator-induced voltages, with the design voltage (set point).

$$\Delta \vec{V}_{(n)}(t) = \vec{V}_{d,(n)} - \vec{V}_{\text{ind},(n)}(t). \quad (34)$$

This error signal is then processed (comb-filtered) to remove the beam-loading effect by comparing it with the error in the *previous* turn ($n - 1$). The resulting signal constitutes an additional input to the generator drive, from which the corrected $\vec{I}_{g,(n)}(t)$ is obtained via the transmitter model. The corresponding generator-induced voltage $\vec{V}_{\text{ind},g,(n)}$ will keep, in principle, the cavity voltage $\vec{V}_{\text{ind},(n)}$ equal to the design voltage $\vec{V}_{d,(n)}$ on a bucket-by-bucket basis in the presence of beam loading. For rf power studies, the generator power is derived from these quantities.

For several cavities at the same harmonic in a given rf station, the total corrected voltage $\vec{V}_{\text{corr},(n)}(t)$ is the sum of the cavity voltages $\vec{V}_{\text{ind},(n)}(t)$ regulated by their corresponding feedback loop. The rf voltage seen by the beam in Eq. (4), which is constant in amplitude and phase over a turn, is replaced for beam tracking by the following voltage that is modulated bucket by bucket:

$$V_{\text{rf},(n)}(t) = V_{\text{corr},(n)}(t) \sin \left[\omega_{\text{rf},(n)} t + \varphi_{\text{corr},(n)}(t) \right]. \quad (35)$$

For multiharmonic rf systems, the contribution from additional rf harmonics should be added as described in Eq. (16).

The performance of the correction to the generator can be further increased by adding a feedforward loop on the beam branch, for example, as a FIR filter [48]. In the LHC and SPS models, the feedforward acts on the present-turn signals of the feedback output voltage and on the beam-induced voltage, respectively. Moreover, the feedback (including feedforward) loop can be part of or act together

with other feedback loops such as analog and digital feedback systems, as in the case of the LHC cavity controller, to provide additional corrections.

VIII. GENERATION OF PARTICLE DISTRIBUTIONS

Once all the BLonD objects are initialized to treat the machine and impedance parameters, an initial particle distribution is needed to start the simulation. In this section, we describe several options that were included in the code to generate an initial particle distribution. The distribution can be generated either matched to the rf bucket or using an arbitrary density function. The density function in the longitudinal phase space is denoted $\mathcal{F}(\Delta t, \Delta E)$. The number of macroparticles to be generated is determined by the user, and each macroparticle corresponds to a fraction of the total beam current.

The criterion for a bunch of particles to be matched is that the density function \mathcal{F} is a function of the Hamiltonian \mathcal{H} (or the action \mathcal{J}). This implies that the particle density is uniform over an iso-Hamiltonian curve, and therefore the particle distribution will remain stationary. Two routines were included to generate bunches matched to the rf bucket: The first one requires the density function \mathcal{F} as a user input while the second one requires to input the line density λ .

The routine using the density function \mathcal{F} as an input is described in Fig. 10(a). In this case, the Hamiltonian is computed numerically on a grid in the $(\Delta t, \Delta E)$ phase space, using the input machine and rf parameters. The density function is then computed on the same grid, using the distribution function \mathcal{F} chosen by the user from the ones given in the left column of Table I. If no impedance object was defined, a random particle distribution is

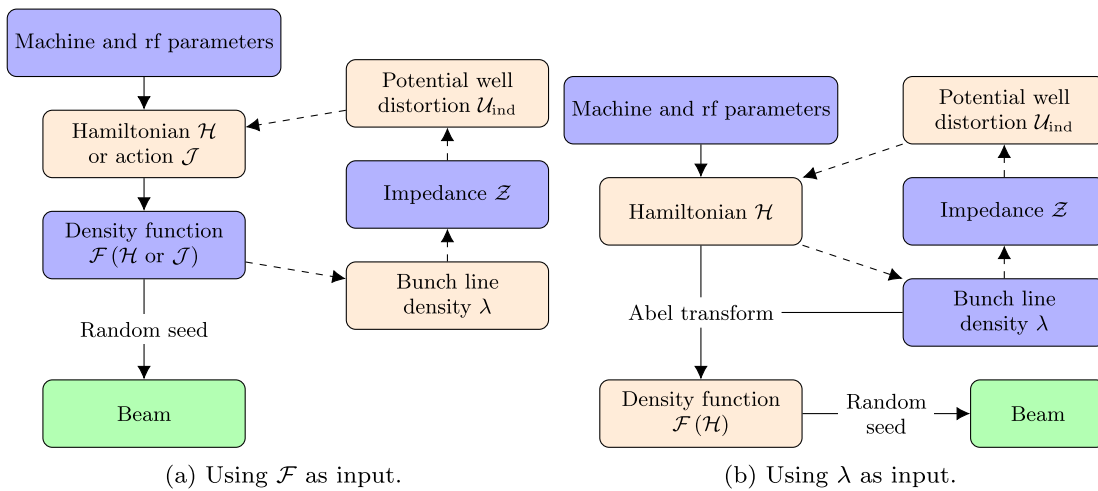


FIG. 10. Flowcharts of the routines used to generate a bunch of particles matched to the rf bucket in BLonD. The generation is an iterative process that minimizes the difference between the target emittance or bunch length asked by the user. The blue boxes represent the user input, the beige boxes are the parameters computed internally in the function, and the green boxes are the resulting beam objects used for tracking. The path in dashed arrows corresponds to the iterative loop for matching with collective effects.

TABLE I. Density functions and their corresponding line density [30].

Name	$\mathcal{F}(\mathcal{H})$	$\lambda(\Delta t)$
Binomial ^a	$\mathcal{F}_0(1 - \frac{\mathcal{H}}{\mathcal{H}_l})^\mu$	$\lambda_0[1 - 4(\frac{\Delta t}{\tau_l})^2]^\mu + 1/2$
Gaussian ^b	$\mathcal{F}_0 e^{-2\frac{\mathcal{H}}{\tau_l}}$	$\lambda_0 e^{-\frac{4\Delta t}{\tau_l}}$

^a $\mu = 0$ waterbag, $\mu = 1/2$ parabolic line density, $\mu = 1$ parabolic in amplitude, $\mathcal{F}(\mathcal{H} \geq \mathcal{H}_l) = 0$ and τ_l is the full bunch length

^b $\tau_l = 4\sigma$.

directly sampled from the density function. If impedance sources are defined, the bunch is iteratively matched to take into account potential-well distortion. This method is particularly useful for studies scanning longitudinal emittance with a fixed-density function.

The second routine using the line density λ as input is described in Fig. 10(b). For this routine, the density in phase space is obtained using the inverse Abel transform [49]. The inverse Abel transform was implemented numerically and is applicable for cases where the potential well has only one minimum. Only half of the bunch profile is required to compute the inverse Abel transform, and the second half of the bunch profile results from the calculated density \mathcal{F} . This implies that in the case of a nonsymmetric potential well due to acceleration or collective effects, one-half of the bunch profile is perfectly reproduced, and the measured and simulated profiles will only agree if the rf and impedance parameters are well known. In the presence of impedance sources, the matching consists of iteratively placing the bunch in the center of the distorted potential well till convergence, without changing the input line density. Once the density function \mathcal{F} is obtained, the particle distribution is sampled randomly. This method is particularly useful for studies starting from bunch profiles identical to the ones obtained in measurements.

Both matching routines were extended for multibunch simulations. The methods implemented in BLonD to generate an initial particle distribution allowed us to perform all kinds of studies for all synchrotrons at CERN: starting with

a matched distribution at any momentum during the acceleration ramp, with and without intensity effects, using the measured line density, etc. In addition, many routines to generate arbitrary density functions are implemented (e.g., bi-Gaussian, coasting beam). Further options are continuously being implemented, to fulfill all needs encountered in the user’s beam dynamics studies.

IX. PERFORMANCE

The demand for extensive and precise longitudinal beam dynamics simulations has become increasingly essential due to ongoing upgrade projects, investigations for future machines, and the operational demands of existing facilities at CERN and other similar research institutes. A holistic case study typically entails thousands of simulations, all geared toward identifying the parameter set that best aligns with the desired target characteristics. The duration of a single simulation can extend up to several weeks, underlying the significance of optimizing the BLonD code’s runtime performance.

BLonD was designed with an emphasis on performance optimization right from the beginning. The initial focus centered on enhancing the code’s serial or single-core efficiency. To achieve this goal, we identified the most time-consuming code regions, including the longitudinal tracking `kick` and `drift` functions, the `hist` operation responsible for generating beam line density, and the FFT operations required for calculating the beam-induced voltage, among others (see Fig. 11). To achieve significant performance gains, we translated these regions to C++, a lower-level compiled language known for its capability in developing performance-critical software, and also leveraged certain highly efficient libraries such as the Boost library [50], the Intel MKL library [51], and the VDT library [52]. As a result of resolving various performance limitations within these hotspot regions, BLonD++ was able to execute simulations that previously took a full day in just 80 minutes [53]. The success of BLonD++ can be attributed to its effective utilization of the processor’s vectorized hardware, allowing for the evaluation of mathematical

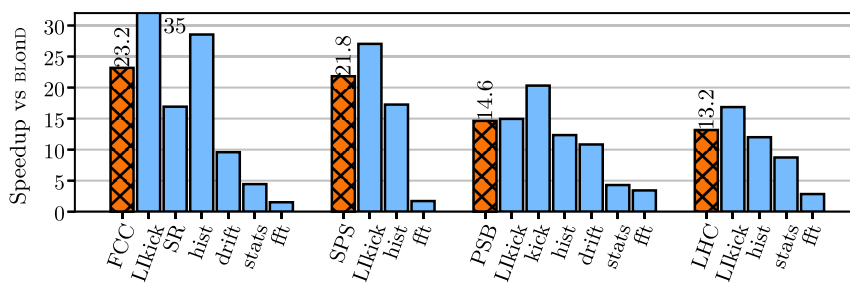


FIG. 11. BLonD++ speedup against the original PYTHON-only BLonD implementation. All four test cases (FCC, SPS, PSB, LHC) were run using only one CPU core. The first bar of every group of bars shows the overall test case speedup, while the remaining bars show the speedup of the most time-consuming code regions, which are “Lkicck”: linearly interpolated energy kick, “SR”: synchrotron radiation, “hist”: line density calculation, “drift”: drift calculation, “stats”: statistics on beam distribution, “fft”: Discrete Fast Fourier Transform.

operations over multiple input elements in a single CPU cycle. Additionally, by fusing together certain computing operations that operate on the same data, the required memory transactions were halved.

In Fig. 11, we observe the remarkable single-core speedup achieved by B_{LonD}++ when compared to the original PYTHON-only B_{LonD} implementation across four representative test cases. These simulations encompass a range of typical B_{LonD} workloads. The first bar in each group illustrates the overall speedup, while the remaining bars depict the speedup per accelerated code region. Notably, the per-testcase speedup varies from 13.2× in the LHC testcase to an impressive 23.2× in the FCC testcase. Additionally, we addressed the most time-consuming code regions by parallelizing them using the OpenMP framework [54], enabling B_{LonD} to scale vertically, i.e., within a computing node.

B_{LonD} workloads comprise a scientifically and computationally challenging task. These workloads are typically inherently parallel and fit naturally in a distributed-memory environment. In anticipation of the ever-growing demand for simulations requiring finer resolution, more realistic modeling, and longer intervals, we introduced a distributed variant of B_{LonD}, known as HBLonD [55], which can efficiently combine multiple, remote computing nodes to execute a B_{LonD} simulation. The Message Passing Interface (MPI) [56] serves as the backbone for communication between these remote nodes.

While internode communication is necessary for the coordination of remote nodes, it is the most significant bottleneck of distributed programs. To minimize it, a set of high-performance computing techniques, including dynamic load-balancing, mixed data- and task-parallelism, and approximate computing, were employed. These state-of-art computing techniques contributed to the impressive scalability demonstrated by HBLonD. Being able to efficiently combine over 600 cores across 32 computing nodes, HBLonD achieves greater than tenfold speedups compared to

B_{LonD}++, thereby reducing the simulation time further by one order of magnitude.

Graphics processing units (GPUs), originally designed for rendering images on display devices, have emerged as a dominant platform for accelerating general-purpose, data-parallel workloads. The most time-consuming tasks involved in a typical B_{LonD} simulation, listed in the caption of Fig. 11, are ideal candidates for GPU acceleration. Therefore, we developed CuB_{LonD} [57], a GPU-accelerated version of B_{LonD} implemented using the CUDA programming language [58]. CuB_{LonD} demonstrated an additional five-fold speedup compared to the CPU-only version. One of the key challenges in implementing CuB_{LonD} was optimizing CPU to GPU memory transfers, which tend to be slower compared to GPU computations. To minimize this overhead, the data arrays involved in a B_{LonD} simulation are transferred to GPU memory and remain there throughout the simulation. Only at the end of the simulation are the results copied back to CPU memory to facilitate plotting, data storage, and other analysis operations.

In Fig. 12, we conducted a weak-scaling, multinode performance evaluation of both HBLonD and CuB_{LonD} against B_{LonD}++. In the weak-scaling experiments, the amount of work per computing node is held constant as the number of nodes increases. The horizontal axis in Fig. 12 represents the number of computing nodes, ranging from one to 16, with each node containing either 20 CPU cores or one GPU platform. The four bars in each group correspond to the measured speedup against B_{LonD}++ for (a) HBLonD, (b) HBLonD using reduced (32 bit instead of 64 bit) arithmetic precision, (c) CuB_{LonD}, and (d) CuB_{LonD} using reduced arithmetic precision. As observed, HBLonD effectively utilizes multiple computing nodes to run a single B_{LonD} simulation, yielding speedups ranging from 10× up to 17× compared to the previous single-node implementation of B_{LonD}. Furthermore, by leveraging GPUs to accelerate the most time-consuming regions, CuB_{LonD} achieves impressive speedups ranging from 46× up to 78× compared to B_{LonD}++.

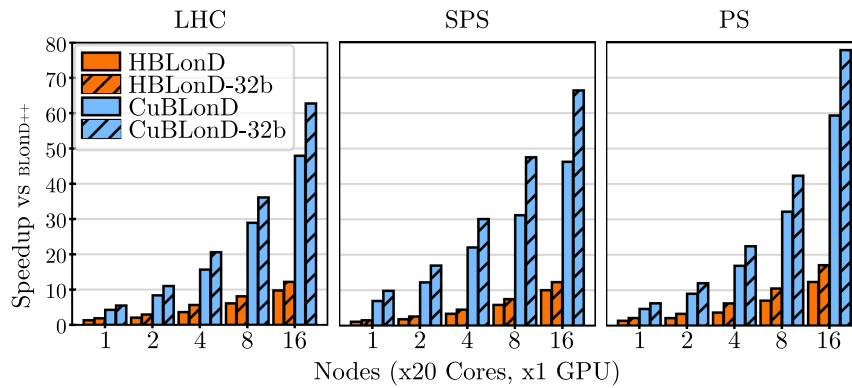


FIG. 12. Multinode performance and scalability of HBLonD and CuB_{LonD}. From left to right, the bars in every group of bars correspond to the speedup compared to B_{LonD}++ of: (a) HBLonD, (b) HBLonD with reduced (32 bit) arithmetic precision, (c) CuB_{LonD}, and (d) CuB_{LonD} with reduced (32 bit) arithmetic precision.

Currently, CuBLonD is distributed alongside the main BLonD code. In conclusion, BLonD stands out as a well-optimized simulator that embraces state-of-the-art high-performance computing standards, effectively utilizing compute resources from multiple processors and GPU cards.

X. BENCHMARKS

Since its original release, BLonD has been used by a great number of scientists in a wide range of applications. The trust in the BLonD suite and its predictions has been established through in-depth testing and benchmarking. The conducted benchmarks, including comparisons with analytical calculations, measurements from experiments run in synchrotrons, or comparisons against other tracking codes [7], are all showing sharp agreement.

In addition, every care has been taken throughout the code development to ensure that BLonD results can be compared to measurement as accurately as possible. To this end, the input distributions can be idealized (e.g., Gaussian, waterbag, etc.) or taken directly from measurements. Also, the control loop features in the code are designed to reproduce the measured beam behavior in the presence of these loops.

Below we show a few examples of code-to-code comparisons and benchmarks against theory and measurements.

A. Comparisons with the MUSIC code

Here we give an example of a benchmark [26,59] between the BLonD and MUSIC [60] codes. As mentioned in Sec. V, a binning of the bunch profile is needed in BLonD to compute the induced voltage at each turn, and there are no restrictions on the types of impedance that can be given as input. In MUSIC, only resonator impedances can be provided, whose parameters allow to build a matrix that is used to propagate the induced voltage from macroparticle to macroparticle, without the need to bin the bunch profile.

In our benchmark, only one bunch is present in the ring, and the impedance consists of just one narrow-band resonator with a quality factor $Q \gg 1$ so that the long-range wakefield couples the same bunch over multiple turns. The resonator has a low resonant frequency f_r compared to the cutoff frequency of the bunch spectrum.

If f_r is close to an integer multiple p of the revolution frequency, then Robinson instability can be observed [29]. Supposing a Gaussian line density with rms bunch length σ_t , the analytical expression for the growth rate of the instability is [29]

$$\frac{1}{\tau_a} = \frac{\eta e^2 N_p}{2E_d T_{\text{rev}}^2 \omega_s} \sum_{m=\pm 1} (m(p\omega_{\text{rev}} + m\omega_s)) \times \text{Re}Z(p\omega_{\text{rev}} + m\omega_s) G_m(x), \quad (36)$$

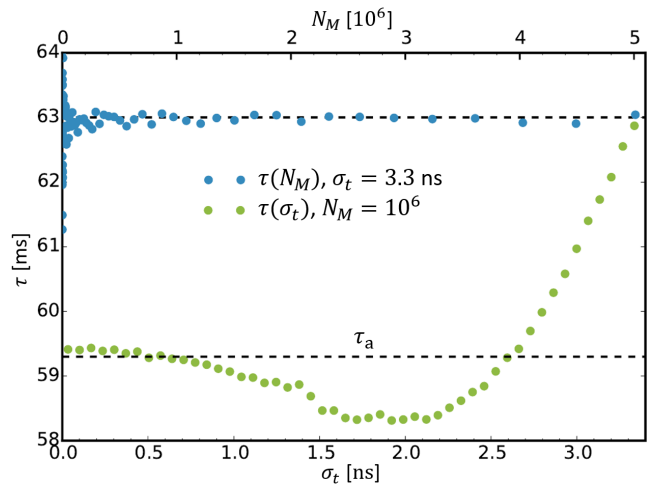


FIG. 13. Instability growth time τ as a function of N_M for $\sigma_t = 3.3$ ns (blue) and as a function of σ_t for $N_M = 10^6$ (green) from MUSIC simulations. The dashed lines mark $\tau = \tau_a = 59.3$ ms and $\tau = 63$ ms.

where $G_m(x) = 2e^{-x^2} I_m(x^2)/x^2$ is the form factor with $x = (pf_{\text{rev}} + mf_s)\sigma_t$ and I_m is the modified Bessel function of the first kind.

In our example, the integer multiple of the revolution frequency is $p = 2$, and the resonator parameters are $f_r = 2f_{\text{rev}} + f_s$, $Q = 5000$ and $R_s = 40$ k Ω . In addition, $N_p = 4 \times 10^{12}$ ppb, $E_d = 13$ GeV, $\eta = 0.0217$, $T_{\text{rev}} = 2.1$ μ s, $f_s = 264.1$ Hz. The rf system has $h = 7$, $f_{\text{rf}} = 3.3$ MHz and $\hat{V}_{\text{rf}} = 165$ kV, while $R_d = 100$ m. The instability growth time computed with Eq. (36) is $\tau_a = 59.3$ ms for $\sigma_t \leq 3.3$ ns.

Firstly, simulations with the MUSIC code are performed, by using initial bunch distributions whose line densities are Gaussian with $\sigma_t = 3.3$ ns. The instability growth time as a function of N_M is evaluated (Fig. 13, blue dots), and convergence to 63 ms is found for $N_M > 10^6$. However, results do not converge to the τ_a obtained above analytically, since the bunch is relatively long and Landau damping increases the analytical growth time. This is confirmed by performing a new set of simulations with MUSIC, selecting $N_M = 10^6$ and varying σ_t (Fig. 13, green dots). The plot shows that the growth time converges to τ_a for sufficiently low σ_t , proving the validity of the MUSIC algorithm.

The second part of our benchmark consists of performing BLonD simulations, by using initial bunch distributions with $N_M = 10^6$ and with Gaussian line densities having $\sigma_t = 3.3$ ns. The spectrum of a Gaussian line density with $\sigma_t = 3.3$ ns becomes negligible above 200 MHz, whereas the considered narrow-band resonant impedance is negligible above 1 MHz. It is then not easy to choose in BLonD the bin size Δt in the time domain or equivalently the maximum frequency $f_{\text{max}} = 1/(2\Delta t)$ in the frequency domain.

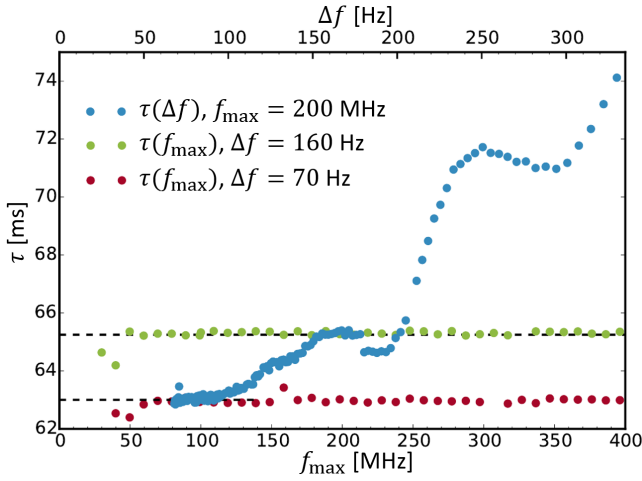


FIG. 14. Instability growth time τ as a function of Δf for $f_{\max} = 200$ MHz (blue), and versus f_{\max} for $\Delta f = 160$ Hz (green) and $\Delta f = 70$ Hz (red). The dashed lines mark $\tau = 63$ ms and $\tau = 65.3$ ms. The simulations are performed with BLonD, using initial bunch distributions with $N_M = 10^6$ and with Gaussian line densities having $\sigma_t = 3.3$ ns. The simulations are performed in the time domain with $t_{\max} = 1/\Delta f$ and $\Delta t = 1/(2f_{\max})$.

In addition, the frequency step $\Delta f = 1/t_{\max}$ for Fourier transforms is another parameter to be carefully considered. To resolve properly the narrow-band resonator impedance, a frequency step not larger than $\Delta f = 70$ Hz is required. This corresponds to a wakefield extending for at least 7000 revolution periods. To compute the induced voltage in the BLonD simulations needed for the benchmark, the time-domain approach is faster than the frequency-domain one, therefore it is preferred.

Figure 14 (blue dots) shows the dependence of the instability growth time on Δf , by choosing $f_{\max} = 200$ MHz to fully cover the frequency range where the initial bunch spectrum is not negligible. The growth time converges to 63 ms when Δf approaches 70 Hz. This is in good agreement with the results obtained with MUSIC (Fig. 13, blue dots), and it confirms that a frequency step of at least 70 Hz is necessary to obtain correct results.

Additional BLonD simulations (Fig. 14, red dots) show that the growth time is still 63 ms for $f_{\max} > 50$ MHz, as long as $\Delta f = 70$ Hz is fixed. This indicates that it is not necessary to consider the entire frequency range where the bunch spectrum is not negligible, as long as the resonator impedance is well resolved. Note that the growth time is not correct if $\Delta f = 160$ Hz, regardless of the value chosen for f_{\max} (Fig. 14, green dots). This confirms that to obtain correct results in BLonD, it is essential to properly resolve the narrow-band resonator impedance.

B. Comparisons with the MELODY code

Loss of Landau damping (LLD) in the longitudinal plane is an important intensity limitation in particle synchrotrons.

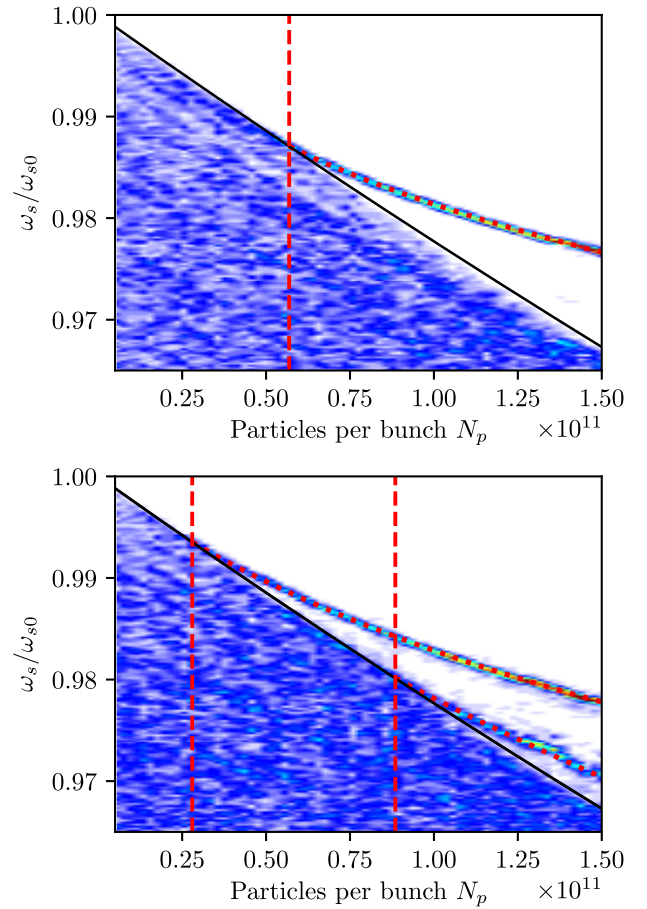


FIG. 15. The real part of the normalized mode frequency found from macroparticle simulations using BLonD (blue color) and MELODY (red dotted lines) as a function of bunch intensity for a broadband resonator impedance with $\text{Im}Z/n = 0.07 \Omega$, $Q = 1$, $f_r = 4$ GHz (top) and $f_r = 8$ GHz (bottom). The maximum incoherent frequency obtained from MELODY is shown with a black solid line. The red dashed lines indicate the LLD intensity thresholds. Plot from [61].

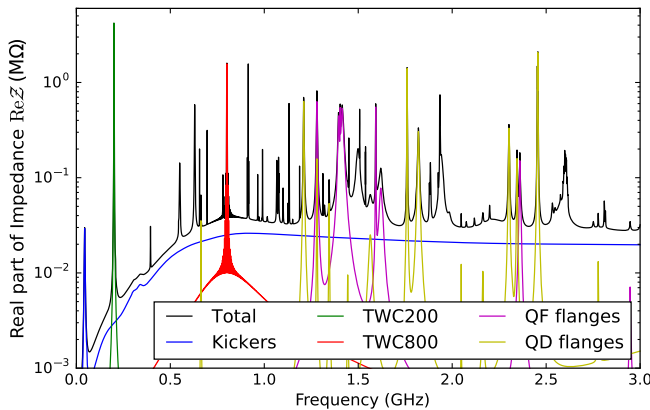
New features were recently discovered in this field and summarized in Ref. [61]. The MELODY (matrix equations for longitudinal beam dynamics) code [62] was developed for numerical studies of LLD. For example, it was shown that the LLD threshold is inversely proportional to the resonant frequency of the broadband impedance model with a quality factor of 1. This is demonstrated in Fig. 15, where we compare the frequency of the emerged coherent modes that were obtained using the MELODY code (red dotted line) and the data of macroparticle simulation obtained by BLonD (blue-shade regions). For this, a bunch matched to the total potential including intensity effects was tracked for 10^6 turns. The center of mass was evaluated at every turn and the spectrum of this signal was computed. The obtained results indicate the incoherent frequency bands and coherent modes once they move above the maximum incoherent frequency (black line). The detachments of the mode correspond to LLD thresholds evaluated by MELODY (red dashed lines).

The LLD thresholds and the mode frequencies above them are well reproduced in macroparticle simulations.

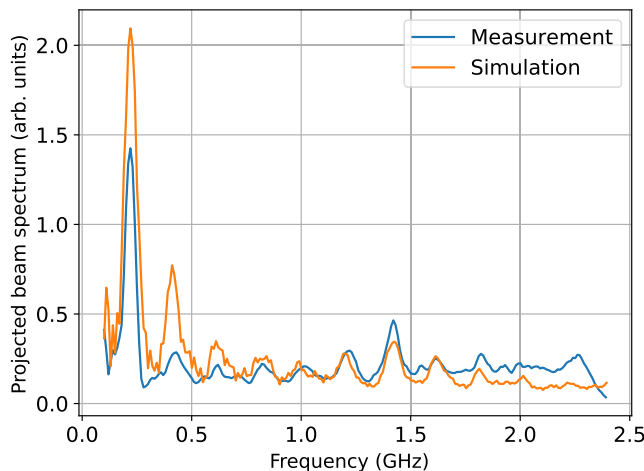
C. Benchmarks with measurements

As an application, we consider debunching measurements in the SPS [63], which are used to identify known and unknown sources of impedance in the machine. Long bunches are injected into the SPS with the rf system switched off, and the beam dynamics are entirely dictated by the machine impedance. Different impedance sources drive micro-wave instabilities, visible as a spatial modulation in the bunch profile.

The SPS impedance model is shown in Fig. 16(a) and is dominated by the impedance of the traveling wave cavities (TWC) at 200 MHz and 800 MHz, their higher-order modes (HOMs), and vacuum flanges around 1.4 GHz. The SPS impedance sources range from broadband, such as the kickers, to narrow-band, such as the 915 MHz HOM. In BLoND, the SPS impedance is modeled by over 200



(a) The SPS impedance model.



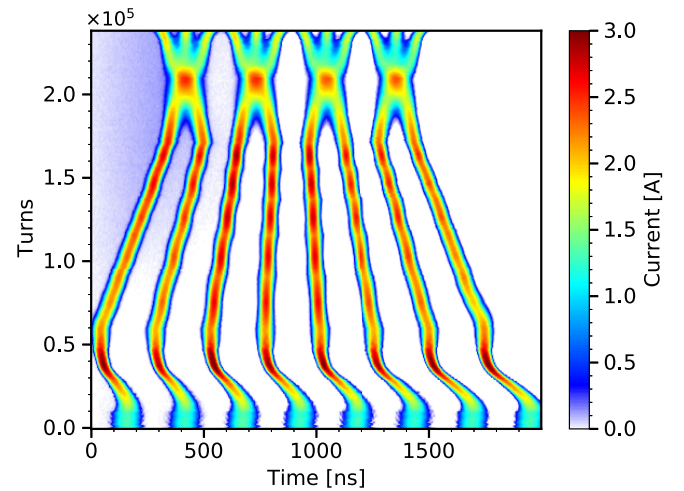
(b) Simulated and measured spectra.

FIG. 16. On top, the SPS impedance model as of 2018 [63], before an extensive impedance reduction campaign. The bottom plot compares simulated and measured spectra of a debunching beam at SPS injection.

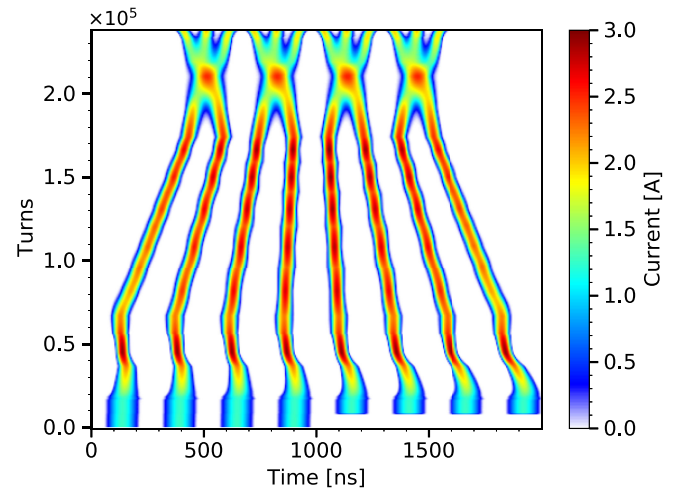
resonators, several traveling wave cavities, and impedance tables.

The sum of the measured beam spectra during the first 500 revolutions is compared with the simulated spectrum in Fig. 16(b). The initial phase-space distribution used in the simulation is based on measured profiles and tomographic reconstruction in the PS. Both spectra in Fig. 16(b) have large amplitudes at 200 MHz and 1.4 GHz due to the significant impedances of the 200 MHz TWC and flanges, respectively.

Another example is the modeling of the rf manipulations in the PS, which are performed to get the nominal bunch spacing of 25 ns for LHC beams. These manipulations are made possible thanks to the large number of rf systems in



(a) Measured bunch profiles.



(b) Simulated bunch profiles.

FIG. 17. Comparison of the measured and simulated evolution of the bunch profiles during rf manipulations in the PS. The measured phase displacement of the beam between machine turns 10 000 and 50 000 is due to the signal delay with changing revolution frequency during acceleration and is not included in the simulation.

the PS covering many rf harmonics. For the selected example shown in Fig. 17, eight bunches are injected from the PSB (two injections of four bunches) in $h = 9$ and accelerated to an intermediate energy. Next, batch compression is done to bring the beam to $h = 14$ by passing through all intermediate harmonics. Bunches are then merged into $h = 7$ [64], before finally being split in three to harmonic $h = 21$. Note that a controlled longitudinal emittance blow-up is done right after the acceleration step with phase modulation of a high-frequency rf system tuned at $h = 436$. The nine harmonics can be programmed in BLonD to simulate the whole process. Beam-loading effects were also included in the simulation as well as feedback systems. As seen in Fig. 17, the simulation reproduces very well the measured beam evolution even for complex rf configurations.

XI. CONCLUSIONS AND OUTLOOK

The Beam Longitudinal Dynamics simulation suite BLonD offers a versatile and modular framework for custom simulations, spanning a wide range of physics phenomena. From fundamental cavity-beam interactions to collective effects, diverse rf manipulations, synchrotron radiation, and feedback systems, BLonD provides the tools for various research needs. The generation of matched particle distributions finds extensive applications across different domains. Moreover, the code optimizations implemented at multiple levels and for various hardware configurations enable faster and more memory-efficient simulations.

In the future, we plan to optimize the algorithm for the Hamiltonian distribution functions used in particle distribution generation. Concurrently, we are actively working on the implementation of algorithms for nonuniform binning and arbitrary impedance. Additionally, ongoing efforts focus on incorporating machine-specific global and local feedback models for different machines, with plans for coupling them in the pipeline.

BLonD continuously evolves to stay at the forefront of simulation studies. Regular updates to existing modules and the addition of new ones allow for a better fit to specific research questions, as well as the expansion of the code's functionality and scope. To ensure robustness and minimize coding flaws, BLonD is backed by comprehensive support mechanisms, including automatic unit-testing, integration testing, and coverage analysis. In conclusion, BLonD stands as a powerful and adaptable tool, and we remain committed to refining it to meet the growing demands of longitudinal beam dynamics research.

[1] J. A. MacLachlan, ESME: Longitudinal phase space particle tracking. Program documentation, FERMILAB Technical Report No. FERMILAB-TM-1274, 1984.

- [2] CERN BLonD Simulation Suite Website (2014–2023), <http://blond.web.cern.ch>.
- [3] CERN BLonD Simulation Suite Code Repository (2014–2023), <https://gitlab.cern.ch/blond/BLonD>.
- [4] S. v. d. Walt, S. C. Colbert, and G. Varoquaux, The NumPy array: A structure for efficient numerical computation, *Comput. Sci. Eng.* **13**, 22 (2011).
- [5] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright *et al.*, SciPy 1.0: fundamental algorithms for scientific computing in Python, *Nat. Methods* **17**, 261 (2020).
- [6] R. Okuta, Y. Unno, D. Nishino, Shohei Hido, and C. Loomis, CuPy: A NumPy-compatible library for NVIDIA GPU calculations, in *Proceedings of Workshop on Machine Learning Systems (LearningSys) in the 31st Annual Conference on Neural Information Processing Systems (NIPS), Long Beach, CA, USA (2017)*, Vol. 6.
- [7] H. Timko, J. Esteban Müller, A. Lasheen, and D. Quartullo, Benchmarking the beam longitudinal dynamics code BLonD, in *Proceedings of 7th International Particle Accelerator Conference, IPAC-2016, Busan, Korea, 2016*, (JACoW, Geneva, Switzerland, 2016), WEPOY045.
- [8] J. Coupard, H. Damerau, A. Funken, R. Garoby, S. Gilardoni, B. Goddard, K. Hanke, D. Manglunki, M. Meddahi, G. Rumolo, R. Scrivens, and E. Chapochnikova, LHC injectors upgrade, Technical design report, CERN, Geneva, Technical Report No. CERN-ACC-2016-0041, 2016.
- [9] C. Ohmori, M. Furusawa, K. Hara, K. Hasegawa, Y. Sugiyama, M. Yoshii, M. Nomura, T. Shimada, F. Tamura, and M. Yamamoto, Batch compression for multi-MW J-PARC, in *Proceedings of IPAC'17, Copenhagen, Denmark (JACoW, Geneva, Switzerland, 2017)*.
- [10] O. Geithner, Z. Anelkovic, M. Bai, A. Braeuning-Demian, V. Chetvertkova, O. Chorny, C. Dimopoulou, W. Geithner, O. Gorda, F. Herfurth, M. Lestinsky, S. Litvinov, S. Reimann, A. Reiter, M. Sapinski, R. Singh, G. Vorobjev, U. Dr. Weinrich, and C. Wetzl, Ion-optical measurements at CRYRING@ESR during commissioning, in *Proceedings of 9th International Particle Accelerator Conference, IPAC'18, Vancouver, BC, Canada, 2018 (JACoW, Geneva, Switzerland, 2018)*, pp. 3161–3164, [10.18429/JACoW-IPAC2018-THPAF077](https://doi.org/10.18429/JACoW-IPAC2018-THPAF077).
- [11] Y. Sugiyama *et al.*, Simulation of phase-space offset injection with second harmonic RF for longitudinal emittance blow-up in J-PARC MR, *J. Phys. Soc. Jpn. Conf. Proc.* **33**, 011031 (2021).
- [12] F. C. Cai, J. C. Yang, J. W. Xia, G. D. Shen, J. Liu, L. J. Mao, D. Y. Yin, Y. Cong, S. Ruan, G. Wang, L. P. Yao, and X. Q. Chen, Beam loading effects and microwave instability in the booster ring of a high intensity heavy-ion accelerator facility, *Phys. Rev. Accel. Beams* **23**, 064402 (2020).
- [13] C. K. Birdsall and A. B. Langdon, *Plasma Physics via Computer Simulation* (Taylor and Francis, New York, 2005).
- [14] A. Wolski, *Beam Dynamics in High Energy Particle Accelerators* (Imperial College Press, London, 2014), <https://www.worldscientific.com/doi/pdf/10.1142/p899>.

- [15] D. Boussard, T. Bohl, T. Linnecar, and U. Wehrle, Non integer harmonic number acceleration of lead ions in the CERN SPS, in *Proceedings of Particle Accelerator Conference* (IEEE, New York, 1995), Vol. 3, pp. 1506–1508.
- [16] S. Krinsky and J. M. Wang, Bunch diffusion due to RF noise, *Part. Accel.* **12**, 107 (1982).
- [17] G. Dôme, Diffusion due to RF noise, in *Proceedings of CERN Accelerator School '85, Oxford, UK* (CERN, Geneva, 1987).
- [18] H. Timko, P. Baudrenghien, E. Shaposhnikova, and T. Mastoridis, Studies on controlled RF Noise for the LHC, in *Proceedings of HB2014, East Lansing, MI, USA* (2015), THO4LR03, <http://jacow.org/HB2014/papers/tho4lr03.pdf>.
- [19] S. Albright and D. Quartullo, Time varying RF phase noise for longitudinal emittance blow-up, *J. Phys. Conf. Ser.* **1350**, 012144 (2019).
- [20] E. Shaposhnikova, T. Argyropoulos, P. Baudrenghien, T. Mastoridis, J. E. Muller, G. Papotti, B. Salvant, H. Timko, C. Bhat, and A. Burov, Flat bunches in the LHC, CERN, Geneva, Technical Report, CERN-ACC-2014-0100, 2014.
- [21] J. Tückmantel, Digital generation of noise-signals with arbitrary constant or time-varying spectra (a noise generation software package and its application), CERN, Geneva, Technical Report No. LHC-PROJECT-Report-1055, CERN-LHC-PROJECT-Report-1055, 2008.
- [22] J.-P. Burnet, C. Carli, M. Chanel, R. Garoby, S. Gilardoni, M. Giovannozzi, S. Hancock, H. Haseroth, K. Hübner, D. Küchler, J. Lewis, A. Lombardi, D. Manglunki, M. Martini, S. Maury, E. Métral, D. Möhl, G. Plass, L. Rinolfi, R. Scrivens, R. Steerenberg, C. Steinbach, M. Vretenar, and T. Zickler, *Fifty years of the CERN Proton Synchrotron: Volume 1*, CERN Yellow Reports: Monographs (CERN, Geneva, 2011).
- [23] D. Boussard and Y. Mizumachi, Production of beams with high line-density by azimuthal combination of bunches in a synchrotron, CERN, Geneva, Technical Report No. CERN-SPS-ARF-79-11, 1979.
- [24] K. Seiya, B. Chase, J. Dey, P. Joireman, I. Kourbanis, and J. Reid, Slip stacking, in *Proceedings of CARE-HHH-APD Workshop on Finalizing the Roadmap for the Upgrade of the CERN & GSI Accelerator Complex (BEAM'07)*, CERN, Geneva, Switzerland (2007), pp. 66–68, <https://cds.cern.ch/record/1059058>.
- [25] R. Ainsworth, P. Adamson, J. Amundson, I. Kourbanis, Q. Lu, and E. Stern, High intensity space charge effects on slip stacked beam in the Fermilab Recycler, *Phys. Rev. Accel. Beams* **22**, 020404 (2019).
- [26] D. Quartullo, Simulations of RF beam manipulations including intensity effects for CERN PSB and SPS upgrades, Ph.D. thesis, Sapienza University, Rome, 2019.
- [27] T. Argyropoulos, T. Bohl, A. Lasheen, G. Papotti, D. Quartullo, and E. Shaposhnikova, Momentum slip-stacking in CERN SPS for the ion beams, in *Proceedings of 10th International Particle Accelerator Conference (IPAC'19)*, Melbourne, Australia, 2019 (JACoW, Geneva, Switzerland, 2019), pp. 3184–3187, [10.18429/JACoW-IPAC2019-WEPTS039](https://doi.org/10.18429/JACoW-IPAC2019-WEPTS039).
- [28] S. C. P. Albright and M. E. Angoletta, Frequency modulated capture of cooled coasting ion beams, in *Proceedings of IPAC'19* (JACoW, Geneva, Switzerland, 2019), pp. 2356–2358.
- [29] A. Chao, *Physics of Collective Beam Instabilities in High Energy Accelerators*, Wiley Series in Beam Physics and Accelerator Technology (Wiley, New York, 1993).
- [30] J. L. Laclare, Bunched beam coherent instabilities, in *Proceedings of CAS 1985* (Geneva, Switzerland, 1987), Vol. I, pp. 264–326, <http://dx.doi.org/10.5170/CERN-1987-003-V-1.264>.
- [31] D. Quartullo, E. Shaposhnikova, and H. Timko, Controlled longitudinal emittance blow-up using band-limited phase noise in CERN PSB, *J. Phys. Conf. Ser.* **874**, 012066 (2017).
- [32] B. W. Zotter and S. Kheifets, *Impedances and Wakes in High Energy Particle Accelerators* (World Scientific, Singapore, 1998), <https://www.worldscientific.com/doi/pdf/10.1142/3068>.
- [33] G. Dôme, The SPS acceleration system travelling wave drift-tube structure for the CERN SPS, CERN Technical Report No. CERN-SPS-ARF-77-11, 1977.
- [34] J. Murphy, S. Krinsky, and R. Gluckstern, Longitudinal wake field for an electron moving on a circular orbit, *Part. Accel.* **57**, 9 (1997), <https://cds.cern.ch/record/1120287>.
- [35] L. Wang and Y. Li, Analysis of the longitudinal space charge impedance of a round uniform beam inside parallel plates and rectangular chambers, *Phys. Rev. ST Accel. Beams* **18**, 024201 (2015).
- [36] M. Migliorati and L. Palumbo, Multibunch and multi-particle simulation code with an alternative approach to wakefield effects, *Phys. Rev. ST Accel. Beams* **18**, 031001 (2015).
- [37] M. Schwarz, P. Basler, M. v. Borstel, and A.-S. Müller, Analytic calculation of the electric field of a coherent THz pulse, *Phys. Rev. ST Accel. Beams* **17**, 050701 (2014).
- [38] M. Benedikt, A. Blondel, O. Brunner, M. Capeans Garrido, F. Cerutti, J. Gutleber, P. Janot, J. M. Jimenez, V. Mertens, A. Milanese, K. Oide, J. A. Osborne, T. Otto, Y. Papaphilippou, J. Poole, L. J. Tavian, and F. Zimmermann, FCC-ee: The Lepton Collider: Future Circular Collider Conceptual Design Report Volume 2. Future Circular Collider, CERN, Geneva, Technical Report No. CERN-ACC-2018-0057, 2019.
- [39] M. Benedikt, M. Capeans Garrido, F. Cerutti, B. Goddard, J. Gutleber, J. M. Jimenez, M. Mangano, V. Mertens, J. A. Osborne, T. Otto, J. Poole, W. Riegler, D. Schulte, L. J. Tavian, D. Tommasini, and F. Zimmermann, FCC-hh: The Hadron Collider: Future Circular Collider Conceptual Design Report Volume 3. Future Circular Collider, CERN, Geneva, Technical Report No. CERN-ACC-2018-0058 (2019).
- [40] J. F. E. Müller, Modification of the simulation code BLonD for lepton rings, CERN, 2017, <https://doi.org/10.5281/zenodo.7675649>.
- [41] D. Boussard, Beam loading, CERN, Geneva, Technical Report No. CERN-SPS-86-10-ARF, 1986.
- [42] B. E. Karlsen-Bæck, Modelling control loops for SPS-LHC beam transfer studies, Master's thesis, Norwegian University of Science and Technology, 2022.
- [43] M. Schwarz, H. Bartosik, E. Shaposhnikova, A. Lasheen, J. Repond, and H. Timko, Studies of capture and flat-bottom

- losses in the SPS, in *Proceedings of 61st ICFA Advanced Beam Dynamics Workshop (HB'18), Daejeon, Korea, 2018* (JACoW, Geneva, Switzerland, 2018), pp. 180–185, [10.18429/JACoW-HB2018-TUP2WA03](https://doi.org/10.18429/JACoW-HB2018-TUP2WA03).
- [44] H. Timko, T. Argyropoulos, I. Karpov, and E. Shaposhnikova, Beam instabilities after injection to the LHC, in *Proceedings of 61st ICFA Advanced Beam Dynamics Workshop (HB'18), Daejeon, Korea, 2018* (JACoW, Geneva, Switzerland, 2018), pp. 163–167, [10.18429/JACoW-HB2018-TUP1WA03](https://doi.org/10.18429/JACoW-HB2018-TUP1WA03).
- [45] L. Medina, T. Argyropoulos, R. Calaga, and H. Timko, Studies of longitudinal beam losses at LHC injection, in *Proceedings of the 12th International Particle Accelerator Conference (IPAC'21), Campinas, Brazil, 2021* (JACoW, Geneva, Switzerland, 2021).
- [46] L. Medina, T. Argyropoulos, P. Baudrengnien, I. Karpov, and H. Timko, in *Proceedings of International Particle Accelerator Conference, IPAC'21, Campinas, Brazil* (JACoW, Geneva, Switzerland, 2021).
- [47] J. Tückmantel, Cavity-beam-transmitter interaction formula collection with derivation, CERN Technical Report No. ATS-Note-2011-002 TECH, 2010.
- [48] P. Baudrengnien and T. Mastoridis, I/Q model of the SPS 200 MHz travelling wave cavity and feedforward design, CERN Technical Report No. CERN-ACC-NOTE-2020-0032, 2020.
- [49] P. W. Krempel, The Abel-type integral transformation with the kernel $(t^2-x^2)^{-1/2}$ and its application to density distributions of particle beams, CERN Technical Report No. CERN Note MPS/Int. BR/74-1, 1974.
- [50] B. Schäling, *The Boost C++ Libraries* (Boris Schäling, 2011).
- [51] intelmkl, “Intel math kernel library,” (2018), <https://software.intel.com/en-us/mkl>.
- [52] D. Piparo, V. Innocente, and T. Hauth, Speeding up HEP experiment software with a library of fast and auto-vectorisable mathematical functions, *J. Phys. Conf. Ser.* **513**, 052027 (2014).
- [53] K. Iliakis, H. Timko, S. Xydis, and D. Soudris, `BLoND++`: Performance analysis and optimizations for enabling complex, accurate and fast beam dynamics studies, in *Proceedings of the 18th International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation, SAMOS '18* (ACM, New York, NY, 2018), pp. 123–130.
- [54] L. Dagum and R. Menon, OpenMP: An industry standard API for shared-memory programming, *IEEE Comput. Sci. Eng.* **5**, 46 (1998).
- [55] K. Iliakis, H. Timko, S. Xydis, and D. Soudris, Scale-out beam longitudinal dynamics simulations, in *Proceedings of the 17th ACM International Conference on Computing Frontiers, CF '20* (Association for Computing Machinery, New York, NY, 2020), pp. 29–38.
- [56] M. Snir, S. Otto, S. Huss-Lederman, D. Walker, and J. Dongarra, *MPI-The Complete Reference, Volume 1: The MPI Core* (MIT Press, Cambridge, MA, 1998), 2nd ed.
- [57] K. Iliakis, H. Timko, S. Xydis, P. Tsapatsaris, and D. Soudris, Enabling large scale simulations for particle accelerators, *IEEE Trans. Parallel Distrib. Syst.* **33**, 4425 (2022).
- [58] D. Luebke, CUDA: Scalable parallel programming for high-performance scientific computing, in *Proceedings of the 2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, Paris, France, 2008* (2007), pp. 836–838, <https://dx.doi.org/10.1109/ISBI.2008.4541126>.
- [59] D. Quartullo, M. Migliorati, and J. Repond, Comparison of different methods to calculate induced voltage in longitudinal beam dynamics codes, in *Proceedings of International Particle Accelerator Conference (IPAC'17), Copenhagen, Denmark, 2017* (JACoW, Geneva, Switzerland, 2017), pp. 4465–4468, [10.18429/JACoW-IPAC2017-THPVA022](https://doi.org/10.18429/JACoW-IPAC2017-THPVA022).
- [60] M. Migliorati and L. Palumbo, Multibunch and multi-particle simulation code with an alternative approach to wakefield effects, *Phys. Rev. ST Accel. Beams* **18**, 031001 (2015).
- [61] I. Karpov, T. Argyropoulos, and E. Shaposhnikova, Thresholds for loss of Landau damping in longitudinal plane, *Phys. Rev. Accel. Beams* **24**, 011002 (2021).
- [62] I. Karpov, Matrix Equations for Longitudinal beam Dynamics (MELODY) code, <https://gitlab.cern.ch/ikarpov/melody>.
- [63] A. Lasheen, T. Argyropoulos, T. Bohl, J. F. Esteban Müller, H. Timko, and E. Shaposhnikova, Beam measurement of the high frequency impedance sources with long bunches in the CERN Super Proton Synchrotron, *Phys. Rev. Accel. Beams* **21**, 034401 (2018).
- [64] I. Bozsik, I. Hofmann, A. Jahnke, and R. W. Müller, in *Computing in Accelerator Design and Operation*, edited by W. Busse and R. Zelazny (Springer Berlin Heidelberg, Berlin, Heidelberg, 1984), pp. 128–133.