NUMERICAL MODELING OF FAST BEAM ION INSTABILITIES

L. Mether[*], G. Iadarola, G. Rumolo, CERN, Geneva, Switzerland

Abstract

The fast beam ion instability may pose a risk to the operation of future electron accelerators with beams of high intensity and small emittances, including several structures of the proposed CLIC accelerator complex. Numerical models can be used to identify necessary vacuum specifications to suppress the instability, as well as requirements for a possible feedback system. Vacuum requirements imposed by the instability have previously been estimated for linear CLIC structures, using the strong-strong macroparticle simulation tool FASTION. Currently, efforts are being made to improve the simulation tools, and allow for equivalent studies of circular structures, such as the CLIC damping rings, on a multi-turn scale. In this contribution, we review the recent code developments, and present first simulation results.

INTRODUCTION

Beam-induced ionization of residual gas present in the vacuum chamber of an accelerator, leads to the formation of ions and electrons along the beam path. Depending on several beam and machine parameters, such particles can accumulate into ion or electron clouds, which may cause serious beam degradation and the excitation of a two-stream instability. In electron machines with bunched beams, ion clouds build up if ions accelerated by a passing bunch do not have the time to reach the opposing chamber wall before the appearance of the subsequent bunch. In linacs, as well as in circular machines operating with a large clearing gap, the ion clouds can build up only over the passage of a bunch train, but may nevertheless lead to the development of an instability, the fast beam ion instability [1].

The Compact Linear Collider (CLIC) is a proposed TeV-scale high-luminosity linear electron-positron collider currently under study [2]. The collider could, in stages, reach a centre-of-mass energy up to 3 TeV, with roughly 20 km long main linacs. The linacs are designed to operate with trains of 312 or more short bunches with a minimum intensity of $N = 3.7 \times 10^9$, and a bunch separation of 0.5 ns. To maximize the luminosity, the transverse beam emittances, in particular in the vertical plane, need to be very small. These small emittance beams are produced in the CLIC injector complex, which consists of injector linacs, followed by two damping rings for each beam. The damping rings are designed to lower the normalized transverse emittances by several orders of magnitude from the injected values, down to the target extracted values of roughly 500 nm and 5 nm in the horizontal and vertical planes respectively. The beams are then further accelerated in a common linac before being transported through the main transfer lines to the beginning of the main linacs.

Due to the high brightness and the short bunch separation of the CLIC beams, the electron beam is at serious risk of suffering from the fast beam ion instability in several structures along the accelerator complex. Whether an instability can develop depends on if the ion species present in the chamber are trapped by the beam, and the residual gas pressure is sufficiently high to excite an instability. Studies of ion trapping, as well as assessments of pressure thresholds for the onset of instability can successfully be made using numerical models. In the past, such studies have been performed for the major linear CLIC structures: the main linac and the main transfer line, using a dedicated simulation tool, FASTION, which was developed specifically for the purpose [3]. To be able to perform similar studies for the damping rings, a number of outstanding challenges need to be addressed.

In the following section, we describe our tools and procedures for modeling fast beam ion instabilities. We outline the main issues with extending the studies to synchrotrons, and describe the modifications made to accommodate them. Subsequently, we present first results from simulation studies of the instability in the CLIC main damping ring. We also discuss in more detail some of the general challenges related to numerical studies of fast beam ion instabilities, and the strategies and solutions we have adopted in our simulation tools in order to address these. In the final section, we draw some concluding remarks.

SIMULATION TOOLS AND DEVELOPMENT

Macroparticle tracking simulations using the particle-in-cell (PIC) method, are a common tool for numerical studies of collective effects, and the one we employ in our model. Within this framework, the machine lattice is divided into a number of segments, each of which is represented by an interaction point where the electromagnetic beam-ion interaction along the segment is modeled in 2D. Ions are generated bunch by bunch, and the beam-ion interaction is simulated separately for each bunch along the train. In our studies, we use strong-strong simulation tools, *i.e.* both the beam and the ion cloud are represented by sets of macroparticles, which can be time-consuming, but offer a comprehensive model of the phenomenon. The ion macroparticles are regenerated in every interaction point, whereas the beam macroparticles, defined by their phase space variables, are transported between the interaction points using the linear transverse transfer matrices.

FASTION

The FASTION code, written in the C language, was developed at CERN specifically to study the fast beam ion instability in linear CLIC machines. It allows to study a user defined gas composition with different species, defined through their

[*] lotta.mether@cern.ch

ISBN 978-3-95450-178-6

individual mass, partial pressure and ionization cross section. Ions are generated either through scattering ionization, or, if the beam field exceeds a given threshold, through field ionization. Field ionization is estimated to take place along a large part of the CLIC main linac, greatly increasing the amount of ions generated by the passing beam [4–6]. Longitudinal dynamics are not included in the code, and each bunch is modeled as a transverse 2D distribution. This can be considered a reasonable approximation, given the short bunch length of the CLIC beams, along which the relatively heavy ion distribution would not move significantly. Acceleration is taken into account by assigning a relativistic gamma to each interaction point.

In order to adapt the FASTION code to the CLIC damping rings, it should be extended with a number of features that are generally relevant in electron synchrotrons, *e.g.* radiation damping, a model for a feedback system, detuning parameters such as chromaticity, and perhaps longitudinal dynamics. Another issue that could be interesting to consider is the effect of the applied magnetic fields on the ion motion. In the CLIC main linac, roughly a quarter of the length consists of quadrupoles, the fields of which have not been taken into account in the instability simulations. In the damping ring, quadrupoles take up a similar fraction, and in addition more than 40 % of the length is covered by dipole fields. Due to the proton mass, the effect of the magnetic fields is not expected to be crucial, as it is for electron clouds, but especially the lightest ion species, which could be trapped by the beam in the beginning of the damping process, could undergo a number of cyclotron periods during the passage of a bunch train.

Rather than proceeding with implementing the desired features into the FASTION code, however, we have adopted the approach of implementing the functionality of FASTION into another tool set, the PyECLOUD-PyHEADTAIL simulation set-up, which already includes these as well as several other potentially useful features.

PyECLOUD and PyHEADTAIL

PyECLOUD is a 2D macroparticle code modeling the build-up of electron clouds in accelerators [7]. PyHEADTAIL is a macroparticle tracking code used for simulating various collective effects [8]. The tools are used extensively at CERN for studies of both the current accelerator complex and future projects, and have each been benchmarked against experiments. Both tools are based on earlier numerical codes, ECLOUD and HEADTAIL [9] respectively, which have been rewritten using mainly the Python language, extended and updated with a modular structure to make them more versatile, maintainable and user-friendly. A more detailed description of the implementation and a discussion on the motivation and underlying principles of the design choices are provided in Ref. [8].

Owing to their modular structure and flexible nature, the two codes can be coupled together to simulate the beam dynamics of various electron cloud induced effects [10]. In this set-up, PyHEADTAIL is used to track the beam through various elements along a machine, which also contains a selected number of electron cloud interaction points distributed along the lattice. In each of these interaction points, the state of the beam macroparticles is passed to PyECLOUD, which models in detail the interaction between the beam and an electron cloud (which can be loaded from an earlier dedicated build-up simulation), and subsequently passes the updated beam phase space coordinates back to PyHEADTAIL for continued tracking.

Practical Implementation

Due to the very similar nature of electron- and ion-induced beam effects, the PyECLOUD-PyHEADTAIL simulation set-up could be applied to the simulation of fast beam ion instabilities, with relatively few adjustments. One main distinction is that the ion cloud in each interaction point should be built up from the beginning in real time along the bunch train passage, taking into account the actual position of each bunch. Below we list a few additional modifications that were required for the practical implementation:

- Generalization to arbitrary charge and mass, both for the beam and cloud particles.
- Modification of impacts in PyECLOUD. While secondary emission is an important ingredient for the electron cloud build-up, we have implemented simple perfect absorber boundary conditions for ions reaching the border of the simulation domain.
- Extension of the gas ionization model already present in PyECLOUD as an electron source, to handle the generation of different ion species.

Whereas PyECLOUD is designed to simulate electron cloud build-up along an arbitrary bunch train pattern, PyHEADTAIL had previously been used only with beams of a single bunch. In both tools, however, a bunch is typically divided longitudinally into slices, with collective interactions modeled slice by slice. For the fast beam ion instability simulations it is essential to track the full bunch train, but the interactions can be modeled bunch by bunch, using strictly transverse 2D distributions for the ion and bunch particles, as is done in FASTION. Benefiting once more from the modular design, it was thus possible to set up multi-bunch beam tracking following a similar structure, slicing the beam into bunches for the ion-beam interaction and applying a single kick per bunch.

In principle, it would be possible with only minor additional adjustments, to add a second layer of slicing to resolve the longitudinal bunch structure in the interactions. We have currently not explored this option, however, since we expect the effect to be relatively insignificant, whereas it would significantly increase the execution time, which is already a limiting factor without the additional operations.

On the other hand, enabling PyECLOUD to run with ions in place of electrons also opens up the possibility of dedicated ion build-up studies using a rigid beam field map, and separate instability simulations based on their output, which is the current approach for electron-cloud-induced instability studies. This makes it possible to study and estimate the effect on the beam of ion trapping and the build-up of ion clouds over multiple turns, which would currently be computationally too demanding for most practical applications using strong-strong simulations. Such studies could be of interest *e.g.* for light sources operating with short clearing gaps, which may become more sensitive to ion-induced instabilities due to future upgrades to smaller beam sizes.

APPLICATION TO CLIC DAMPING RING

Table 1: Main Damping Ring Parameters in Simulations

Parameter	**Symbol, unit**	**Value**
Energy	E [GeV]	2.86
Bunch population	N [10^9]	4.1
Norm. horizontal emittance	$\gamma\epsilon_x$ [µm]	472
Norm. vertical emittance	$\gamma\epsilon_y$ [µm]	4.8
Bunch length (r.m.s.)	σ_s [mm]	1.6
Bunch spacing	ΔT_b [ns]	0.5
Bunches per train	N_b	312
Circumference	C [m]	427.5

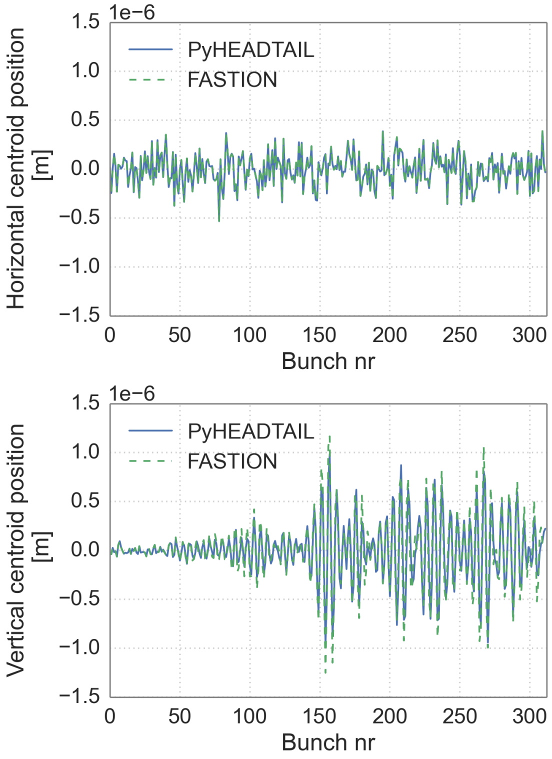

Figure 1: Horizontal and vertical bunch-by-bunch centroid position after one turn of tracking with P = 20 nTorr.

To test the new implementation of ion-induced instabilities within PyECLOUD and PyHEADTAIL, we have set up some simulations for the CLIC main damping ring, to benchmark the results against FASTION simulations. For this purpose we use a simple simulation scenario, excluding any of the effects that motivated the recent development, since these are not available in FASTION. The main parameters used for the study, corresponding to the equilibrium values for the damping ring, are summarized in Table 1. The residual gas consists of a single species, water vapor, with the mass number A =18, which is the lightest species expected to be trapped for the chosen beam parameters.

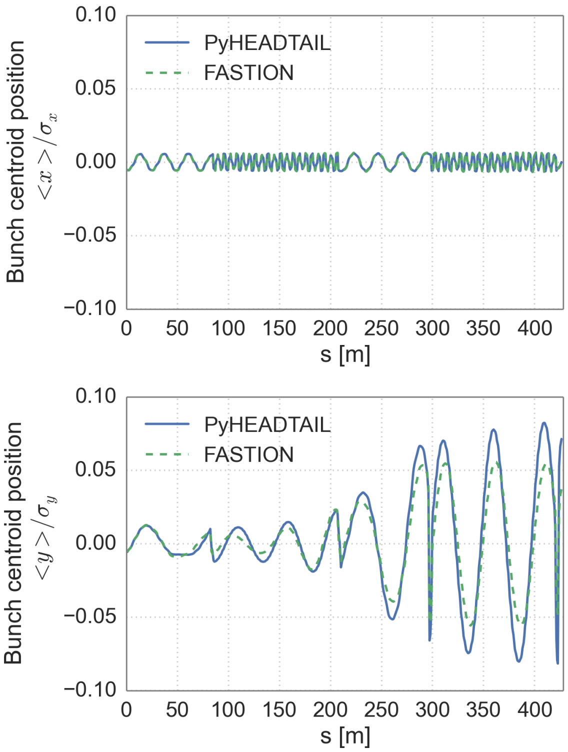

Figure 2: Evolution of the horizontal and vertical centroid position of the last bunch of the train over one turn in the machine with P = 20 nTorr.

In the first study, a bunch train is initialized with identical macroparticle phase space coordinates in each code. The train is tracked for one turn in the machine, which is divided into 677 interaction points, each representing a segment roughly 60 cm long. The residual gas pressure is set to 20 nTorr, which is sufficiently high to induce a growth of unstable motion in a single turn. The transverse bunch-by-bunch centroid position of the full train after one turn of tracking with each code is shown in Fig. 1. There is a clear growth of unstable motion in the vertical plane, whereas the horizontal plane is unaffected, as expected for a flat transverse beam profile. Figure 2 displays the evolution over one turn in the machine of the centroid position of the last

ISBN 978-3-95450-178-6

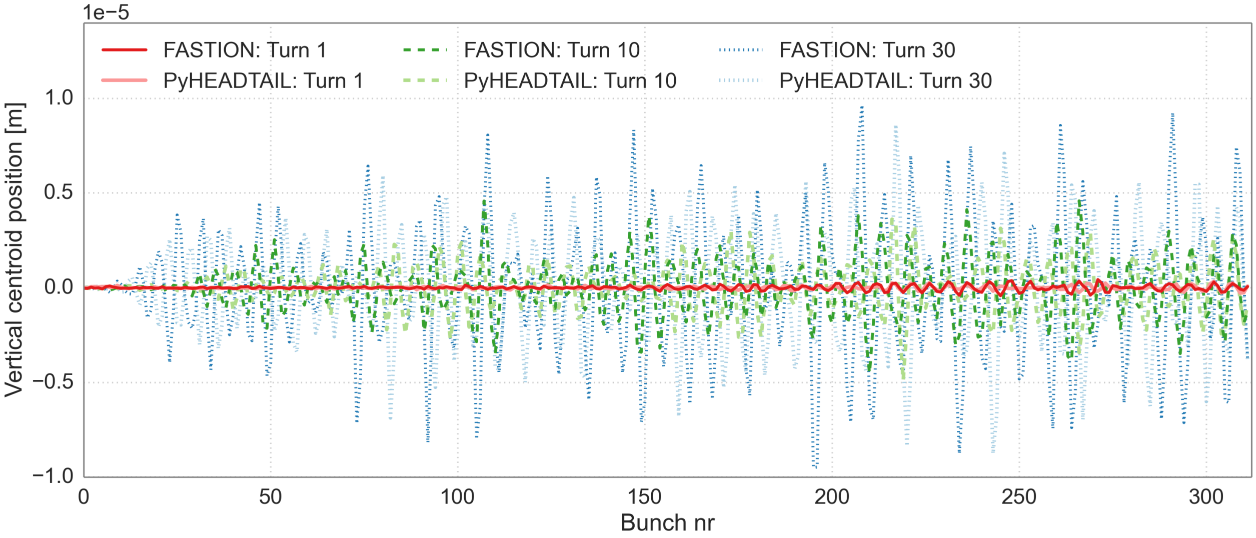

Figure 3: Vertical bunch-by-bunch centroid position after 1, 10 and 30 turns of tracking with P = 10 nTorr.

bunch of the train. In both cases, the results obtained with the two different tools are in good agreement. We believe that the small discrepancies can be attributed to the randomness in the explicit coordinates of the ion distribution generated around each bunch.

A second benchmarking study was performed, with the phase space coordinates of the bunch train macroparticles initialized with different random jitter seeding the instability in each code. The bunch trains were tracked for a hundred turns, with the machine lattice divided into 260 interaction points and the gas pressure set to 10 nTorr. Snapshots of the bunch-by-bunch centroid positions of the full train after one, ten, and thirty turns are displayed in Fig. 3. Even though initialized with different seeds, it can be seen that the growth of unstable motion for a given bunch along the train evolves at a similar pace with respect to the number of executed turns in both cases. A similar agreement can be seen also on the vertical emittance growth of the last bunch of the train, shown in Fig 4.

ASPECTS ON PERFORMANCE AND ACCURACY

One of the main challenges with using a strong-strong simulation tool for modeling the fast beam ion instability is the relatively large computational load of each interaction, resulting in long execution times. This becomes a serious limitation especially for studies of the damping rings, where it would be desirable to model the evolution of the instability during the entire damping period, or at least a considerable part at a time, since both the species of ions trapped by the beam and the dynamics responsible for initiating the instability depend crucially on the beam size.

In the CLIC main damping ring the transverse damping time is 2 ms, corresponding to roughly 1400 turns. With a sufficiently small distance between interaction points, which we consider to be ≤ 2 m, simulating one turn around the ring currently takes 20 minutes with FASTION, yielding a run time of 20 days for a full damping time. The PyECLOUD-PyHEADTAIL implementation currently exhibits execution times roughly 50 % slower than FASTION, after some initial optimization. Code profiling studies have shown that a large part of this difference comes from the FFT calculations used in the PIC solver, on which some further optimization is still required. Moreover, to improve the run-time performance further, efforts are under way to create a layer of parallelization that can be used for the ion studies, as well as for electron cloud and other multi-bunch studies.

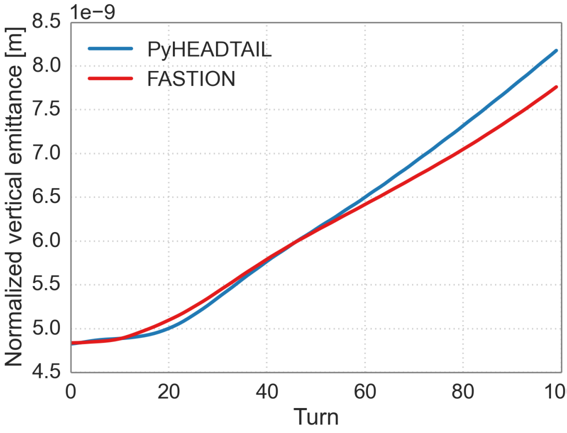

Figure 4: Evolution of the normalized vertical emittance of the last bunch of the train over 100 turns with P = 10 nTorr.

For particle-in-cell simulations of two-stream instabilities in general, it can be challenging to resolve accurately the electric field of the beam, with a simulation domain spanning

ISBN 978-3-95450-178-6

the size of the surrounding cloud, or even the entire beam chamber. This is especially true for electron and positron machines, which typically have much smaller beam sizes compared to the chamber radius than hadron machines. For fast beam ion instability studies it is particularly important to have a good resolution of the electric fields within the beam, since the variations in the kick given to particles at slightly different locations around the centre of the beam, are an important ingredient in exciting the instability.

Simply increasing the number of PIC grid cells representing the full simulation domain easily leads to unacceptably long execution times, and may require excessive amounts of memory. For very flat transverse beam profiles typical in lepton machines, using rectangular grid cells may be beneficial, but *e.g.* in the case of the CLIC main linac where the beam size becomes extremely small, we found it not to be a sufficient measure to obtain the required resolution with a reasonable execution time. The problem could be addressed through the use of multiple nested grids, such that the grid resolution can be increased around the beam where it is needed, but kept at a lesser resolution elsewhere. In FASTION, a dual-grid solution using an internal, fine grid spanning the area around the beam, and a coarser grid spanning the full simulation domain for ions outside of the beam was implemented. A similar solution, but allowing for an arbitrary number of grids is currently being implemented in PyPIC, the PIC module used by PyECLOUD.

CONCLUSION

In order to ensure that the operation of the damping rings in the proposed CLIC accelerator complex will not be affected by the fast beam ion instability, systematic simulation studies are needed to determine viable vacuum and feedback specifications. To this end, the functionality of the FASTION code has been implemented in a new simulation set-up using the existing simulation codes PyECLOUD and PyHEADTAIL, which include all the necessary features for studying synchrotrons. First studies for the main damping ring have been performed, and the results have been successfully benchmarked against FASTION. Making use of the many features available in the new simulation set-up, the effects on the instability of several different elements can now be estimated, such as the self-space charge of the ions, the electric field boundary for various beam chamber profiles, energy or position dependent detuning and the movement of ions in bending and focusing magnetic fields. Another significant benefit is the shared workload of maintenance and development, which is already bearing fruit in the currently ongoing developments on parallelization and optimization of the PIC methods.

REFERENCES

[1] T. O. Raubenheimer and F. Zimmermann, “Fast beam-ion instability. I. Linear theory and simulations”, *Phys. Rev. E*, vol. 52, no. 5, pp. 5487–5498, Nov. 1995.

[2] M. Aicheler *et al.*, “A multi-TeV linear collider based on CLIC technology: CLIC conceptual design report”, CERN, Geneva, Switzerland, Report CERN-2012-007, Oct. 2012.

[3] G. Rumolo and D. Schulte, “Fast ion instability in the CLIC transfer line and main linac”, in *Proc. 11th European Particle Accelerator Conf. (EPAC’08)*, Genoa, Italy, Aug. 2008, pp.655-–657.

[4] G. Rumolo and D. Schulte, “Update on fast ion instability simulations for the CLIC main linac”, in *Proc. 23rd Particle Accelerator Conf. (PAC’09)*, Vancouver, Canada, May 2009, pp. 4658–4660.

[5] J.-B. Jeanneret, G. Rumolo, and D. Schulte, “Vacuum specifications for the CLIC main linac”, in *Proc. 1st Int. Particle Accelerator Conf. (IPAC’10)*, Kyoto, Japan, 2010, pp. 3401–3403.

[6] A. Oeftiger and G. Rumolo, “Fast beam-ion instabilities in CLIC main linac: vacuum specifications”, CERN, Geneva, Switzerland, Rep. CERN-OPEN-2011-050, Nov. 2011.

[7] G. Iadarola, “Electron cloud studies for CERN particle accelerators and simulation code development”, Ph.D.thesis, CERN-THESIS-2014-047, 2014.

[8] K. Li *et al.* “Code development for collective effects”, presented at the 57th ICFA Advanced Beam Dynamics Workshop on High-Intensity and High-Brightness Hadron Beams (HB2016), Malmö, Sweden, July 2016, paper WEAM3X01.

[9] G. Rumolo and F. Zimmermann, “Practical user guide for HEADTAIL”, Rep. CERN-SL-Note-2002-036-AP, 2002.

[10] G. Iadarola, A. Axford, H. Bartosik, K. Li, and G. Rumolo, “Effect of electron cloud in quadrupoles on beam instability”, *Proc. 6th Int. Particle Accelerator Conf. (IPAC’15)*, Richmond, VA, USA, May 2015, pp. 409–411.

ISBN 978-3-95450-178-6