# RECENT IMPROVEMENTS IN THE TRACKING CODE PLACET*

A. Latina, E. Adli, H. Burkhardt, G. Rumolo, D. Schulte, R. Tomas, CERN, Geneva, Switzerland
Y. Renier, LAL, Paris, France

*Abstract*

The tracking code Placet has recently undergone several improvements. A redesign of its internal data structures and a new user interface based on the mathematical toolbox Octave considerably expanded its simulation capabilities. Several new lattice elements, optimization algorithms and physics processes were added to allow for more complete start-to-end simulations. Finally, the use of the AML language and the Universal Parser Library have extended its interfacing capability. A review of these new features is presented in this paper.

## EMBEDDING OF OCTAVE

The tracking code Placet has been recently extended to embed the Octave's interpreter [1]. Octave is a high-level interactive language for numerical computations that is mostly compatible with MatLab©. Octave can do arithmetic for real and complex scalars and matrices, solve sets of nonlinear algebraic equations, integrate functions and has a large set of tool-boxes and optimization routines.

The version of the code that includes Octave is called Placet-Octave. Placet-Octave is based on the original Tcl/Tk core but embeds the Octave's command line interpreter and can run scripts where both Tcl/Tk and Octave's instructions are present. The two interpreters share the same data structures in memory, so there is no overhead due to data exchange when passing from one environment to the other.

Placet-Octave offers to the user all Octave's commands, libraries and tool-boxes that are available for the stand-alone version of Octave, plus several commands that have been specifically created to run the Placet simulations. Some of these commands are built-in the code, as fast C++ routines, others are written as Placet-Octave's scripts themselves. This is transparent to the user. Placet-Octave's commands can be divided into two categories: commands that allow the user to inquire and modify the status of a beamline and commands that allow to run specific Placet-actions, such as tracking the bunches, or minimizing the final emittance in presence of imperfections.

The user can access Placet-Octave's functions from a Tcl script using the keyword `Octave`. This keyword has two ways of operation:

- `Octave`
  (without any argument) interrupt the script execution and open an interactive Octave command line interface
- `Octave { here some octave code }`
  execute a batch of Octave's commands and continue the execution of the script

Here is an example of a Placet-Octave call in Tcl:

```
Octave {
  # calculate the response matrix
  R0 = placet_get_response_matrix("beamline", BI, CI);

  # track a bunch through a misaligned beamline
  placet_test_no_correction("beamline",
    "beam",
    "misalign");

  # read the bpms, calculate the correction and correct
  Bpm = placet_get_bpm_readings("beamline", BI);
  Corr = R0 \ Bpm;
  placet_vary_corrector("beamline", CI, Corr);
}
```

Here, $BI$ and $CI$ are vectors containing the list of bpms and correctors, respectively; this code performs a one-to-one correction. A full list of Placet-Octave's commands is available in the Placet's manual [2].

## NEW COMMANDS

Several new keywords have enriched the Placet's set of commands.

### *New Lattice Elements*

`CrabCavity`. It simulates a crab cavity taking into account wakefield effects. Input parameters are voltage, frequency and phase of the cavity, plus a set of wakefield modes. For an example, see Fig. 1 and [3].
`DynamicElement`. Through a specific application programming interface, based on the C language, this command allows to use an external shared library as an element type to be inserted in the lattice.

### *New Commands*

`TestRfAlignment`. This new alignment routine reduces the emittance growth due to wakefield effects in the accelerating cavities by moving the supports of the girders where the cavities are placed on. Girders are moved in order to minimize the sum of the squared positions read in the cavity position monitors. Fig. 2
`BeamlineSaveAML`. It saves the beamline in Accelerator Markup Language (AML) file format. AML is a specification of the XML language designed to be used with

* Work supported by the Commission of the European Communities under the 6th Framework Programme "Structuring the European Research Area", contract number RIDS-011899.

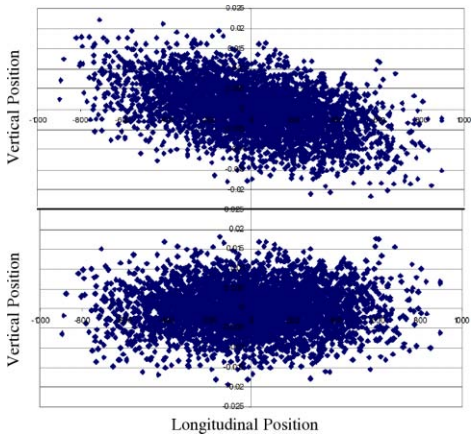


Figure 1: Removal of spurious vertical bunch rotations at the IP, caused by tilted crab cavities. Upper plot shows the $Y - Z$ phase space at the IP of the ILC when the crab cavity has a 10 mrad roll misalignment. Lower plot shows the phase space profile after the anti-crab cavity correction.

the Universal Accelerator Parser (UAP) [4]. The UAP is a project of the Cornell University to simplify the data exchange among the most diffused codes (XSIF and MAD deck files are supported, for instance).

`LinkElements`. This command links two elements of the lattice so that they are rigidly integral with each other; that is, any misalignment applied to one of the two will be propagated to its partner as if they lied on the same support. This command is useful when converting *composite* elements, such as QUADBPMs, from MAD files, because such elements are usually divided in three parts during the conversion:

$$\boxed{\text{QUADBPM}} = \underbrace{\boxed{\text{1/2 QUAD}} + \boxed{\text{BPM}} + \boxed{\text{1/2 QUAD}}}_{\text{elements to be linked}}$$

`GetTransferMatrix`. It returns the transfer matrix of a beamline (or a part of it).

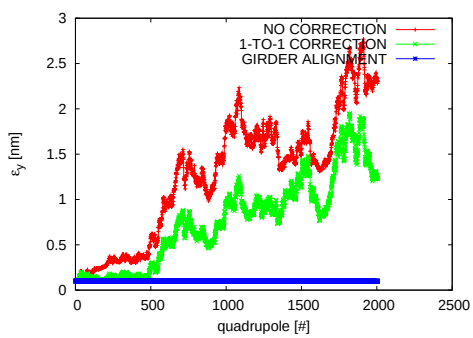


Figure 2: Successful example of girder alignment in the CLIC main linac. The red line shows the emittance growth due to cavity vertical misalignment (threfore purely due to wakefields effects) without correction. Emittance is recovered after a 1-to-1 correction (green line) and a girder alignment (blue line) as described in the text.

### *New optimization routines*

Placet-Octave allowed to access all optimization routines available for Octave. Making use of the Octave's function `fmins()`, that finds the minimum of a function of several variables using the Nelder&Mead Simplex algorithm, a new commands has been created:

```
placet_optimize("beamline",
  "merit_function",
  "correctors",
  "leverages", ["constraints"]);
```

This routine minimizes an arbitrary merit function provided by the user, changing arbitrary attributes (leverages) of a set of correctors. This optimization routine has been used for

- sophisticated Twiss parameters matching with emittance growth reduction (see the case of the CLIC injector linac, where the large energy spread of the bunch caused a considerable emittance growth, when applying the standard matching procedures; [5] and Fig. 3).
- non linear alignment techniques (see the algorithm proposed for the static alignment of the CLIC beam delivery system in [6])

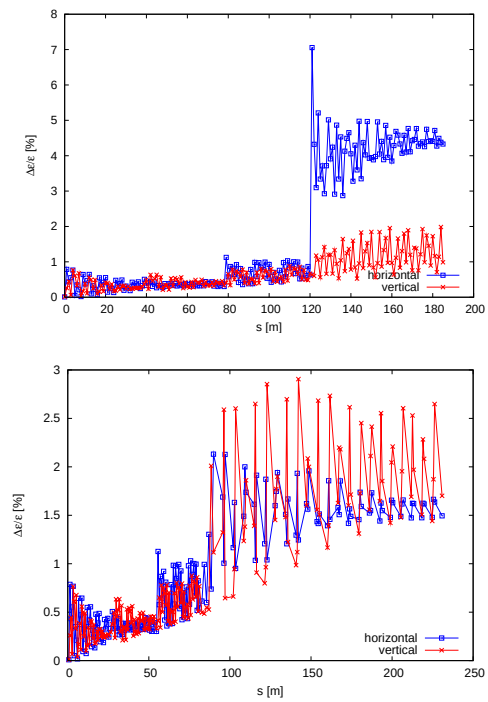


Figure 3: Twiss parameters matching using the new Placet's optimization functions. Upper plot shows the emittance growth when using a lattice uniquely matched using MAD. Lower plot shows the emittance growth after a re-matching that takes into account the emittance, using Placet.

## NEW PHYSICAL PROCESSES

### *Coherent Synchrotron Radiation Emission*

A module for the simulation of Coherent Synchrotron Radiation emission (CSR) has been implemented. The methodology follows closely the implementation of Elegant [7], which is based on the the work of [8]. This methodology uses a 1D CSR model and does not take into account shielding, but it has the advantage that it correctly calculates the transient build-up of the CSR in a bend. The Placet model has currently implemented CSR in the bending magnets, plus a simple field exponential decay model for the drift spaces(update to better models is planned in the near future). The module is very easy to use, as CSR is enabled simply by setting a parameters for the `SBend` element. A throughout benchmarking of the code against Elegant was performed (with F. Stulle, PSI). The results showed excellent agreement between Placet and Elegant for CSR in bends and at high energies (for energies below $\sim$ 100 MeV, the results differs substantially due to the $v = c$ assumption of Placet). See Fig. 4.

## INTERFACING PLACET

Placet has also been extended in its interfacing capabilities. Its interface toward the the halo and tail generation package HTGEN [9] has recently been streamlined and does no longer require external libraries.

### *Reading AML Deck Files*

Besides the new command `BeamlineSaveAML`, that writes the beamline on disk in AML format, a tool to read AML deck files into Placet has also been created. This opens the possibility to exchange beamline information with any code making use of AML. Tools converting from MAD, XSIF and BMAD into AML already exist.

### *Placet-BDSIM*

In order to simulate the impact of halo particles on the beam pipe walls, under the effect of kicks due to the collimator wakefields or other imperfections, Placet has been interfaced to BDSIM [10]. BDSIM is a code designed to track single particles and their secondaries deriving from the interactions with the materials, based on Geant4.

Combining the abilities of BDSIM and Placet allows to achieve an accurate simulation of the secondary particles generation in the collimators and their tracking. The interface is based on the following schema: tracking is performed in parallel by both codes. BDSIM tracks the halo particles and simulates the emission of secondary particles in the collimators, while Placet tracks core and halo, including secondary particles. At each collimator's entrance and exit, the two codes synchronize each others knowledge of the beam. A first application of this combo can be found in [11].

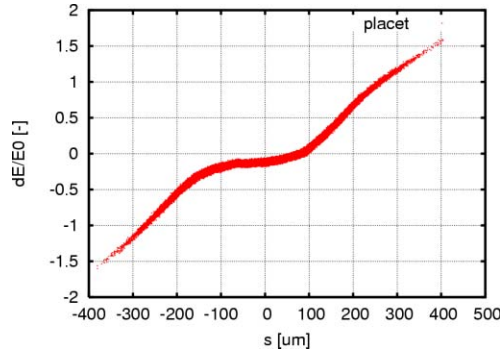


Figure 4: PLACET calculations of CSR in the PSI benchmark chicane.

## CONCLUSIONS

The tracking code Placet is reaching its maturity: the user interface has been improved to support Octave's instructions, simplifying the writing of even more complex simulation scripts, new elements and new physical effects have been added, allowing to perform more complete start to end simulations. For instance, Placet can now take into account Coherent Synchrotron Radiation emission, that might be relevant in the sections of the accelerator where the energy is low, such as the RTML; as well as it can properly simulate and track the secondary particles emitted in the collimation system of the beam delivery of a future linear collider. Its interfacing capabilities have also been expanded, opening for further, future, extensions.

## REFERENCES

[1] J.W. Eaton et al., `http://www.octave.org`

[2] D. Schulte et al., `http://savannah.cern.ch/projects/placet`

[3] G.Burt et al., "Anti-Crab Cavities for the Removal of Spurious Vertical Bunch Rotations Caused by Crab Cavities"; *submitted for publication on PRST-AB*

[4] D. Sagan et al., "The Accelerator Markup Language and the Universal Accelerator Parser", Proceedings EPAC 2006, Edinburgh, Scotland (2006); `http://www.lns.cornell.edu/ dcs/aml`

[5] A. Ferrari et al., "Beam Dynamics Studies in the CLIC Injector Linac"; CERN-AB-2007-071; CLIC-Note-723

[6] R: Tomas, et al., "Alignment of the CLIC BDS"; *these proceedings*

[7] M. Borland; Argonne National Laboratory Advanced; Photon Source Report No. LS-287, 2000.

[8] E. L. Saldin et al., "On the coherent radiation of an electron bunch moving in an arc of a circle"; NIM A 398 (1997) 392

[9] H. Burkhardt et al., `http://hbu.home.cern.ch/hbu/HTGEN.html`

[10] G. Blair et al., "The BDSIM Toolkit", 2006, EUROTeV-Report-2006-014

[11] S. Malton et al., "Full Simulation of CLIC and ILC Collimation System"; *these proceedings*