

Beam Longitudinal Dynamics Simulation Suite BLonD

H. Timko, S. Albright, T. Argyropoulos, K. Iliakis,* B. E. Karlsen-Bæck,
I. Karpov, A. Lasheen, L. Medina, D. Quartullo, and J. Repond
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
(Dated: October 4, 2022)

The beam longitudinal dynamics code BLonD has been developed at CERN since 2014 and has become a central tool for 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 BLonD-specific 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 multi-bunch distributions matched to a given impedance model. BLonD is furthermore a well-tested and optimized simulation suite, which is demonstrated through examples, too.

PACS numbers: 29.20.D, 29.27.-a, 07.05.Tp

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 can combine for a given study 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. It is an open-source particle tracking code for 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], and Scipy [5].

The code uses macro-particles 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 $10^4 - 10^7$ macro-particles/bunch are used. BLonD has a modular structure that allows the user to model different effects according to his/her needs. The BLonD code's unique feature

is that it disentangles the equations of the beam particles and the radio-frequency (rf) system, and tracks both of 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 (multi-)bunch phase-space distributions with collective effects, rf phase noise and sinusoidal rf phase modulation, and its overall flexibility.

By now, the BLonD suite has been thoroughly benchmarked [6] and applied for all the CERN existing and future synchrotrons; the outcome of the simulation studies has often been guiding the baseline choices for machine upgrades and studies. Several other institutes have started to use the code as well [7, 8]. To mention a few 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

* konstantinos.iliakis@cern.ch

simulation results. The successful operation of the Large Hadron Collider (LHC) relies on controlled emittance blow-up performed during a 14-million-turn energy ramp with several feedback loops interacting. The proposed electron-positron Future Circular Collider (FCC) poses challenges with its numerical requirements for resolving small beams in a large machine.

In this article, we present the unique features and special applications of the BLoND code. The paper is structured as follows: starting from the (i) reference frame and beam-cavity interactions, we then describe (ii) the modulation of rf parameters, (iii) the modeling of impedance and collective effects, (iv) synchrotron radiation and quantum excitation, (v) global and local feedback models, over the (vi) generation of particle distributions, to (vii) optimizations and (viii) benchmark techniques.

II. 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 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.

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 sub-cycling 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. 1.

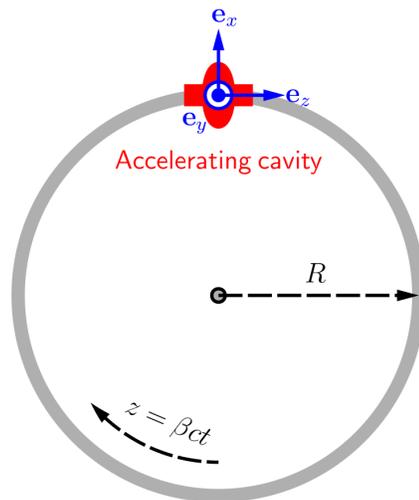


FIG. 1. Reference frame for the equations of motion and the reference time. A given particle on the orbit R is described with the energy E and the longitudinal coordinate t .

The 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,

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

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 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.

A. Periodicity

In some specific cases, such as the PSB operated at the $h = 1$ harmonic, a single bunch can cover the entire machine circumference, with the head and the tail of the bunch crossing the boundaries between previous, present, and next turns, see Fig. 2. For such cases, the periodicity

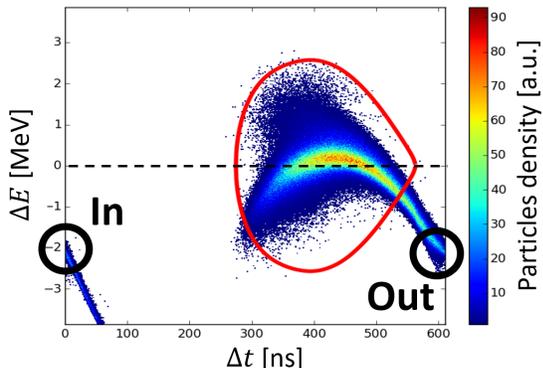


FIG. 2. Phase-space plot of unstable bunch in accelerating bucket ($h = 1$). The time coordinate on the horizontal axis spans from 0 to T_{rev} . With the periodic condition enabled, particles that exit the time frame from the right enters back into the frame from the left.

of the equations of motion can be enabled. The periodicity will ensure that all particles remain synchronized to the present time frame, by tracking twice the particles that lag behind, and by pausing the tracking of particles that are ahead in time.

If the periodicity algorithm is not applied, the equations of motion make all the particles that cross the line $\Delta t = T_{\text{rev}}$ in phase space drift away from the rf bucket. This ignores the geometry of the ring assumed in the equations of motion, and could give undesirable results for example when the beam losses have to be computed accurately or when an un-bunched beam has to be captured inside an rf bucket.

The periodicity algorithm is also useful when parts of the rf bucket cross the line $\Delta t = T_{\text{rev}}$ 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.

III. 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. Below, we show some examples on 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 blow-up [9, 10], both band-limited rf phase noise of the main harmonic [11, 12] and single-frequency modulation of a high harmonic [13] 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 [14] can be applied.

In the case of single-frequency modulation, $\Delta\varphi$ is computed as

$$\Delta\varphi_n = A \sin\left(2\pi \int f_n dt\right) + \varphi_{\text{off}} \quad (6)$$

where A is the modulation amplitude, f_n is the modulation frequency and φ_{off} is an offset about which the modulation is applied. To correctly simulate the phase modulation a frequency modulation is computed at the same time, defined by:

$$\Delta\omega_n = \frac{1}{2\pi} \frac{d\Delta\varphi_n}{dt} \omega_n. \quad (7)$$

B. Fixed-field manipulations: slip stacking

Momentum slip-stacking (MSS) is one of the most complicated rf manipulations and is currently being studied in simulations for the ion beams in the SPS [15] using the BLoND code. It permits two high-energy particle beams of slightly different momenta to slip azimuthally, relative to each other, in 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 and it is perturbed by the other. When the two beams are in the desired azimuthal position, the full beam is recaptured with a much 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. 3. Details on how MSS is going to be applied in the SPS can be found in [16, 17].

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

$$\frac{\Delta\omega_{\text{rf}}}{\omega_{\text{rf},d}} = -\eta_0 \frac{\Delta p}{p_d}, \quad (8)$$

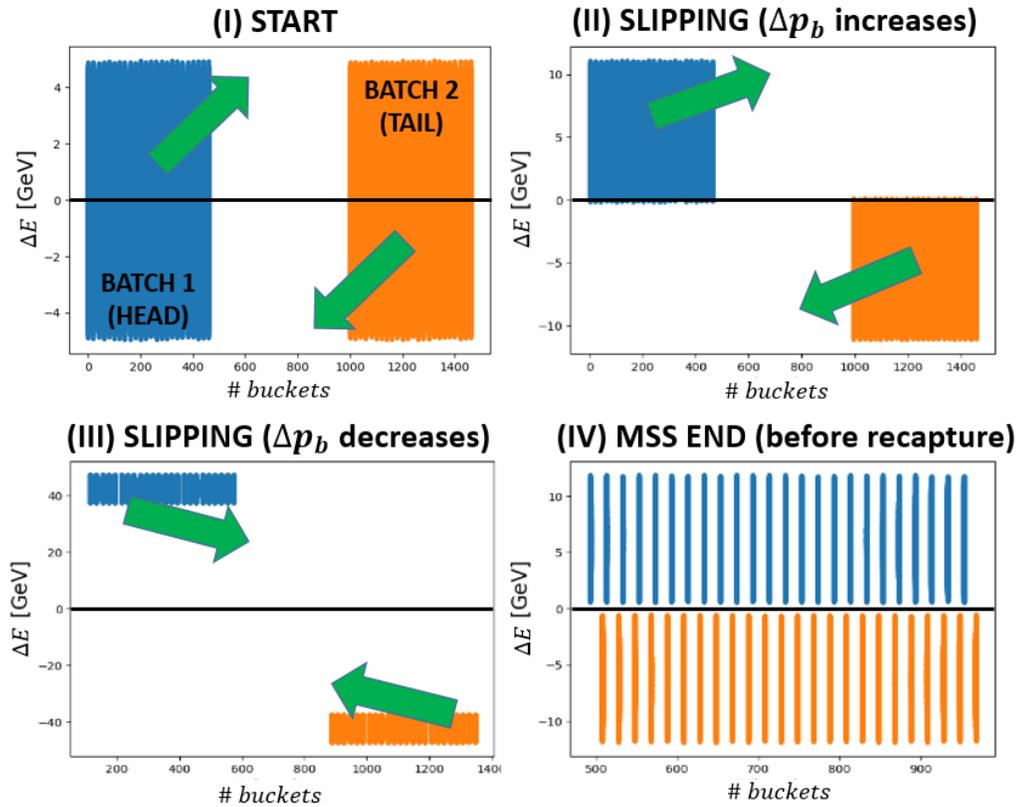


FIG. 3. Illustration of the MSS procedure simulated in BLonD. Two batches, starting from Phase I (top left) are separated in energy and move in the longitudinal phase space relative to each other. The black line marks $\Delta E = E - E_d = 0$. In Phase II (top right), the energy distance between the batches increases, while the opposite occurs in Phase III (bottom left). Recapture is done in Phase IV (bottom right).

where $\Delta\omega_{\text{rf}}$ and Δp are the changes of the rf angular frequency and beam momentum with respect to their designed 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

$$\Delta\varphi_{\text{rf}} = \frac{2\pi h \Delta\omega_{\text{rf}}}{\omega_{\text{rf,d}}}. \quad (9)$$

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

$$V_{\text{rf}} = \hat{V}_{\text{rf},1} \sin(\omega_{\text{rf},1}t + \phi_{\text{rf},1}) + \hat{V}_{\text{rf},2} \sin(\omega_{\text{rf},2}t - \phi_{\text{rf},2}), \quad (10)$$

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

IV. MODELING IMPEDANCE AND COLLECTIVE EFFECTS

In particle accelerators with high-intensity and high-brightness beams, the performance is usually limited by the electromagnetic interaction of the particles with each

other and their surroundings, which are known as collective effects.

An outcome of these effects is known as potential-well distortion, which causes, on one hand, incoherent effects as a shift in the synchronous phase and in the synchrotron frequency, as well as bunch lengthening [18]. For example, the synchrotron frequency shift may affect the performance of a controlled longitudinal emittance blow-up [19]. On the other hand, coherent effects can trigger instabilities, which can affect individual bunches or couple several bunches, and can produce uncontrolled emittance blow-up and high beam losses [20].

BLonD can accurately simulate these effects using the concept of wakefields and impedances [21]. The electromagnetic interaction is described by the so-called wake function $W(t)$, which represents the electric field excited by a point charge as experienced by a test charge.

Computing the interaction between the individual macroparticles is a $O(N^2)$ problem. In most practical cases, it is desirable to apply a binning to the particle distribution and calculate the wake potential, also called induced voltage $V_{\text{ind}}(\Delta t)$, which 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:

$$V_{\text{ind}}(\Delta t) = -q N_p \int_{-\infty}^{+\infty} \lambda(\tau) W(\Delta t - \tau) d\tau, \quad (11)$$

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

The energy kick $E_{\text{ind}}(\Delta t) = q V_{\text{ind}}(\Delta t)$ due to the induced voltage enters the term E_{other} in Eq. (4), where the minus sign in Eq. (11) ensures an energy loss for a positive wake potential.

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

$$Z(\omega) = \int_{-\infty}^{+\infty} W(t) e^{-j\omega t} dt, \quad (12)$$

$$V_{\text{ind}}(\Delta t) = -\frac{q N_p}{2\pi} \int_{-\infty}^{+\infty} Z(\omega) \Lambda(\omega) e^{j\omega \Delta t} d\omega, \quad (13)$$

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 it correctly. However, in 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 time domain will require significantly less resources. Alternatively, a broadband impedance will result in a wakefield that would require too fine resolution in 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., [22]) 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

$$Z(f) = \frac{R_s}{1 + jQ \left(\frac{f}{f_r} - \frac{f_r}{f} \right)}, \quad (14)$$

where R_s is the shunt impedance, f_r is the resonant frequency, and Q the quality factor. 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, and a model for traveling wave cavities. Coherent synchrotron radiation can be modeled as an impedance, too.

Finally, BLoND also takes as input 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. For more details, see BLoND documentation.

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_i .

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

In frequency domain, the method consists of discretizing Eq. (13) as

$$V_{\text{ind}}[n] = -q N_p \text{IDFT} \{ Z[k] \Lambda[k] \}, \quad (15)$$

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.

$$V_{\text{ind}}[n] = -q N_p \text{IDFT} \{ \text{DFT}(W[n]) \text{DFT}(\lambda[n]) \}. \quad (16)$$

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 FFT is $O(L \log L)$, to be compared to $O(NM)$ using the direct convolution algorithm in time domain.

Given the similarities between Eq. (15) and Eq. (16), BLoND has a single implementation for both methods, with the difference that when doing time-domain calculations, a pseudo-impedance is defined as $Z^*[k] = \text{DFT}(W^*[n])$, where the $*$ represents that the signal is zero-padded. Similarly, the beam spectrum is defined as $\Lambda^*[k] = \text{DFT}(\lambda^*[n])$.

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:

$$V_{\text{ind}}[n] = -\frac{q T_{\text{rev}}}{2\pi T_s} \frac{Z}{n} \frac{d\lambda[n]}{dn}, \quad (17)$$

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., [23]).

BLoND implements also two special algorithms that are limited to resonator impedances. The MuSiC algorithm [24] uses a propagation matrix to compute the induced voltage without binning. The second algorithm is an adaptation of a semi-analytic method [25] 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.

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.

V. 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 [26], and a 100 TeV hadron machine (FCC-hh) [27]. 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.

The synchrotron radiation module in BLoND applies the energy kicks due to the average energy loss per particle per turn U_0 , a damping term proportional to the particle energy offset, and quantum excitation. The detailed implementation is summarized in Ref. [28]. Later, the option of an ‘empty rf station’ was added, which applies all kicks except for the rf kick. This is especially

important for FCC-ee $t\bar{t}$, which has a collision energy of 365 GeV and $U_0 = 9.2$ GeV. Its double rf system is distributed in two opposite locations of the ring and consists of multi-cell cavities operating at 400 MHz and 800 MHz providing 4 GV and 6.9 GV of rf voltage, respectively.

The importance of a correct distribution of kicks can be understood through a simulation example, in which a single bunch is generated with a normalized relative rms energy spread of about 2.2×10^{-3} , using eight rf stations, out of which only two provide half of the total voltage each. The others are ‘empty’ and activated by setting zero rf voltage for them. The evolution of the normalized rms energy spread shows that the equilibrium value agrees well with the analytic expectation [29], see Fig. 4. 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.

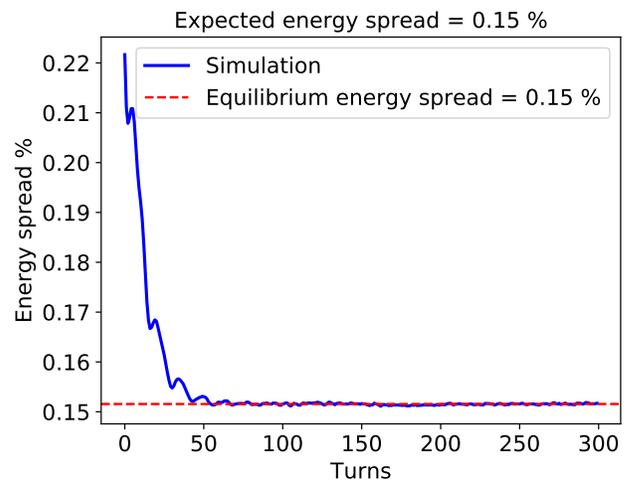


FIG. 4. Evolution of the normalized rms energy spread in FCC-ee $t\bar{t}$ with 365 GeV collision energy. Simulation and machine parameters: the two rf stations contain a 400 MHz rf system with 4 GV rf voltage and 800 MHz rf system with 6.9 GV rf voltage in total; six rf stations are empty.

VI. 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,

$$\Delta\varphi_{\text{rf},(n)} = \sum_{k=1}^n \frac{\omega_{\text{rf},(k)}}{h(k)\omega_{\text{rev},(k)}} 2\pi h(k). \quad (18)$$

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.

B. Cavity feedback models

In the presence of beam and due to beam loading [30], 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.

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. This becomes especially important for the bunch-to-bucket transfer between consecutive stages of an accelerator chain, between the PS-SPS and SPS-LHC [31–33]. 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 [34]. The exact implementation of the cavity controller is machine-dependent, and models for the SPS (see Fig. 5) 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) [35]:

$$\vec{V}_{\text{ind},(n)}(t) = \vec{V}_{\text{ind,b},(n)}(t) + \vec{V}_{\text{ind,g},(n)}(t), \quad (19)$$

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 one full turn and are discretized at the rf frequency. As described in Sec. IV, the beam-induced voltage is the result of the cavity response to the rf beam current $\vec{I}_{\text{b},(n)}(t)$ which is obtained from beam profile $\lambda_{(n)}(t)$ at f_{rf} . Likewise, the generator component in Eq. (19) is the result of the cavity response to the generator current $\vec{I}_{\text{g},(n)}(t)$ according to the corresponding generator model (and its time evolution [36]). 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 $\vec{V}_{\text{d},(n)}$,

$$\Delta\vec{V}_{(n)}(t) = \vec{V}_{\text{d},(n)} - \vec{V}_{\text{ind},(n)}(t). \quad (20)$$

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 $\vec{I}_{\text{g},(n)}(t)$ 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 $\vec{V}_{\text{d},(n)}$ 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

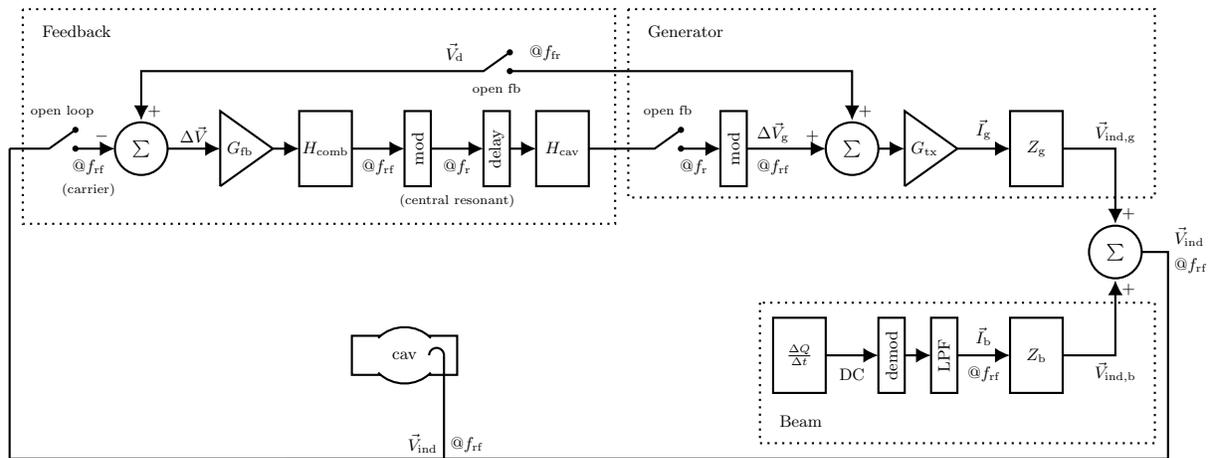


FIG. 5. Schematic of the SPS one-turn delay feedback implementation in BLoND. The correction to the rf voltage along each turn is calculated by the difference of the cavity voltage (antenna), i.e. the sum of the beam- and generator-induced voltages, with the design voltage (set point).

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:

$$V_{\text{rf},(n)}(t) = V_{\text{corr},(n)}(t) \sin(\omega_{\text{rf},(n)}t + \varphi_{\text{corr},(n)}(t)). \quad (21)$$

For multi-harmonic rf systems, the contribution from additional rf harmonics should be added as described in Eq. (10).

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 [37]. 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 analogue and digital feedback systems, as in the case of the LHC cavity controller, to provide additional corrections.

VII. 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 macro-particles to be generated is determined by the user, and each macro-particle corresponds to a fraction of the total beam current.

The criterion for a bunch of particles to be matched is

TABLE I. Density functions and their corresponding line density [18].

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

^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 $\tau_l = 4\sigma$.

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. 6a. 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 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. 6b. For this routine, the density in phase space is obtained using the inverse Abel transform [38]. The inverse Abel transform was implemented

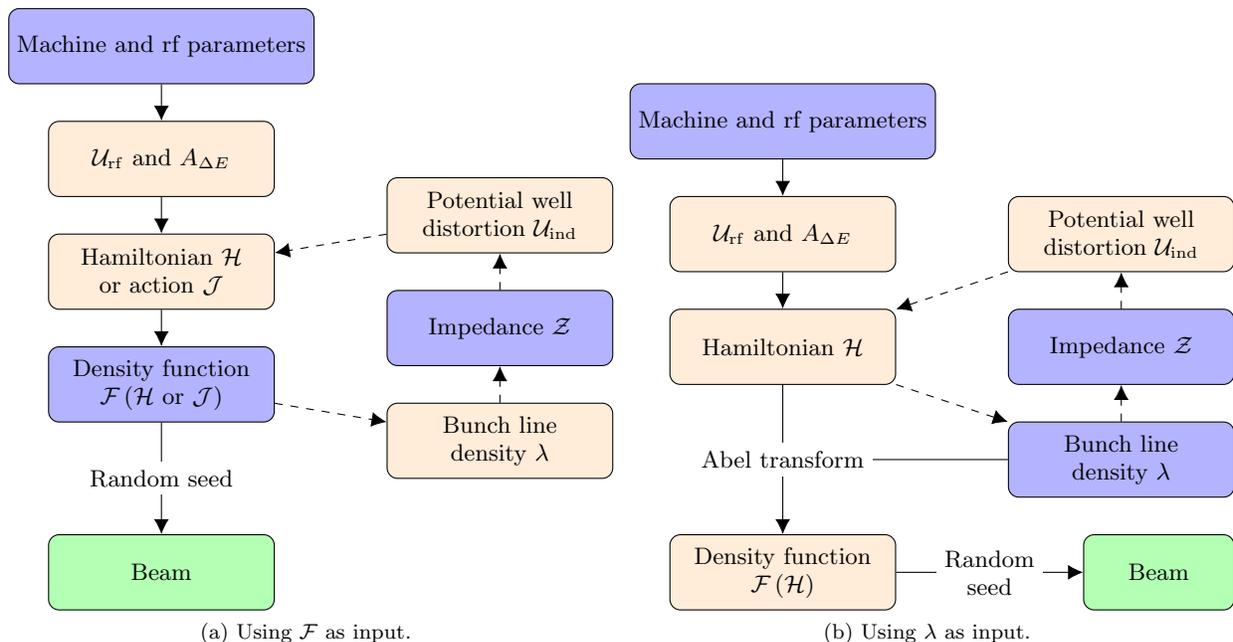


FIG. 6. Flowcharts of the routines used to generate a bunch of particles matched to the rf bucket in BLoND. The generation is an iterative process that minimizes the difference between the target emittance or bunch length asked by the user. The blue boxes represent the user input, the beige boxes are 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.

numerically, and is applicable for cases where the potential well has only one minimum. Only a half of the bunch profile is required to compute the inverse Abel transform, the second half of the bunch profile results from the calculated density \mathcal{F} . This implies that in case of a non-symmetric 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 multi-bunch simulations. The methods implemented in BLoND to generate an initial particle distribution allowed to perform all kind 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 users' beam dynamics studies.

VIII. OPTIMIZATIONS

The demand for extensive and accurate longitudinal beam dynamics simulations is dictated by the ongoing upgrade projects, the studies for future machines as well as the operation of existing machines at CERN and other similar research facilities. A complete case study is typically composed of thousands of simulations, with the aim of identifying the set of parameters that optimally satisfies the target characteristics. Depending on the case, a single simulation can last up to several weeks. Therefore, optimizing the BLoND code in terms of run-time performance is crucial.

BLoND was developed with performance in mind from the beginning. At first, the attention was focused on optimizing the serial or single-core performance of the code. For this purpose, the most time consuming code regions, the hotspots, were identified and translated to C++. C++ is a lower-level, compiled language that generally allows for the development of performance critical software, and is supported by highly-efficient libraries such as the Boost library [39], the Intel MKL library [40], and the VDT library [41]. After analyzing and resolving a series of performance limitations in the hotspot regions, BLoND++ was able to execute a previously day long simulation in 80 minutes [42]. In Fig. 7, we can see the single-core speedup of BLoND++ compared to the original Python-only BLoND implementation, in four testcases that are well-representative of typical BLoND workloads. The first bar of every group shows the overall speedup while

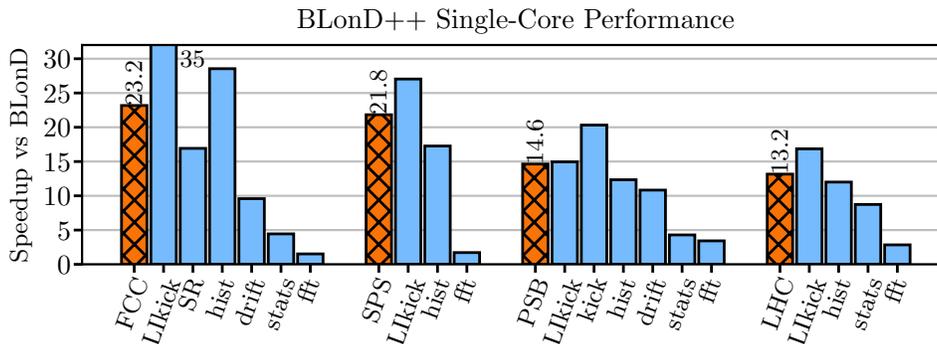


FIG. 7. BLonD++ speedup against the original Python-only BLonD implementation. All four testcases (FCC, SPS, PSB, LHC) were run using only one CPU core. The first bar of every group of bars shows the overall testcase speedup, while the remaining bars show the speedup of the most time-consuming code regions, which are ‘Lkick’: linearly-interpolated energy kick, ‘SR’: synchrotron radiation, ‘hist’: line density calculation, ‘drift’: drift calculation, ‘stats’: statistics on beam distribution, ‘fft’: Discrete Fast Fourier Transform.

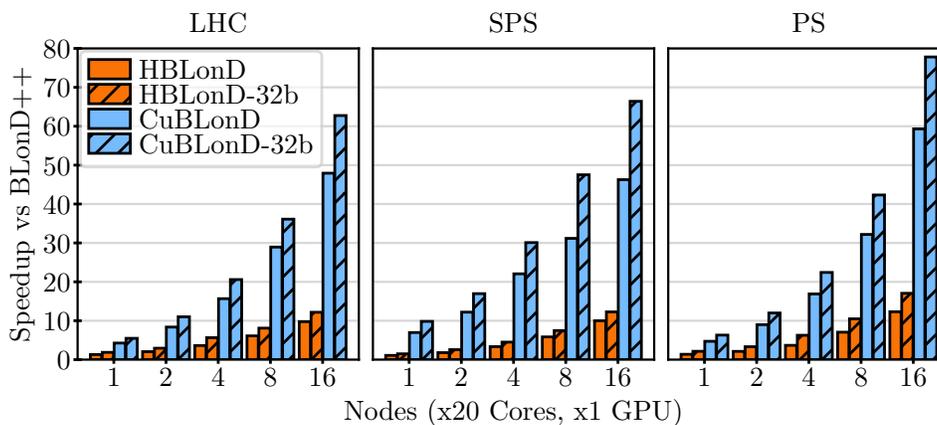


FIG. 8. Multi-node performance and scalability of HBLonD and CuBLonD. From left to right, the bars in every group of bars correspond to the speedup compared to BLonD++ of: a) HBLonD, b) HBLonD with reduced (32-bit) arithmetic precision, c) CuBLonD, and d) CuBLonD with reduced (32-bit) arithmetic precision.

the remaining bars in each group show the speedup per accelerated code region. Since different testcases require a different configuration in terms of simulated macroparticles, number of beam slices, and simulation iterations, the per-testcase speedup ranges from 13.2 \times , in the LHC testcase, to 23.2 \times , in the FCC testcase. In addition, the most time-consuming code regions were parallelized using the OpenMP framework [43], enabling BLonD to scale vertically, i.e. within the boundaries of a single computing node.

BLonD workloads comprise a scientifically and computationally challenging task. These workloads are typically inherently parallel and fit naturally in a distributed-memory environment. To anticipate the continuous need for simulations with finer resolution, more realistic modeling and longer simulation intervals, we developed a distributed version of BLonD, called HBLonD [44], that can efficiently combine multiple, remote computing nodes to simultaneously calculate a BLonD simulation. The Message Passing Interface (MPI) [45], was used to allow the remote nodes to communicate. State-of-the-art high-

performance computing techniques, like dynamic load-balancing, mixed-parallelism, and approximate computing, contributed to the impressive scalability demonstrated by HBLonD. Being able to efficiently combine up-to more than 600 cores across 32 computing nodes, HBLonD can offer greater than 10-fold speedups compared to BLonD++, thus reducing the simulation time further by one order of magnitude.

Graphics processing units (GPUs), originally designed to render images shown in display devices, e.g. a computer monitor, have lately become the dominant platform for accelerating general-purpose, data-parallel workloads. The most time-consuming tasks involved in a typical BLonD simulation, which are the particle tracking and the induced voltage calculation, are good candidates for GPU acceleration. Therefore, we implemented a GPU accelerated version of BLonD using the CUDA programming language [46], called CuBLonD [47], which demonstrated an additional five-fold speedup compared to the CPU only version. In Fig. 8, we performed a weak-scaling, multi-node performance evaluation of HBLonD

and CuBLonD against BLonD++. In weak-scaling experiments, the amount of work per computing node is kept constant as the number of nodes increases. The horizontal axis of Fig. 8 shows the number of computing nodes, ranging from one to 16. Each computing node contains either 20 cores or one GPU platform. From left to right, the four bars in every group of bars correspond to the measured speedup against BLonD++ of: a) HBLonD, b) HBLonD using reduced (32-bit instead of 64-bit) arithmetic precision, c) CuBLonD, and d) CuBLonD using reduced (32-bit instead of 64-bit) arithmetic precision. As we can see, HBLonD can effectively combine multiple computing nodes to run a single BLonD simulation, demonstrating speedups ranging from $10\times$ up to $17\times$ compared to the previous, single-node implementation of BLonD. Furthermore, CuBLonD using GPUs to accelerate the calculation of the most time-consuming regions, achieves speedups that range from $46\times$ up to $78\times$ compared to BLonD++.

Presently CuBLonD is distributed together with the main BLonD code. To conclude, BLonD is an all-round optimized simulator, that adopts state-of-art high performance computing standards, and is capable of effectively utilizing the compute resources of multiple processors and GPU cards.

IX. 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 [6], 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 behaviour in the presence of these loops.

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

A. Code-to-code comparison with the MuSiC code

Here we give an example of benchmark between the BLonD and MuSiC [48] codes and we compare the obtained results with an analytical formula in narrow-band assumption [16, 49].

For a resonator impedance with a quality factor $Q \gg 1$ and relatively low resonant frequency f_r , the wakefield can couple multiple bunches or even the same bunch on

multiple turns. If the resonant frequency f_r is close to an integer multiple p of the revolution frequency, then Robinson instability can be observed [21]. Supposing a Gaussian line density with rms bunch length σ_t , the analytical expression for the growth rate is [21]

$$\frac{1}{\tau_a} = \frac{\eta e^2 N_p}{2E_d T_{\text{rev}}^2 \omega_s} \sum_{m=\pm 1} \left(m (p\omega_{\text{rev}} + m\omega_s) \times \right. \\ \left. \times \text{ReZ}(p\omega_{\text{rev}} + m\omega_s) G_m(x) \right), \quad (22)$$

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 the studied example, the wakefield of a single bunch couples the bunch to itself over thousands of turns. The integer multiple 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 $\hat{V}_{rf} = 165$ kV, while $R_d = 100$ m. The instability growth time computed with Eq. (22) is $\tau_a = 59.3$ ms for $\sigma_t \leq 3.3$ ns and the results from MuSiC and BLonD should converge to τ_a for short bunches (no Landau damping).

The initial bunch spectrum with $\sigma_t = 3.3$ ns decays at 200 MHz whereas the resonant impedance is negligible above 1 MHz. It is then not easy to choose in BLonD the bin size Δt in time-domain, or equivalently the maximum frequency $f_{\text{max}} = 1/(2\Delta t)$ in frequency domain. In addition, the frequency step $\Delta f = 1/t_{\text{max}}$ for Fourier transforms is another important parameter, since the wakefield decays slowly and it is not evident how many turns to take into account. The time domain approach for induced voltage calculation is used in BLonD since the narrow-band resonator would require a very small frequency step, making simulations computationally heavy. Indeed, the resonator impedance is not perfectly resolved even choosing $\Delta f = 70$ Hz, which corresponds to $t_{\text{max}} = 7000 T_{\text{rev}}$. The MuSiC approach avoids all these difficulties since it operates without slices and the number of macro-particles N_M is the only parameter to be studied.

The instability growth time for $\sigma_t = 3.3$ ns was examined in MuSiC as a function of N_M (Fig.9). When increasing N_M , convergence is observed (63.0 ms) but not to τ_a , since the bunch is relatively long and Landau damping decreases the analytical growth rate. For lower bunch lengths, the growth time converges to τ_a , proving the validity of the MuSiC algorithm.

Using $\sigma_t = 3.3$ ns and $N_M = 10^6$ in BLonD, the dependence of the growth time on Δf and f_{max} was studied (Fig.10). Choosing $f_{\text{max}} = 200$ MHz to properly cover the bunch spectrum, the growth time converges to 63.0 ms for Δf approaching 70 Hz, as expected from the MuSiC simulations. A scan of f_{max} for two given values of Δf shows consistency of results, unless when

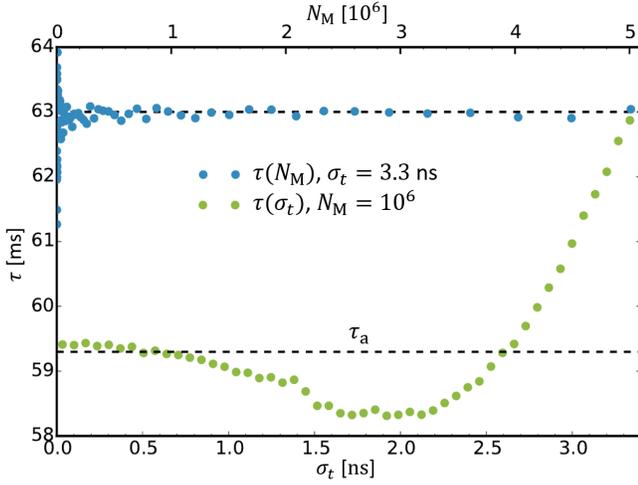


FIG. 9. Instability growth time τ as a function of N_M for $\sigma_t = 3.3$ ns (blue) and as a function of σ_t for $N_M = 10^6$ (green) from MuSiC simulations. The dashed lines mark $\tau = \tau_a = 59.3$ ms and $\tau = 63$ ms.

$f_{\max} < 50$ MHz, in which case the bunch spectrum is not properly covered and unreliable results are obtained.

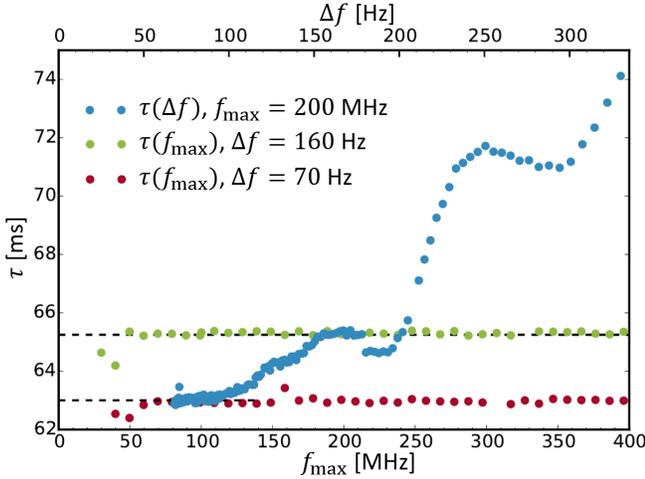


FIG. 10. Instability growth time τ as a function of Δf for $f_{\max} = 200$ MHz (blue) and versus f_{\max} for $\Delta f = 160$ Hz (green) and $\Delta f = 70$ Hz (red) from BLonD simulations ($\sigma_t = 3.3$ ns, $N_M = 10^6$). The dashed lines mark $\tau = 63$ ms and $\tau = 65.3$ ms. All the results shown are obtained through simulations in time-domain with $t_{\max} = 1/\Delta f$ and $\Delta t = 1/(2f_{\max})$.

B. Comparisons with the MELODY code

Loss of Landau damping (LLD) in the longitudinal plane is an important intensity limitation in particle synchrotrons. New features were recently discovered in this field and summarized in Ref. [50]. The new MELODY

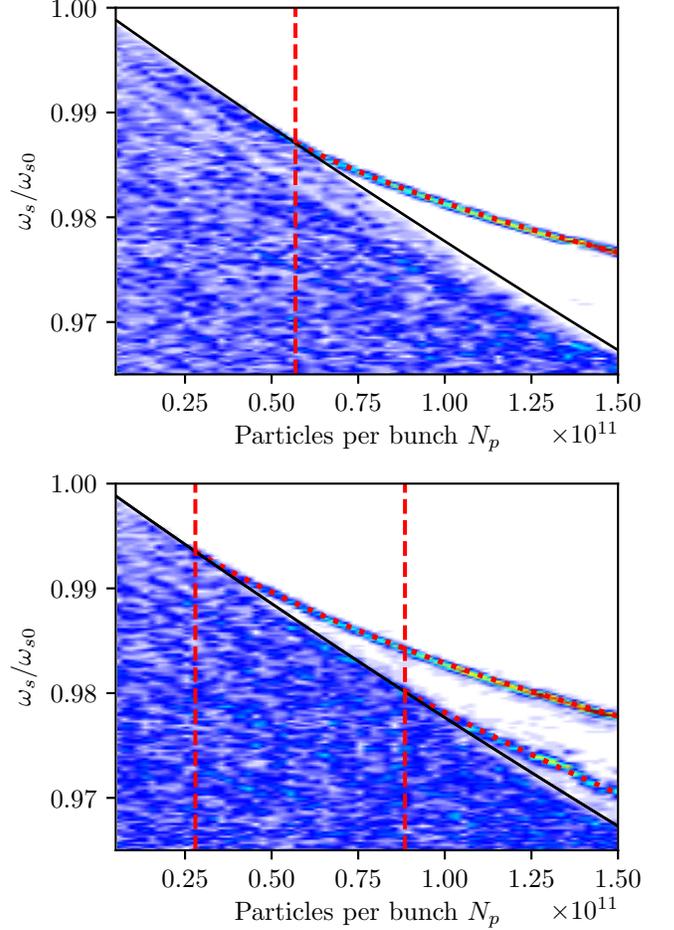


FIG. 11. The real part of the normalized mode frequency found from macro-particle simulations using BLonD (blue color) and from MELODY (red dotted line) as a function of bunch intensity for a broad-band resonator impedance with $\text{Im}Z/n = 0.07 \Omega$, $Q = 1$, $f_r = 4$ GHz (top) and $f_r = 8$ GHz (bottom). The maximum incoherent frequency obtained from MELODY is shown with black solid line. The dashed red lines indicate the LLD intensity thresholds. Plot from [50].

simulator (matrix equations for longitudinal beam dynamics) [51] was written for numerical studies of LLD. For example, it was shown that the LLD threshold inversely proportional to the resonant frequency of the broad-band impedance model with quality factor of one. This is demonstrated in Fig. 11, where we compare the emerged coherent modes that were obtained using the MELODY code and the BLonD code. The obtained results agree well.

C. Benchmark with measurements

As an application, we consider debunching measurements in the SPS [52], which are used to identify known and unknown sources of impedance in the machine. Long

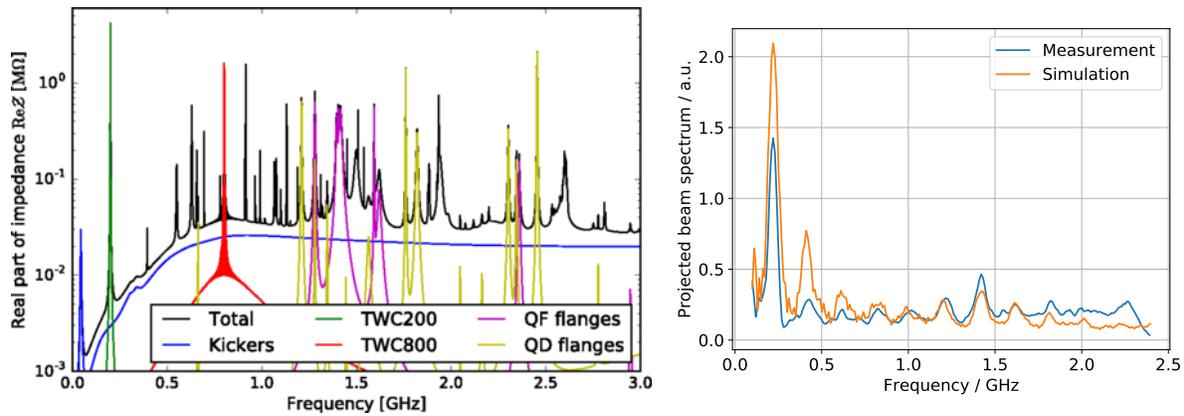


FIG. 12. The SPS impedance model (left) as of 2018 [52], before an extensive impedance reduction campaign. Simulated and measured spectra (right) of a debunching beam at SPS injection.

bunches are injected into the SPS with the rf system switched off, and the beam dynamics is 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. 12 (left) and is dominated by the impedance of the travelling 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 resonators, several travelling 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. 12 (right). The initial phase-space distribution used in the simulation is based on measured profiles and tomographic reconstruction in the PS. Both spectra in Fig. 12 (right) 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 the PS covering many rf harmonics. For the selected example shown in Fig. 13, 8 bunches are injected from the PSB (two injections of 4 bunches) in $h = 9$ and accelerated to an intermediate energy. Next, a batch compression is done to bring the beam to $h = 14$ by passing through all intermediate harmonics. Bunches are then merged into $h = 7$, before being 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 on Fig. 13, the simulation reproduces very well the measured beam evolution even for complex rf configurations.

X. CONCLUSIONS AND OUTLOOK

The Beam Longitudinal Dynamics simulation suite BLoND has been presented. Its modular structure allows the user to build custom simulations modeling a range of physics phenomena, starting from basic cavity-beam interaction, over collective effects and a wide range of rf manipulations, to synchrotron radiation and feedback systems. The generation of matched particle distributions can be widely applied. Code optimizations on various levels and for a diverse range of hardware allow the user to make faster and more memory-efficient simulations.

In the future, it is planned to optimize the algorithm for the hamiltonian distribution functions that are used for the generation of particle distributions. The implementation of different algorithms for non-uniform binning and arbitrary impedance is on-going, too. Machine-specific global and local feedback models are constantly being added for various machines and are planned to be coupled.

BLoND is a constantly evolving and developing code base. Existing modules are updated and new modules are added to better tailor the studied phenomena or extend the current scope and functionality of the code. To ensure proper behavior and shielding against coding flaws, BLoND is supported by automatic unit-testing, integration testing and coverage analysis.

[1] J. A. MacLachlan, *ESME: Longitudinal Phase Space Particle Tracking. Program Documentation*, Tech. Rep.

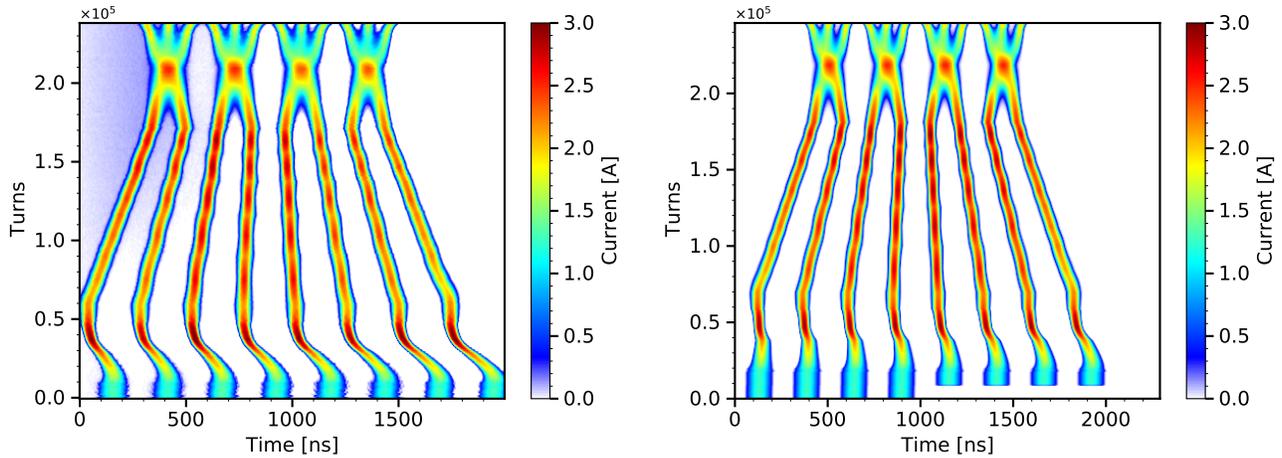


FIG. 13. Comparison of the measured (left) and simulated (right) evolution of the bunch profiles during rf manipulations in the PS. The measured phase displacement of the beam between machine turns 10000 and 50000 is due to the signal delay with changing revolution frequency during acceleration and is not included in the simulation.

- [2] (2014-2022), CERN BLonD Simulation Suite Website: <http://blond.web.cern.ch>.
- [3] (2014-2022), CERN BLonD Simulation Suite Code Repository: <https://github.com/blond-admin/BLonD>.
- [4] S. v. d. Walt, S. C. Colbert, and G. Varoquaux, *Computing in Science & Engineering* **13**, 22 (2011).
- [5] E. Jones, T. Oliphant, P. Peterson, *et al.*, “SciPy: Open source scientific tools for Python,” (2001–), [Online; accessed 2018-28-05].
- [6] H. Timko, J. Esteban Müller, A. Lasheen, and D. Quartullo, in *Proceedings, 7th International Particle Accelerator Conference (IPAC 2016): Busan, Korea, May 8-13, 2016* (2016) p. WEPOY045.
- [7] C. Ohmori, M. Furusawa, K. Hara, K. Hasegawa, Y. Sugiyama, M. Yoshii, M. Nomura, T. Shimada, F. Tamura, and M. Yamamoto, in *Proc. IPAC’17* (Copenhagen, Denmark, 2017).
- [8] 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. Wetzel (2019).
- [9] S. Krinsky and J. M. Wang, *Part. Accel.* **12**, 107 (1982).
- [10] G. Dôme, in *CERN Accelerator School ‘85, Oxford, UK* (1987).
- [11] H. Timko, P. Baudrenghien, E. Shaposhnikova, and T. Mastoridis, in *Proc. of HB2014, East Lansing, USA* (2015) p. THO4LR03.
- [12] S. Albright and D. Quartullo, *Journal of Physics: Conference Series* **1350**, 012144 (2019).
- [13] 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*, Tech. Rep. CERN-ACC-2014-0100 (CERN, 2014).
- [14] J. Tückmantel, *Digital Generation of Noise-Signals with Arbitrary Constant or Time-Varying Spectra (A noise generation software package and its application)*, Tech. Rep. LHC-PROJECT-Report-1055. CERN-LHC-PROJECT-Report-1055 (CERN, Geneva, 2008).
- [15] 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*, Tech. Rep. CERN-ACC-2016-0041 (CERN, Geneva, 2016).
- [16] D. Quartullo, *Simulations of RF beam manipulations including intensity effects for CERN PSB and SPS upgrades*, Ph.D. thesis, Sapienza University, Rome (2019), presented 08 Feb 2019.
- [17] T. Argyropoulos, T. Bohl, A. Lasheen, G. Papotti, D. Quartullo, and E. Shaposhnikova, in *Proc. IPAC’19* (Melbourne, Australia, 2019).
- [18] J. L. Laclare, in *Proceedings of CAS 1985 Vol. I* (Geneva, Switzerland, 1987) pp. 264–326.
- [19] D. Quartullo, E. Shaposhnikova, and H. Timko, *Journal of Physics: Conference Series* **874**, 012066 (2017).
- [20] F. J. Sacherer, in *the 5th IEEE Particle Accelerator Conference (PAC’73)* (San Francisco (CA), USA, 1973) pp. 825–829.
- [21] A. Chao, *Physics of Collective Beam Instabilities in High Energy Accelerators*, Wiley Series in Beam Physics and Accelerator Technology (Wiley, 1993).
- [22] B. W. Zotter and S. Kheifets, *Impedances and Wakes in High Energy Particle Accelerators* (WORLD SCIENTIFIC, 1998) <https://www.worldscientific.com/doi/pdf/10.1142/3068>.
- [23] L. Wang and Y. Li, *Phys. Rev. ST Accel. Beams* **18**, 024201 (2015).
- [24] M. Migliorati and L. Palumbo, *Phys. Rev. ST Accel. Beams* **18**, 031001 (2015).
- [25] M. Schwarz, P. Basler, M. v. Borstel, and A.-S. Müller, *Phys. Rev. ST Accel. Beams* **17**, 050701 (2014).
- [26] 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, *Future Circular Collider*, Tech. Rep. CERN-ACC-2018-0057 (CERN, Geneva, 2018) submitted for publication to *Eur. Phys. J. ST*.
- [27] M. Benedikt, M. Capeans Garrido, F. Cerutti, B. God-

- dard, 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, *Future Circular Collider*, Tech. Rep. CERN-ACC-2018-0058 (CERN, Geneva, 2018) submitted for publication to Eur. Phys. J. ST.
- [28] J. F. E. Müller, *Modification of the simulation code BLoND for lepton rings*, Tech. Rep. FCC-DRAFT-TECH-2017-001 (CERN, 2017).
- [29] S. Y. Lee, *Accelerator Physics*, 4th ed. (WORLD SCIENTIFIC, 2019) <https://www.worldscientific.com/doi/pdf/10.1142/11111>.
- [30] D. Boussard, *Beam loading*, Tech. Rep. CERN-SPS-86-10-ARF (CERN, Geneva, 1986).
- [31] M. Schwarz, H. Bartosik, E. Shaposhnikova, A. Lasheen, J. Repond, and H. Timko, in *Proc. 61st ICFA Advanced Beam Dynamics Workshop (HB'18), Daejeon, Korea, 17-22 June 2018*, ICFA Advanced Beam Dynamics Workshop No. 61 (JACoW Publishing, Geneva, Switzerland, 2018) pp. 180–185, <https://doi.org/10.18429/JACoW-HB2018-TUP2WA03>.
- [32] H. Timko, T. Argyropoulos, I. Karpov, and E. Shaposhnikova, in *Proc. 61st ICFA Advanced Beam Dynamics Workshop (HB'18), Daejeon, Korea, 17-22 June 2018*, ICFA Advanced Beam Dynamics Workshop No. 61 (JACoW Publishing, Geneva, Switzerland, 2018) pp. 163–167, <https://doi.org/10.18429/JACoW-HB2018-TUP1WA03>.
- [33] L. Medina, T. Argyropoulos, R. Calaga, and H. Timko, in *Proc. of International Particle Accelerator Conference (IPAC'21), Campinas, Brazil, 24-28 May, 2021* (2021).
- [34] L. Medina, T. Argyropoulos, P. Baudrenghien, I. Karpov, and H. Timko, in *Proc. of International Particle Accelerator Conference (IPAC'21), Campinas, Brazil, 24-28 May, 2021* (2021).
- [35] G. Dôme, *The SPS acceleration system travelling wave drift-tube structure for the CERN SPS*, Tech. Rep. CERN-SPS-ARF-77-11 (CERN, 1977).
- [36] J. Tückmantel, *Cavity-Beam-Transmitter Interaction Formula Collection with Derivation*, Tech. Rep. (CERN, 2010).
- [37] P. Baudrenghien and T. Mastoridis, *I/Q Model of the SPS 200 MHz Travelling Wave Cavity and Feedforward Design*, Tech. Rep. CERN-ACC-NOTE-2020-0032 (CERN, 2020).
- [38] P. W. Krempf, *The Abel-type Integral Transformation with the Kernel $(t^2-x^2)^{-1/2}$ and its Application to Density Distributions of Particle Beams*, Tech. Rep. CERN Note MPS/Int. BR/74-1 (CERN, 1974).
- [39] B. Schäling, *The boost C++ libraries* (Boris Schäling, 2011).
- [40] intelmkl, “Intel math kernel library,” (2018).
- [41] D. Piparo, V. Innocente, and T. Hauth, *Journal of Physics: Conference Series* (2014).
- [42] K. Iliakis, H. Timko, S. Xydis, and D. Soudris, in *Proceedings of the 18th International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation*, SAMOS '18 (ACM, New York, NY, USA, 2018) pp. 123–130.
- [43] L. Dagum and R. Menon, *IEEE computational science and engineering* **5**, 46 (1998).
- [44] K. Iliakis, H. Timko, S. Xydis, and D. Soudris, in *Proceedings of the 17th ACM International Conference on Computing Frontiers*, CF '20 (Association for Computing Machinery, New York, NY, USA, 2020) p. 29–38.
- [45] M. Snir, S. Otto, S. Huss-Lederman, D. Walker, and J. Dongarra, *MPI-The Complete Reference, Volume 1: The MPI Core*, 2nd ed. (MIT Press, Cambridge, MA, USA, 1998).
- [46] D. Kirk *et al.*, in *ISMM*, Vol. 7 (2007) pp. 103–104.
- [47] K. Iliakis, H. Timko, S. Xydis, P. Tsapatsaris, and D. Soudris, *IEEE Trans. Parallel Distrib. Syst.* **33**, 4425–4439 (2022).
- [48] M. Migliorati and L. Palumbo, *Physical Review Special Topics - Accelerators and Beams* **18** (2015), 10.1103/PhysRevSTAB.18.031001.
- [49] D. Quartullo, M. Migliorati, and J. Repond, in *Proc. of International Particle Accelerator Conference (IPAC'17), Copenhagen, Denmark, 14-19 May, 2017*, International Particle Accelerator Conference No. 8 (JACoW, Geneva, Switzerland, 2017) pp. 4465–4468, <https://doi.org/10.18429/JACoW-IPAC2017-THPVA022>.
- [50] I. Karpov, T. Argyropoulos, and E. Shaposhnikova, *Phys. Rev. Accel. Beams* **24**, 011002 (2021).
- [51] I. Karpov, Matrix Equations for Longitudinal beam Dynamics (MELODY) code, <https://gitlab.cern.ch/ikarpov/melody>.
- [52] A. Lasheen, T. Argyropoulos, T. Bohl, J. F. Esteban Müller, H. Timko, and E. Shaposhnikova, *Phys. Rev. Accel. Beams* **21**, 034401 (2018).