BNL-102228-2014-TECH
RHIC/AP/120;BNL-102228-2013-IR

The Particle Tracking Program SIXTRACK on the RAP SUN Computers

W. Fischer

December 1996

Collider Accelerator Department
Brookhaven National Laboratory

U.S. Department of Energy
USDOE Office of Science (SC), Basic Energy Sciences (BES)

DISCLAIMER

The Particle Tracking Program SIXTRACK on the RAP SUN Computers

Wolfram Fischer

December 19, 1996

Abstract

This note describes the use of the particle tracking program SIXTRACK on the RAP SUN computers. An installation program prepares job submission and output areas. Job preparation and execution and the use of postprocessing tools are explained. Two examples, for the SIXTRACK tracking and the DA version respectively, are given as an illustration. All SIXTRACK related directories and files are listed and briefly described.

1 Introduction

The tracking code SIXTRACK [1–3] is a simulation program for the particle motion in accelerators and storage rings. It is optimized to carry pairs of particles over large turn numbers for the detection of chaotic motion and loss borders. The input and output format of the program and its many features are described in detail in the user's reference manual [1] and are not the subject of this note.

Here, we will describe the use of SIXTRACK and its auxiliary programs on the RAP SUNs. Installation of job submission and output areas, job preparation an execution and the use of postprocessing tools is shown.

There are two program versions to be distinguished. One version is used for element-by-element tracking over up to some million turns. Another version (the DA version) is used to produce one-turn-maps for realistic lattices (the same lattices as for element-by-element tracking). One-turn-maps give the phase space coordinates after one turn as function of the initial phase space coordinates and possible parameters (like corrector strengths) in the form of Taylor expansions up to a certain order. There are run examples for both program version.

2 Installation

To set up a job preparation (`$USER/track/sixjobs`) and output (`/track/mac/97/$USER`) area, run the program `sixinstall`. It will create those directories, copy a number of files in the job preparation area (which may serve as input example) and create two example outputs in the output area. The full directory structure and file names are given in Sec. 3.

3 Directories and Programs

The following is a list of SIXTRACK related directories and files.

`/rap/lattice_tools/bin/SUN/...`	directory for executable programs
`.../mad8six`	conversion program from MAD survey file to SIXTRACK input file fort.2
`.../six`	shell script for interactive SIXTRACK run (vector version)
`.../sixinstall`	installation program
`.../sixnyc`	shell script that finds unfinished SIXTRACK jobs
`.../sixpaw`	program that converts SIXTRACK turn-by-turn output into a paw++ input file
`.../sixtrack`	SIXTRACK executable (vector version)
`.../sixtrack_da`	SIXTRACK executable (DA version)
`.../tune`	computes the tune as function of amplitude or momentum from SIXTRACK turn-by-turn output
`/rap/lattice_tools/six/...`	directory for source code and long-term data storage
`.../maps/`	directory related to map manipulations
`.../sixdata/...`	SIXTRACK input files storage
`.../hera94/...`	SIXTRACK input files for the HERA-p lattice 1994
`.../fort.16`	SIXTRACK fort.16 input file
`.../fort.2`	SIXTRACK fort.2 input file
`.../fort.3`	SIXTRACK fort.3 input file
`.../fort.21`	sixrmul fort.21 input file (multipole order)
`.../fort.22`	sixrmul fort.22 input file (geometric errors at 3000 A)
`.../fort.23`	sixrmul fort.23 input file (persistent current errors and time dependence at 250 A)
`.../fort.3.post`	SIXTRACK fort.3 input file for postprocessing only
`.../sixdoc/`	directory that contains documentation
`.../sixpost/`	directory that contains the source code for **sixpaw** and **tune**
`.../sixtools/`	directory that contains the source code for **mad8six**, **sixmad** and **sixrmul**
`.../sixtrack/...`	directory that contains the source code for SIXTRACK
`.../source_sixda/`	source code for SIXTRACK DA version
`.../source_sixve/`	source code for SIXTRACK vector version
`.../utilities/`	directory that contains the source code for **six_job** and **sixda_job** as well as the shell scrips **six_run** and **sixda_run**
`$USER/track/...`	directory for tracking jobs
`.../sixjobs/...`	start directory for SIXTRACK jobs
`.../data_cell/`	storage directory for SIXTRACK input files (SPS FODO cell)
`.../data_hera/`	storage directory for SIXTRACK input files (HERA-p)
`.../data_post/`	storage directory for SIXTRACK input file (postprocessing)
`/track/mac/97/$USER/...`	output directory for SIXTRACK jobs
`.../example_sixda/`	output example of a SIXTRACK DA job
`.../example_sixve/`	output example of a SIXTRACK tracking job

4 Job Preparation and Execution

For a typical tracking run (for example for HERA) the following input files are needed:

`fort.2`	lattice information
`fort.3`	tracking parameters
`fort.16`	multipole errors

The most important output files are

`fort.6`	summary of all operations (ascii format)
`fort.90-k`	turn-by-turn phase space coordinates for the kth pair (binary format)

The input files have to be put into the `$USER/track/sixjobs` directory. Examples of input files can be found in `/track/mac/97/$USER/example_sixda`, `/track/mac/97/$USER/example_sixve` or `$USER/track/sixjobs/data_*`. The fort.16 input file for the HERA proton ring, containing the multipole error information, can be created by the `sixrmul` program. This program needs as input the files `fort.21` (multipole order), `fort.22` (geometric errors) and `fort.23` (persistent current errors).

When the prepared input files are in `$USER/track/sixjobs`, a job can be prepared and executed. To do this the `six_job` program can be used. If the `fort.3` input files contains a `DIFF` block (cf. Ref. [1]) for Differential Algebra operations (the one-turn-map computation), the SIXTRACK DA program version must be used by running `sixda_job` instead.

In `six_job` or `sixda_job` a jobname has to be given. A directory with this name will be created in the output area and the input files will be copied into this directory. In addition, a shell script (`six_run` or `sixda_run`) is copied into this directory with which the job can be executed. Execution is started from `six_ll` or `sixda_ll` when desired.

5 Postprocessing Tools

Once a job is finished turn-by-turn data can be further postprocessed. One possibility is to use SIXTRACK's internal postprocessing capability by changing the `fort.3` input file into one that only performs postprocessing (an examples is `$USER/track/sixjobs/data_post/fort.3`). SIXTRACK can then run interactively in the output directory with the shell script `six`.

The program `sixpaw` produces a file `out.hbook` from the turn-by-turn data. The n-tuples in this file (phase space coordinates as a function of the turn number) can be viewed with `paw++`. Execute `paw++`, load the file `out.hbook` in the `Main Browser` and click on `LUN1`. Click then on one of the icons in the `Main Browser` window (each icons contains the data for one particle) and plot by choosing a x, y and maybe a z coordinate from the list. The `Plot` button will produce the plot. For details about `paw++`, see the users manual.

With `tune` the tune can be computed as a function of amplitude or momentum. For this the data must have been analyzed with the `POST` block of SIXTRACK's internal postprocessor already. The result of `tune` will be written on file `fort.1` as a table.

6 Examples

Two examples may serve as illustration. The directory
`/track/mac/97/$USER/example_sixve` contains the output of a tracking run using input files for HERA-p. There 10 particle pairs have been tracked for 1000 turns. The run can be reproduced by copying `fort.2`, `fort.3` and `fort.16` into `$USER/track/sixjobs` and execute `six_ll` in the same directory (a different jobname has to be chosen).

The directory `/track/mac/97/$USER/example_sixve` contains the output of a run that produced a second order one-turn-map in six phase space variables. Note that it is necessary to switch of the RF in the `fort.3` input file (`SYNC` block) if a one-turn-map in four phase space variable is to be produced.

References

[1] F. Schmidt, "SIXTRACK, version 1.1, single particle tracking code treating transverse motion with synchrotron oscillations in a symplectic manner", CERN SL/90-52 (AP) (1990).
http://hpariel.cern.ch/frs/report/six.ps.Z or http://hpariel.cern.ch/frs/six/six.html

[2] F. Schmidt, "SIXTRACK - a single particle tracking code", Proceedings of the Workshop on Nonlinear Problems in Future Particle Accelerators, Capri, Italy (1990).

[3] G. Ripken and F. Schmidt, "A symplectic six-dimensional thin-lens formalism for tracking", DESY 95-063 and CERN SL/95-12 (AP).