

bi - Beam Instability Code

Ivan Bazarov
Cornell University, Ithaca, NY, 14853

March 1, 2003

Multipass, multi-bunch beam break-up (BBU) has been long known to be a potential limiting factor in the operation of high current linac-based recirculating accelerators. Several novel applications based on energy recovery linac (ERL) concept demand currents much higher than what has been previously demonstrated, therefore, proper modeling of BBU instability is of importance. The `bi` code (stands for beam instability) uses a general algorithm that allows one to simulate either transverse or longitudinal instability, as well as transient effects for arbitrary bunch patterns due to interaction with higher order modes (HOM) of the accelerator. This document explains how to set up a simulation using the code.

1 Program structure

`bi` sets up a physical model of the linac in which the bunch train sees a string of cavities and HOM voltages are updated in a consistent manner with proper time delays, transfer maps taken into account. The code can be obtained from <http://www.lepp.cornell.edu/~ib38/bbu/>, and compilation instructions are found in `readme` file along with a brief explanation of how to use the code. There are two executable files: `bi` is the actual tracking code, and `findbi` is a utility that allows one to find the threshold of current at which the instability starts to occur, which is achieved by multiple calls of the `bi` program.

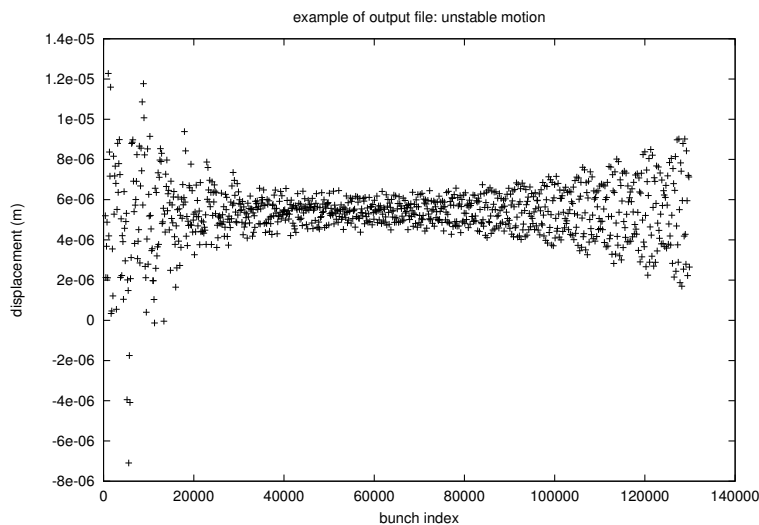


Figure 1: Example of output file

- **synopsis:**

```
bi input [-i current] [-c output] [-w output] [-v]
```

- **example:**

```
bi calib.in -c coor.out -w wake.out
```

This will call input file `calib.in` and will write tracking results to file `coor.out`, which has a format: bunch index, transverse coordinate, transverse momentum. An example of the output file is shown in the Figure 1. `-w` option writes beam-induced voltage to a file, i.e. output file will have the format: bunch index, voltage. `-v` option prints diagnostics messages to ensure that input file was set up properly. **Note:** beam current, if specified in the command line with option `-i` or with `findbi` (see below), is given in *milliamperes*.

- **synopsis:**

```
findbi input current
```

- **example:** `findbi calib.in -c coor.out -w wake.out`

2 Input file format

Refer to `calib.in` input file. Comment lines starting with exclamation mark `!` are ignored. Whitespaces are also ignored, only the order of numbers is important. The program checks for consistency whether the number of entries matches the expected number of variables. Program terminates if the two don't match. It is a good idea to always check a new input file with `-v` verbose option of `bi`.

Units for all entries are as follows: [charge] C , [distance] m , [frequency] Hz , [momentum] eV/c , [time] s , [R/Q] Ω .

The first number specifies *ALL homs*. This is a number of cavities that a bunch sees in its lifetime (most of them more than once). E.g. if a linac consists of 2 cavities, and there are two recirculations, this number would be 4 (at the moment, only the simplest two-pass case is realized. Implementing of an arbitrary topology of recirculator is straightforward. Bug the author if needed). *Number of hom entries per GLOBAL hom* simply allows one to enter more than one HOM per cavity. E.g. in the example above, suppose that for each of the 2 cavities, one wishes to include 6 HOMs. Then, the number 6 would specify to the program that each *cavity* has a list of HOMs associated with it.

Randomization of frequency allows one to apply uniform random spread to HOM frequencies from cavity to cavity. *Displacement of homs* generates uniform random offset of the axis of symmetry of the HOMs. *Noise amplitude* and *time noise is on* specifies initial beam fluctuations that trigger the instability.

Transfer matrices must be supplied in the input file. They should map the beam from cavity to cavity only, i.e. there should be *number of ALL homs* - 1 of them. Use your favorite optics code to generate these. **Note:** matrices should be in units of $m/(eV/c)$ for m_{12} and $(eV/c)/m$ for m_{21} .

3 Contact Information

E-mail: bazarov@cornell.edu

There is a version of `bi` that allows to simulate longitudinal BBU as well. It runs slower but the algorithm is general enough to allow setting up of a cloud of particles to model situation when a point charge approximation of the bunch fails to properly describe beam-cavity interaction. Contact me if you need it. Additional information about physics that went into the code is available at <http://www.lepp.cornell.edu/~ib38/bbu/doc/>.