EUROPEAN LABORATORY FOR PARTICLE PHYSICS
CERN – SL Division

CERN/SL/94–56 (AP)
Update January 2012

SixTrack

Version 4.2.16

Single Particle Tracking Code Treating Transverse Motion with Synchrotron Oscillations in a Symplectic Manner

User's Reference Manual

F. Schmidt

Abstract

The aim of SixTrack is to track two nearby particles taking into account the full six–dimensional phase space including synchrotron oscillations in a symplectic manner. It allows to predict the long–term dynamic aperture which is defined as the border between regular and chaotic motion. This border can be found by studying the evolution of the distance in phase space of two initially nearby particles. Parameters of interest like nonlinear detuning and smear are determined via a post–processing of the tracking data. An analysis of the first order resonances can be done and correction schemes for several of those resonances can be calculated. Moreover there is the feature to calculate a one–turn map to very high order and the full six–dimensional case, using the LBL differential algebra. This map allows a subsequent theoretical analysis like normal form procedures which are provided by É. Forest [1].

The linear elements are usually treated as thick elements in SixTrack. In that case there is at least one non–zero length element in the structure file which is not a drift–element. If the accelerator, however, is modelled exclusively with drifts and kicks SixTrack automatically uses the thin–lens formalism according to G. Ripken [2]. A common header of output data and the format of these data has been found for MAD and SixTrack tracking data.

Geneva, Switzerland
April 27, 2015

Update titlepage

Contents

1 Introduction **2**

2 Versions and Service **4**

3 Input Structure **6**
3.1 General Input . . . 6
3.1.1 Program Version . . . 6
3.1.2 Print Selection . . . 6
3.1.3 Comment Line . . . 7
3.1.4 Iteration Errors . . . 7
3.1.5 MAD – SixTrack Conversion . . . 8
3.2 Machine Geometry . . . 8
3.2.1 Single Elements . . . 8
3.2.1.1 Linear Elements . . . 8
3.2.1.2 Nonlinear Elements . . . 9
3.2.1.3 Multipole Blocks . . . 10
3.2.1.4 Cavities . . . 10
3.2.1.5 Beam–Beam Separation . . . 11
3.2.1.6 "Phase–trombone" or matrix element . . . 11
3.2.1.7 AC dipole . . . 11
3.2.1.8 Dipole edge . . . 11
3.2.1.9 Solenoid . . . 12
3.2.1.10 Crab Cavity . . . 12
3.2.1.11 Beam Position Monitor . . . 12
3.2.2 Block Definitions . . . 12
3.2.3 Structure Input . . . 13
3.2.4 Displacement of Elements . . . 13
3.3 Special Elements . . . 14
3.3.1 Multipole Coefficients . . . 14
3.3.2 Aperture Limitations . . . 15
3.3.3 Power Supply Ripple . . . 15
3.3.4 Dynamic Kicks . . . 16
3.3.5 Beam–Beam Element . . . 22
3.3.6 "Phase Trombone" Element . . . 23
3.4 Organising Tasks . . . 23
3.4.1 Random Fluctuation Starting Number . . . 24
3.4.2 Organisation of Random Numbers . . . 25
3.4.3 Combination of Elements . . . 25
3.5 Processing . . . 26
3.5.1 Linear Optics Calculation . . . 26
3.5.2 Tune Variation . . . 27
3.5.3 Chromaticity Correction . . . 27
3.5.4 Orbit Correction . . . 28

3.5.5 Decoupling of Motion in the Transverse Planes . . . 29
3.5.6 Sub–resonance Calculation . . . 29
3.5.7 Search for Optimum Places to Compensate Resonances . . . 30
3.5.8 Resonance Compensation . . . 30
3.5.9 Differential Algebra . . . 31
3.5.10 Normal Forms . . . 32
3.5.11 Corrections . . . 33
3.5.12 Post–processing . . . 34
3.6 Initial Conditions for Tracking . . . 36
3.6.1 Tracking Parameters . . . 37
3.6.2 Initial Coordinates . . . 40
3.6.3 Synchrotron Oscillation . . . 41
3.7 Extra output files . . . 41
3.7.1 Dumping of beam population . . . 41

4 Acknowledgement **44**

A List of Keywords **45**

B List of Default Values **47**
B.1 Default Tracking Parameters . . . 47
B.2 Default Size Parameters . . . 48

C Input and Output Files **49**

D Data Structure of the Data–Files **52**

E Tracking Examples **57**
E.1 Input Example . . . 57
E.2 Output Example . . . 59
E.3 Plot Example . . . 63

Bibliography **70**

List of Tables

2.1 External Routines . . . 4

3.1 Iteration Errors . . . 7
3.2 Different Types of Linear Elements . . . 8
3.3 Different Types of Nonlinear Elements . . . 9
3.4 Available function types in DYNK. . . . 17
3.5 Initial Coordinates of the 2 Particles . . . 42

A.1 List of Keywords . . . 45

B.1 Default Tracking Parameters . . . 47
B.2 Default Size Parameters . . . 48

C.1 List of Input and Output Files. . . . 49

D.1 Header of the Binary Data–Files . . . 52
D.2 Format of the Binary Data . . . 53
D.3 Post–processing Data . . . 54
D.4 4D Linear Parameters . . . 55
D.5 Format of file with external errors # 16 and internal errors written to # 9 . . . 56
D.6 Format of file # 34 for detuning and distortion calculation with external program "SODD" [20] . . . 56

Chapter 1

Introduction

The Single Particle Tracking Code SixTrack is optimised to carry two particles [1] through an accelerator structure over a large number of turns. It is an offspring of RACETRACK [3] written by Albin Wrulich and its input structure has been changed as little as possible so that slightly modified RACETRACK input files or those of other offsprings like FASTRAC [4] can be read in.

The main features of SixTrack are:

1. Treatment of the full six–dimensional motion including synchrotron motion in a symplectic manner [5]. The energy can be ramped at the same time considering the relativistic change of the velocity [6].

2. Detection of the onset of chaotic motion and thereby the long–term dynamic aperture by evaluating the Lyapunov exponent.

3. Post–processing procedure allowing
   - calculation of the Lyapunov exponent
   - calculation of the average phase advance per turn
   - FFT analysis
   - resonance analysis
   - calculation of the average, maximum and minimum values of the Courant–Snyder emittance and the invariants of linearly coupled motion
   - calculation of smear
   - plotting using the CERN packages HBOOK, HPLOT and HIGZ [7, 8, 9]

4. Calculation of first–order resonances and of correction schemes for the resonances [10].

5. Calculation of the one–turn map using the differential algebra techniques. The original DA package by M.Berz [11] has been replaced by the package of LBL [1]. The Fortran code is transfered into a Map producing via the (slightly modified) "DAFOR" code [12].

6. The code is vectorised, with two particles, the number of amplitudes, the different relative momentum deviations $\frac{\Delta p}{p_o}$ in parallel [13].

7. Operational improvements:
   - free format input
   - optimisation of the calculation of multipole kicks
   - improved treatment of random errors
   - each binary data–file has a header describing the history of the run (Appendix D)

[1] Two particles are needed for the detection of chaotic behaviour.

The SixTrack input is line oriented. Each line of 80 characters is treated as one string of input in which a certain sequence of numbers and character strings is expected to be found. The numbers and character strings must be separated by at least one blank, floating point numbers can be given in any format, but must be distinguished from integer numbers. Omitted values at the end of an input line will keep their default values (B.1), and lines with a slash "/" in the first column will be ignored by the program.

For detailed questions concerning rounding errors, calculation of the Lyapunov exponent and determination of the long–term dynamic aperture, see [14].

In chapter 3, the input structure of SixTrack is discussed in detail. To facilitate the use of the program, a set of appendices are added, giving a list of keywords (Appendix A), a list of default values (Appendix B), the input and output files (Appendix C), a description of the data structure of the binary data–files (Appendix D) and tracking examples (Appendix E).

Chapter 2

Versions and Service

There are two versions: for element by element tracking there is a vector version, and there is a version to produce a one–turn map using the LBL Differential Algebra package. In both cases the input structure file # 2 is used to determine if the thick or thin linear element mode has to be used.

To use the power of the Differential Algebra, for instance to calculate the 6–D closed orbit in an elegant fashion, the tracking versions may also be equipped with a low order map facility to avoid the otherwise huge demand on memory.

It must be mentioned that in the linear thin–lens version dipoles have to be treated in a special way. See section 3.2.1.3 for details.

To convert MAD files into SixTrack input a special conversion program *mad_6t* [15] has been developed (see also 3.1.5).

The following subroutines are taken from various packages:

Table 2.1: External Routines

Package	**Routine**	**Purpose**
NAGLIB	E04UCF, E04UDM, E04UEF, X04ABF	using internally Normal Forms
HBOOK	HBOOK2, HDELET, HLIMIT, HTITLE	graphic basics
HPLOT	HPLAX, HPLCAP, HPLEND, HPLINT,	graphic options
	HPLOPT, HPLSET, HPLSIZ, HPLSOF	
HIGZ	IGMETA, ISELNT, IPM, IPL	graphic output

All versions can be downloaded from the web. Please first go to: *http://cern.ch/Frank.Schmidt/Sourc* update in case of problems contact the author via:

FRANK.SCHMIDT@CERN.CH

An updated version of this manual can be retrieved from the WWW at the location: *http://cern.ch/Frank.Schmidt/Documentation/doc.html*.

Lastly, I would like to give a short historical overview how the versions of SixTrack have evolved.

- **Version 1**

The first version has been an upgrade of RACETRACK [3] to include the full 6D formalism for long linear elements by G. Ripken [5].

- **Version 2**

  The DA–package and the Normal Form techniques [11, 16] have been added to allow the production of high–order one–turn Taylor maps and their analysis. The 6D thin–lens formalism [2] has also been included to speed–up the tracking without appreciable deterioration of the accelerator model for very large Hadron colliders like the LHC.

- **Version 3**

  For the present version the beam–beam kick à la Bassetti and Erskine [17] has been included together with the 6D part by Hirata et al. [18]. Moreover, this 6D part has been upgraded to include the full 6D linear coupling [19]. Lastly, the LBL DA–package has replaced the original one by Berz and all operations, needed to set–up the accelerator structure, are now performed with the help of Forest's LieLib package [1].

- **Version 4 – in preparation**

  Upgrading the program to FORTRAN90. This is of interest in particular as É. Forest has wrapped his tools in this more powerful language. Using operator overloading it will be possible to perform the map production with a code which is almost identical to that which does the normal tracking.

Chapter 3

Input Structure

The idea of RACETRACK input is to use a sequence of input blocks, each block with a specific keyword in the first line, the keyword "NEXT" in the last line and the input data in the lines in between. The keyword "ENDE" ends this sequence, and all blocks after this keyword are ignored. This system makes it easy to read input and allows easy change and addition of input blocks. It was therefore also used in SixTrack.

3.1 General Input

3.1.1 Program Version

Description The *Program Version* input block determines if all of the input will be in the input file # 3 or if the geometry part of the machine (see 3.2) will be in a separate file # 2. The latter option is useful if tracking parameters are changed but the geometry part of the input is left as it is. The geometry part can be produced directly from a MAD input file (see 3.1.5).

Keyword FREE or GEOM

Number of data lines 0

Format *keyword comment title*

keyword The first four characters of the first line of the input file # 3 are reserved for the keyword (FREE for free format input with all input in file # 3; GEOM if the geometry part is in file # 2)

comment Following the first four characters, 8 characters are reserved for comments

title The next 60 characters are interpreted as the title of the output file # 6

3.1.2 Print Selection

Description Use of the *Print Selection* input block causes the printing of the input data to the output file # 6. It is advisable to always use this input block to have a complete protocol of the tracking run.

Keyword PRIN

Number of data lines 0

3.1.3 Comment Line

Description An additional comment can be specified with this block. It will be written to the binary data files (Appendix D) and will appear in the post–processing output as well.

Keyword COMM

Number of data lines 1

Format A string of up to 80 characters.

3.1.4 Iteration Errors

Description For the processing procedures, the number of iterations and the precision to which the processing is to be performed are chosen with the *Iteration Errors* input block. If the input block is left out, default values will be used.

Keyword ITER

Number of data lines 1 to 4

Format Each data line holds three values as in table 3.1, except for the fourth line one which the horizontal and vertical aperture limits can be additionally specified. This has been added to avoid artificial crashes for special machines.

Table 3.1: Iteration Errors

data line	integer	double	default value	number of iterations for	demanded precision of	variations of
1	ITCO		50	closed orbit calculation		
		DMA	1e-12		closed orbit displacements	
		DMAP	1e-15		derivative of closed orbit displacements	
2	ITQV		10	Q adjustment		
		DKQ	1e-10			quadrupole strengths
		DQQ	1e-10		tunes	
3	ITCRO		10	chromaticity correction		
		DSM0	1e-10			sextupole strengths
		DECH	1e-10		chromaticity correction	
4		DE0	1e-9			momentum spread for chromaticity calculation
		DED	1e-9			momentum spread for evaluation of dispersion
		DSI	1e-9		desired orbit r.m.s. value; compensation of resonance width	
		APER(1)	1000[mm]		horizontal aperture limit	
		APER(2)	1000[mm]		vertical aperture limit	

3.1.5 MAD – SixTrack Conversion

Description A converter has been developed [15] which is directly linked to MAD8. It produces the geometry file # 2; an appendix to the parameter file # 3 which defines which of the multipole errors are switched on; the error file # 16 and the file # 8 which holds the transverse misalignments and the tilt of the nonlinear kick elements. It also produce a file (unit 34) with linear lattice functions, phase advances and multipole strengths needed for resonance calculations for the program *SODD* [20].

3.2 Machine Geometry

3.2.1 Single Elements

Description The *Single Elements* input block defines the name and type of linear and nonlinear elements, the inverse bending radius or multipole strength respectively, and the strength and length of the elements. Linear and nonlinear elements are distinguished by length; linear elements have a nonzero length and nonlinear elements have zero length. Both kinds of elements can appear in the input block in arbitrary order. The input line has a different format for linear and nonlinear elements. Moreover, the multipoles, being a set of nonlinear elements, are treated in a special way. The maximum number of elements is set as a parameter (see Appendix B.2).

Keyword SING

Number of data lines variable

Format See the following three sections.

3.2.1.1 Linear Elements

Description Each linear single element has a name, type, inverse bending radius, focusing and a nonzero length.

Format *name type* ϱ^{-1} K *length*

name May contain up to sixteen characters

type As shown in the table 3.2

ϱ^{-1} Inverse bending radius in m^{-1}

K Focusing strength in m^{-2}

length Magnet length in meters

Table 3.2: Different Types of Linear Elements

type	ϱ^{-1}	K	description
0	0	0	drift length magnet
1	X	0	horizontal (rectangular) bending
2	0	X	quadrupole (– focusing, + defocusing)
3	X	0	horizontal (sector) bending
4	X	0	vertical (rectangular) bending
5	X	0	vertical (sector) bending
6	X	X	horizontal combined function magnet
7	X	X	vertical combined function magnet
8	X	0	edge focusing

Remarks

1. For the horizontal plane the bending radius is defined to be negative ($\varrho < 0$). This is different from other programs like MAD [21].

2. $K < 0$ corresponds to a horizontal focusing quadrupole.

3. For the length of an edge focusing element (type=8) the same value must be used as for the corresponding bending magnet. A sector bending magnet is transformed into a rectangular magnet with an edge focusing element of positive length on either side, while for the opposite transformation a negative length is required.

4. It is important to note that the splitting of a rectangular magnet, which is sometimes necessary if multipole errors are to be introduced, does change the linear optics. It is therefore advisable to replace the rectangular magnet with a sector magnet, which can be split without affecting the linear optics, and make an overall transformation into a rectangular magnet via edge focusing elements. Do not forget to use the total length of dipole as the length of the edge focusing element.

3.2.1.2 Nonlinear Elements

Format *name type* K_n*–strength r.m.s.–strength length*

name May contain up to sixteen characters

type As shown in table 3.3

K_n**–strength** Average multipole strength

r.m.s.–strength Random multipole strength

length Must be $= 0$

Table 3.3: Different Types of Nonlinear Elements

type	strength	description
0	–	observation point (for instance for aperture limitations)
1 –1	$b_1[\mathrm{rad} \cdot \mathrm{m}^0]$ a_1	horizontal bending kick vertical bending kick
2 –2	$b_2[\mathrm{rad} \cdot \mathrm{m}^{-1}]$ a_2	normal quadrupole kick skew quadrupole kick
$\vdots$		
10 –10	$b_{10}[\mathrm{rad} \cdot \mathrm{m}^{-9}]$ a_{10}	normal 20^{th} pole skew 20^{th} pole

Remarks

1. Because the horizontal bending magnet is defined to have a negative bending radius, the sign for normal elements is different from other programs like MAD, while skew elements have the same sign.

2. Again contrary to other programs the factor $(n-1)!$ is already included in the multipole strength, which is defined as follows:

   - for normal elements $b_n(\mathrm{SixTrack}) = \frac{-1}{(n-1)!} L_{element} b_n(\mathrm{MAD})$

- for skew elements $a_n(\text{SixTrack}) = \frac{1}{(n-1)!} L_{element} a_n(\text{MAD})$

3. Unlike in RACETRACK, the horizontal and vertical displacements do not fit into the 80 character input lines of SixTrack. They have to be introduced in a separate *Displacements of Elements* input block (see 3.2.4).

3.2.1.3 Multipole Blocks

Description A set of normal, normal–r.m.s., skew and skew–r.m.s. errors can be combined effectively. The actual values for the strengths have to be given in a separate *Multipole Coefficient* input block (see 3.3.1) which must have the same name. To consider the curvature of dipoles which are replaced by drifts and dipole kicks this block is used in two different ways.

Format *name type cstr cref length*

- Marker for high order kick (default)

  name May contain up to sixteen characters

  type Must be = 11

  cstr The bending strength given in the *Multipole Coefficient* input block (3.3.1) is multiplied with this factor.

  cref The reference radius given in the *Multipole Coefficient* input block (3.3.1) will be multiplied by this factor. If it is zero the multipole block will be ignored.

  length Must be = 0

- Default + dipole curvature

  name May contain up to sixteen characters

  type Must be = 11

  cstr The bending strength [rad] of horizontal or vertical dipole.
  Internally the value is set to one to allow the processing of a multipole block (3.3.1).

  cref The length [m] of the dipole that is approximated by a kick. Internally this value is set to one to allow the processing of a multipole block (3.3.1).

  length

  - *length = -1* : horizontal dipole
  - *length = -2* : vertical dipole

Remark The definition of the multipole strength in a block will be given in (3.3.1).

3.2.1.4 Cavities

Format *name type u0 harm lag*

name May contain up to sixteen characters

type Type identifier is +12 and −12 for above and below transition energy respectively.

u0 Circumference voltage in [MV]

harm Harmonic number

lag Lag angle in the cavity (zero is default)

3.2.1.5 Beam–Beam Separation

Format *name type h-sep v-sep strength-ratio* σ_hor^2 σ_ver^2 σ_lon^2

name May contain up to sixteen characters

type 20

h-sep Horizontal beam–beam separation [mm]

v-sep Vertical beam–beam separation [mm]

strength-ratio Strength ratio with respect to the nominal beam–beam kick strength. This is useful, in particular for 4D, to allow for splitting one beam–beam kick into several (longitudinally close by) kicks.

σ_hor^2 when the flag $lhc = 2$ is set in the BEAM block of the fort.3 file, this column represent the horizontal σ for the strong beam [mm^2]

σ_ver^2 when the flag $lhc = 2$ is set in the BEAM block of the fort.3 file, this column represent the vertical σ for the strong beam [mm^2]

σ_lon^2 this variable is for future purposes, at the present it is always equal to zero.

Remark These beam–beam elements become active when the "Beam–Beam" input block 3.3.5 is used.

3.2.1.6 "Phase–trombone" or matrix element

Format *name type*

name May contain up to sixteen characters

type 22

Remark These "trombone" elements become active when the "Phase Trombone Element" input block 3.3.6 is used.

3.2.1.7 AC dipole

Format *name type ACdipAmp Qd ACdipPhase*

name May contain up to sixteen characters

type Type identifier is +16 and −16 for horizontal and vertical AC dipoles respectively.

ACdipAmp Maximum excitation amplitude [Tm].

Qd Excitation frequency in units of $[2 \times \pi]$.

ACdipPhase Phase of the harmonic excitation in radians.

Remark The length of the ramps and the flat top are specified in the "Displacement" block 3.2.4. The energy introduced in the "Initial coordinates" block 3.6.2 is used to compute the deflection angle.

3.2.1.8 Dipole edge

Missing!

dipedge
type 24

3.2.1.9 Solenoid

Missing!

type 25

3.2.1.10 Crab Cavity

Multipole RF kicks – ±26, ±27, ±28

Format *name type Voltage Frequency Phase*

name May contain up to sixteen characters

type Type identifier is +23 and −23 for horizontal and vertical crab cavities respectively.

Voltage Crab Cavity voltage [MV].

Frequency Crab Cavity frequency [MHz].

Phase Phase of the excitation in radians.

3.2.1.11 Beam Position Monitor

Format *BPMname 0 0 0 0*

BPMname Must start with "BP" and maybe followed by forteen characters.

Remark This element dumps the coordinates of the 1st particle to the file with name *BPMname*. The file contains 7 columns: x,x', y,y', ct,$\delta p/p$ and E. Usual SixTrack units are used. Any number of BPM elements can be used but the names must differ.

3.2.2 Block Definitions

Description In four–dimensional transverse tracking, the linear elements between nonlinear elements can be combined to a single linear block to save computing time.

Keyword BLOC

Number of data lines variable but at least one

Format

- first data line: *mper msym(1) . . . msym(mper)* (integers)
- from second data line on: *block–name* {*element–name*}

mper Number of super–periods. The following set of blocks is considered a *super–period*. The accelerator consists of *mper* super–periods.

msym(i) ± 1 for each super–period. If *msym(i)*=1, the i'th super–period will be built up in the order in which linear elements appear in the blocks below. If *msym(i)*=–1, the super–period will be built up in reverse order.

block–name The name of the block with up to sixteen characters

element–name The element names have to appear as a linear element in the list of "single elements" (3.2.1.1). If one line is too short to contain all the elements of a block, a line with additional elements to the same block can be added. At least 5 (five) blanks must appear at the beginning of the extra line so that names of blocks and names of linear elements in a block can be distinguished.

Remarks

1. When synchrotron oscillation is introduced, the linear elements can no longer be lumped into one block, because in that case even a drift length magnet is a nonlinear element with respect to the longitudinal plane. However, the block structure is still kept to make use of the speed–up in case one can restrict the studies to the four–dimensional case.
2. The maximum number of blocks and the maximum number of entries in each block are defined as parameters (Appendix B.2).
3. The inversion of a super–period (*msym(i)*= –1) is presently no longer allowed.

3.2.3 Structure Input

Description The model of the accelerator is put together by constructing a sequence of blocks of linear elements, nonlinear elements, observation points, and possibly a cavity with the keyword "CAV" used if this name does not appear in the list of single elements (3.2.1) with type ±12. In that case, its parameters are given in the *Synchrotron Oscillations* input block (3.6.3).

Format { *structure–element* | CAV | GO }

structure–element Structure elements must appear as nonlinear and observation elements in the single element list or in the list of blocks of the *Block Definition* input block (3.2.2).

CAV A cavity can be introduced by a keyword "CAV". This element does not appear in the single element list (3.2.1).

GO Starting point: the keyword "GO" denotes where the tracking is started and where the tracked coordinates are recorded at each turn.

Remark Repetition of parts of the structure is indicated by parentheses with a multiplying factor N in front of them. If the left parenthesis "(" occurs in a line of input, the factor N is expected to be found in the preceding characters. If the characters are blank, N is set to 1. The right parenthesis ")" signals the end of the sequence to be repeated.

3.2.4 Displacement of Elements

Description This block allows to displace nonlinear elements in horizontal and vertical positions. With the r.m.s. values of the horizontal and vertical displacements it is possible to achieve a displacement that is different from element to element.

To simulate a measured closed orbit at the position of nonlinear elements, it is convenient to use the *Displacement of Elements* input block instead of trying to produce a closed orbit by dipole kicks.

Keyword DISP

Number of data lines variable

Format *name xd xdrms yd ydrms*

name Name of the element which is displaced

xd Horizontal displacement [mm]

xdrms R.m.s. of horizontal displacement [mm]

yd Vertical displacement [mm]

ydrms R.m.s. of vertical displacement [mm]

In the case of an AC dipole these variables are not meant for displacing this element but are used for the following AC dipole parameters:

Format *name nfree nramp1 nplato nramp2*

name May contain up to sixteen characters

nfree Number of turns free of excitation at the begining of the run.

nramp1 Number of turns to ramp up the excitation amplitude from 0 to *ACdipAmp*.

nplato Number of turns of constant excitation amplitude.

nramp2 Number of turns to ramp down the excitation amplitude.

Remark In RACETRACK the displacements had been included in the *Single Element* input block (3.2.1). In SixTrack they must be given in the separate *Displacement of Elements* input block because of the limited length of one line of input.

3.3 Special Elements

One advantage of SixTrack, that has been adopted from RACETRACK, is that it easily allows to define elements for a specific purpose. The special elements implemented till now are found in this section.

3.3.1 Multipole Coefficients

Description Sets of normal and skew multipoles of up to tenth order, each with an r.m.s. value, can be combined with this block. The multipole kick is calculated using a Horner scheme which saves considerably in computation time. Moreover, using the multipole block reduces the number of elements in the single element list (3.2.1).

Keyword MULT

Number of data lines 2 to 12

Format

- first data line: *name* R_0 δ_0
- data lines 2 to 12: B_n *r.m.s.–*B_n A_n *r.m.s.–*A_n

name Name of the multipole block which must appear in the list of single elements (3.2.1.3).

R_0 Reference radius (in mm) at which the magnet errors are calculated. This makes it convenient to use values from field measurements.

δ_0 Bending strength of the dipole (in mrad). Field errors of line 2–11 are taken to be relative to the bending strength.

Remarks

1. The B_n and A_n are related to the b_n, a_n of the single nonlinear element (3.2.1.2) in the following way:
$$b_n = \delta_0 B_n R_0^{1-n} 10^{3n-6}; a_n = \delta_0 A_n R_0^{1-n} 10^{3n-6}$$
2. The sign convention and the factorial $(n!)$ are treated as for the single nonlinear elements in (3.2.1.2).
3. Multipoles of different names can be set to be equal using the “ORG” input block.
4. 22–poles are included (n = 11). By enlarging the parameter “MMUL”(Appendix B.2) up to 40–poles (MMUL=20) can be treated. To make the change of MMUL effective, it is of course necessary to recompile the program.

3.3.2 Aperture Limitations

Description This input data block is used to introduce additional collimators or aperture limitations in the machine. Each nonlinear element can be used for this purpose. Rectangular or elliptical shapes of the aperture limitations are allowed. On top of that there is a general (rectangular) aperture check at each non–zero length element. The general aperture values are chosen to be large enough (B.1) to define the short–term dynamic aperture.

Keyword LIMI

Number of data lines variable

Format *name type–of–limitation xaper yaper*

name The name of any nonlinear (zero length) element in the *Single Element* input block (3.2.1.2) except multipole blocks (3.2.1.3).

type–of–limitation Two types of aperture limitations are allowed:
“RE” for a rectangular aperture shape, i.e.

$$x_i < \text{xaper}, y_i < \text{yaper}$$

“EL” for an elliptical aperture shape, i.e.

$$\frac{x_i^2}{\text{xaper}^2} + \frac{y_i^2}{\text{yaper}^2} < 1$$

xaper Aperture in the horizontal plane in mm

yaper Aperture in the vertical plane in mm

3.3.3 Power Supply Ripple

The RIPP block is been deprecated since release 4.5.20, and the functionality is now provided by the DYNK block (3.3.4). A fort.3 file containing a RIPP block is therefore no longer valid, and will result in an error message. The description below is therefore only provided as a reference for those who need to convert old input files.

Description If power supply ripple is to be considered this input data block can be used. A nonlinear quadrupole is expected as a ripple element (type=2 and zero length in the single element list (3.2.1.2)), but in principle other nonlinear elements are also allowed. Ripple depth, ripple frequency and starting phase of the ripple frequency are the input parameters.

Keyword RIPP

Number of data lines variable

Format *name ripple–depth ripple–frequency start–phase nrturn*

name Name of the nonlinear element in the “single element” block (3.2.1.2)

ripple–depth Maximum kick strength of the ripple element, a quadrupole kick is usually expected

ripple–frequency Given in number of turns (a real value is allowed) of one ripple period

start–phase Initial phase of the ripple element

nrturn Initial number of turns, for prolongation runs the number of turn already done

3.3.4 Dynamic Kicks

Description The DYNamic Kicks module [27] allows time-dependent modification of the settings of single elements. Currently supported are thin “standard” elements (type ±1–±10 3.2.1.2) and crab cavities/RF multipoles (type ±23, ±26, ±27 and ±28 3.2.1.10). Different element types support different attributes, where the standard elements support setting the multipole strength K_n while crab cavities support setting of the voltage, frequency, and phase relative to the bunch. The settings can be computed on-the fly using several functions, loaded from input files, or a combination.

Further, unless explicitly switched off using a NOFILE statement, DYNK produces an output file “dynksets.dat”. This file contains the setting of all elements and attributes for which DYNK is active. It is written in all turns of the simulation, even if DYNK is not active in that exact turn.

Keyword DYNK

Number of data lines variable

Format There are four types of statements possible in a DYNK block, listed below. On top of this, lines starting with “/” are treated as a comment and ignored.

FUN *FUN function-name function-type arg1 arg2 arg3 . . .*
This statement defines a function, i.e. something which when evaluated produces a numerical value which can be used to set the value of an element attribute. The functions in DYNK all have a unique name, and they may take up to 7 arguments (a limitation imposed by the internal parameter *getfields_n_max_fields*). The function type must be one of those listed in Table 3.4. A function may be defined so that it uses the result of another function, which must be defined above it in the DYNK block. This requirement avoids any possibility for infinite recursion. The functions are only evaluated when needed, i.e. when used by a SET statement in that turn. The functions may thus be evaluated multiple times in one turn (if used by multiple SET statements which are active in that turn, or referenced by multiple other FUN statements which are themselves used more than once in that turn), or it may not be evaluated at all. The functions are always evaluated as a function of the current turn number t, which may be shifted by a turn-shift specified in a SET statement.

Table 3.4: Available function types in DYNK.

Type name	Arguments	Description
"System" functions		
GET	*element-name[string]* *function-name[string]*	Extracts the original value of a setting, i.e. as specified in the SINGLE ELEMENT section.
FILE	*filename[string]*	Loads the settings from file; the file is expected to be an ascii file with two columns where the first column is the turn number (should start at 1 and include all turns up to as long as is wanted), and the second column is the value for that turn number.
FILELIN	*filename[string]*	Similar to FILE, but any double can be used as the turn number as long as they are monotonically rising. When evaluated, the function interpolates from the line-segments specified in the file.
RANDG	*seed1[int]* *seed2[int]* *mu[real]* *sigma[real]* *mcut[int]*	Returns a pseudorandom number generated from a Gaussian distribution. The mean value and width is controlled by *mu* and *sigma*, while *mcut* is the maximum number of sigmas to generate numbersup to, set to 0 to disable this cut. The integers *seed1* and *seed2* are the seed used to initialize the RANECU generator. Note that every RANDG function defined in DYNK uses its own separate random number stream.
2-operand operators		
ADD	*function-name-1[string]* *function-name-2[string]*	Evaluate the functions referenced by *function-name-1* and *function-name-2*, and return the sum of the results.
SUB	*function-name-1[string]* *function-name-2[string]*	Similar to ADD, but return the result of function1 minus function2.
MUL	*function-name-1[string]* *function-name-2[string]*	Similar to ADD, but return the product of the results.
DIV	*function-name-1[string]* *function-name-2[string]*	Similar to ADD, but return the result of function1 divided by function2
POW	*function-name-1[string]* *function-name-2[string]*	Similar to ADD, but return the result of function1 raised to the power of function2.
1-operand operators		
MINUS	*function-name*	Returns the value of the named function, with the oposite sign.
SQRT	*function-name*	Returns the square root of the value generated by the named function.
SIN	*function-name*	Returns the sine of the value generated by the named function.
COS	*function-name*	Returns the cosine of the value generated by the named function.
LOG	*function-name*	Returns the natural logarithm of the value generated by the named function.
LOG10	*function-name*	Returns the common logarithm of the value generated by the named function.
EXP	*function-name*	Returns the natural exponential function e^x, where x is the value generated by the named function.
(The table continues on the next page)		

Type name	Arguments	Description
Polynomial and elliptical functions		
CONST	*value[real]*	Always returns the value specified.
TURN	(none)	Return the turn number, i.e. $y(t) = t$.
LIN	*a[real] b[real]*	Computed value from the linear function $y(t) = a \cdot t + b$.
LINSEG	*x1[real] x2[real] y1[real] y2[real]*	The function is defined by a line segment between the points (x_1, y_1) and (x_2, y_2), and undefined for $x < x_1$ and $x > x_2$. It is required that $x_1 < x_2$.
QUAD	*a[real] b[real] c[real]*	Computed value from the quadratic function $y(t) = a \cdot t^2 + b \cdot t + c$.
QUADSEG	*x1[real] x2[real] y1[real] y2[real] deriv1[real]*	The quadratic function is defined by overlapping the quadratic curve segment which passes through the points (x_1, y_1) and (x_2, y_2), and $\mathrm{d}y/\mathrm{d}x$ at x_1 is *deriv1*. The quadratic coefficients a, b, c are calculated as $a = \frac{deriv1}{x_1 - x_2} + \frac{y_2 - y_1}{(x_1 - x_2)^2}$, $b = \frac{y_2 - y_1}{x_2 - x_1} - (x_1 + x_2) \cdot a$ and $c = y_1 + (-x_1^2 \cdot a - x_1 \cdot b)$.
Trancendental functions		
SINF	*A[real] omega[real] phi[real]*	Computes $y(t) = A \sin(\omega t + \phi)$.
COSF	*A[real] omega[real] phi[real]*	Computes $y(t) = A \cos(\omega t + \phi)$.
COSF_RIPP	*A[real] period[real] phi[real]*	Computes $y(t) = A \cos\left(\frac{2\pi(t-1)}{\text{period}} + \phi\right)$. This specialized cosine is provided for compatibility, to be used when replacing old *RIPP* blocks.
Specialized functions		
PELP	*tinj[real] Iinj[real] Inom[real] A[real] D[real] R[real] te[real]*	This function describes a patched "Parabolic-Exponential-Linear-Parabolic" function, as used for ramping the LHC dipoles and described in [28, Appendix C] and [29]. The parameters are: • The injection time *tinj*, which is the time (in turn numbers) when the ramp starts. • The injection value *Iinj*, which is the value when $t \leq t_{inj}$ • The final value *Inom*, which is the value after the end of the ramp. • The acceleration parameter A, which describes how fast the current is growing in the first (parabolic) segment. • The decelertation parameter D, which describes how fast the current growths flattens out in the forth (parabolic) segment. • The ramp rate R, which describes the maximum ramp rate, seen in the third (linear) segment. • The start time of the ramp *te*, which describes at what time it switches from the parabolic (first) to the exponential (second) segment.

SET *SET element-name attribute-name function-name first-turn last-turn turn-shift*
This statement defines an element setpoint, which changes an element/attribute to the value computed by the given function. The *SET* becomes active when the turn number reaches *first-turn*, and switches off once *last-turn* has been passed. When switched off, the value applied in `last-turn` stays for the rest of the simulation, or until overwritten by another *SET*. If *last-turn* equals -1, the *SET* is active untill the end of the simulation.

The argument *turn-shift* is an integer (positive, negative, or zero) number which is added to the current turn number before computing the function. Thus, in order to (as an example) apply an exponential decay from the value v_0 starting in turn t_0 using the function defined as $f(t) = V_0 \exp(-t/\tau)$, a *turn-shift* $-t_0$ should be applied.

NOFILE The presence of this statement in a DYNK block switches off the normal writing of the output file "dynksets.dat" in every line, instead producing a file only containing the message "### DYNK file output was disabled with flag NOFILE in fort.3 ###". This can be useful to save disk space in very long simulations.

DEBU This statement switches on extra "debugging" output from DYNK. This can be useful if debugging the code or if debugging the input.

Output file *dynksets.dat* When a DYNK block is present in the input file, a file "dynksets.dat" is created and in the current working directory. Unless a *NOFILE* statement is present, this file contains first a header "# turn element attribute SETidx funname value", followed by rows of data in the format specified in the header. This data is written for all element/attribute combinations and in all turns, wether a SET is active for this element/attribute in this turn or not. If no *SET* is active when the line is written out, the *SETidx* is written as -1, and the *funname* is "N/A". If a SET is active when the line is written out, the *SETidx* is the index of the currently active *SET* statement, where the first statement occuring in fort.3 has index 1 etc. Similarly, the *funname* is the name referencing the currently active *FUN* statement.

Examples

Replacement of RIPP block One use of the DYNK block is to replace the functionality of the RIPP block (Section 3.3.3). The FUN type COSF_RIPP is provided for exactly this purpose, and provides an exact replacement. As an example, the RIPP block in the SixTest test-case prob1 looks like (slightly reduced in size):

```
RIPPLE OF POWER SUPPLIES------------------------------------------------
         dmqx1f50l5+2      3.2315D-10    224.9
         dmqx2af50l5+2    -3.2315D-10    224.9
         dmqx1f10mel5+2    2.5246D-16    0.0011245
NEXT
```

This can be replaced by the following:

```
DYNK
NOFILE
FUN RIPP-dmqx1f50l5+2 COSF_RIPP 3.2315D-10 224.9 0.0
SET dmqx1f50l5+2 average_ms RIPP-dmqx1f50l5+2 1 -1 0
FUN RIPP-dmqx2af50l5+2 COSF_RIPP -3.2315D-10 224.9 0.0
SET dmqx2af50l5+2 average_ms RIPP-dmqx2af50l5+2 1 -1 0
FUN RIPP-dmqx1f20kl5+2 COSF_RIPP 2.5246D-12 0.56225 0.0
SET dmqx1f20kl5+2 average_ms RIPP-dmqx1f20kl5+2 1 -1 0
NEXT
```

Here, each RIPP data line is replaced with two lines, one FUN statement for generating the function, and one SET statement for applying the value. Note that the SET statements have an end-time "-1", meaning it is used untill the end of the simulation. Also note the precense of the NOFILE flag, which is used to not generate a potentially very large (for very long simulations) dynkfile.dat output file.

Starting tracking inside a bump This example was taken from the paper [27], and demonstrates how a bump can be temporarilly disabled if the starting point of the tracking is inside of it. The reason for doing this is removing the neccessity of generating a starting distribution with the bump already applied. Here, the HL-LHC v1.1 lattice is used, with vertical crab cavities around the first interaction point (IP1, ATLAS), which is also the point where the tracking is started. The crab cavities opening the bump are called CRAB_IP1_L1···4, while the closing cavities are CRAB_IP1_R1···4. The DYNK block for this looks like:

```
DYNK
FUN zero CONST 0.0
FUN CV_1R1 Get CRAB_IP1_R1 voltage
FUN CV_1R2 GET CRAB_IP1_R2 voltage
FUN CV_1R3 GET CRAB_IP1_R3 voltage
FUN CV_1R4 GET CRAB_IP1_R4 voltage
SET CRAB_IP1_R1 voltage zero 1 1 0
SET CRAB_IP1_R2 voltage zero 1 1 0
SET CRAB_IP1_R3 voltage zero 1 1 0
SET CRAB_IP1_R4 voltage zero 1 1 0
SET CRAB_IP1_R1 voltage CV_1R1 2 2 0
SET CRAB_IP1_R2 voltage CV_1R2 2 2 0
SET CRAB_IP1_R3 voltage CV_1R3 2 2 0
SET CRAB_IP1_R4 voltage CV_1R4 2 2 0
NEXT
```

Here, the function "zero" is defined such that it always returns 0.0, and is used to switch off the closing cavities in the first turn, i.e. when the beam exits the bump. Further, the functions CV_1R1···1R4 and CV_1L are used to store the original value of the voltages, without having to explicitly enter them into the DYNK block.

The SET statements then first sets the voltage of all the cavities to zero in turn 1, and then in turn 2 sets it to their respective "switched on" voltages. The SET statements end after turn 2, but the last values are retained.

This means that when the simulation starts with the bunch in IP1, it exits the bump without any kicks from the closing crab cavities. It then comes around (still in turn 1), and encountered the switched-on opening cavities CRAB_IP1_L1···4, which crabs the beam. After passing through IP1, the turn counter is increased from 1 to 2, triggering the SET statements to switch on the closing cavities CRAB_IP1_R1···4 as well.

Ramp and exponential decay of crab voltage, combined with a linear drift of crab phase This slightly more complicated example builds on the example given above. It shows how to change two parameters (voltage and phase) of several objects. It also demonstrates how functions can be chained together, making more complicated functions. Some of the resulting functions are shown in Figure 3.1, and the DYNK block here looks like:

```
DYNK
/DEBUG
FUN zero CONST 0.0
FUN CV_R1 GET CRAB_IP1_R1 voltage
FUN CV_R2 GET CRAB_IP1_R2 voltage
FUN CV_R3 GET CRAB_IP1_R3 voltage
FUN CV_R4 GET CRAB_IP1_R4 voltage
```

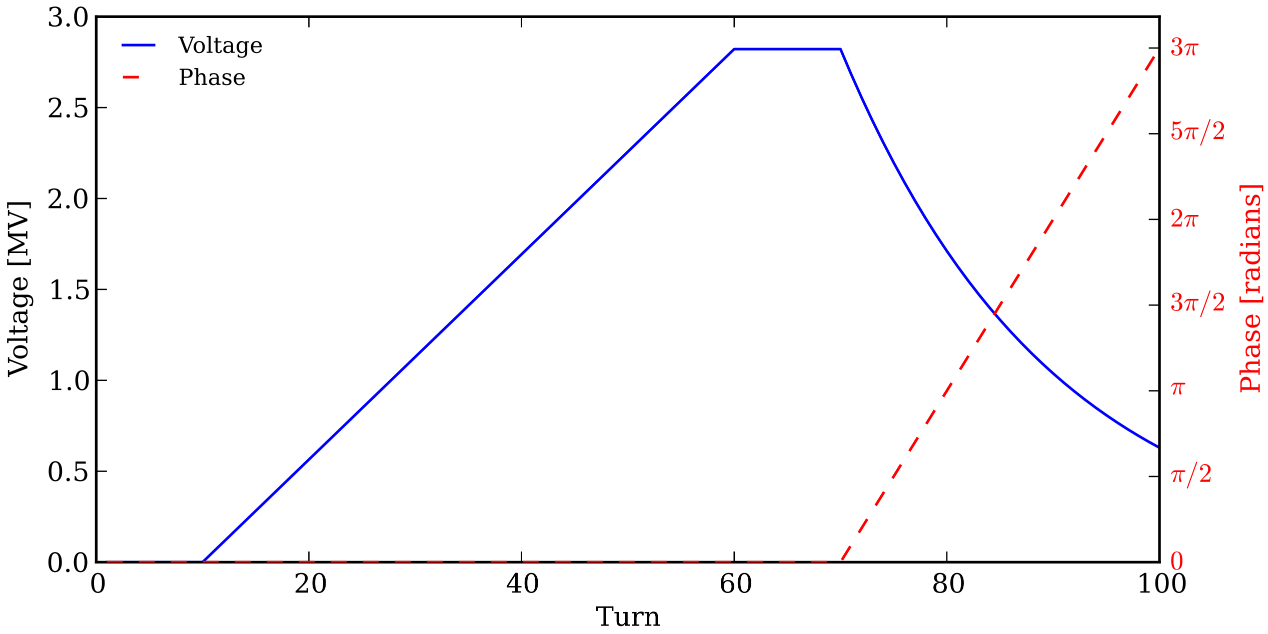

Figure 3.1: Singals generate by DYNK example for ramp + exponential decay of crab voltage, and also linear drift of crab phase. Only the signals for CRAB_IP1_L1 are shown. The plot is made from the data in dynksets.dat.

```
FUN CV_L  GET CRAB_IP1_L1 voltage
FUN ramp LIN 0.02 0
FUN ramp_R1 MUL CV_R1 ramp
FUN ramp_R2 MUL CV_R2 ramp
FUN ramp_R3 MUL CV_R3 ramp
FUN ramp_R4 MUL CV_R4 ramp
FUN ramp_L  MUL CV_L  ramp
SET CRAB_IP1_R1 voltage zero 1 10 0
SET CRAB_IP1_R2 voltage zero 1 10 0
SET CRAB_IP1_R3 voltage zero 1 10 0
SET CRAB_IP1_R4 voltage zero 1 10 0
SET CRAB_IP1_L1 voltage zero 1 9 0
SET CRAB_IP1_L2 voltage zero 1 9 0
SET CRAB_IP1_L3 voltage zero 1 9 0
SET CRAB_IP1_L4 voltage zero 1 9 0
/
SET CRAB_IP1_R1 voltage ramp_R1 11 61 -11
SET CRAB_IP1_R2 voltage ramp_R2 11 61 -11
SET CRAB_IP1_R3 voltage ramp_R3 11 61 -11
SET CRAB_IP1_R4 voltage ramp_R4 11 61 -11
SET CRAB_IP1_L1 voltage ramp_L 10 60 -10
SET CRAB_IP1_L2 voltage ramp_L 10 60 -10
SET CRAB_IP1_L3 voltage ramp_L 10 60 -10
SET CRAB_IP1_L4 voltage ramp_L 10 60 -10
/
/Voltage decay and detuning
FUN expCore LIN -0.05 0.0
FUN decay EXP expCore
FUN decayScaled MUL decay CV_L
SET CRAB_IP1_L1 voltage decayScaled 70 100 -70
```

```
SET CRAB_IP1_L2 voltage decayScaled 70 100 -70
SET CRAB_IP1_L3 voltage decayScaled 70 100 -70
SET CRAB_IP1_L4 voltage decayScaled 70 100 -70
FUN phasedrift LIN 0.3141592654 0.0
SET CRAB_IP1_L1 phase phasedrift 70 100 -70
SET CRAB_IP1_L2 phase phasedrift 70 100 -70
SET CRAB_IP1_L3 phase phasedrift 70 100 -70
SET CRAB_IP1_L4 phase phasedrift 70 100 -70
NEXT
```

The first functions defined here are the same as above, storing the default values (as defined in the single element list) for the relevant elements and also zero. Then follows a normalized linear ramp function "ramp", with gradient 0.02 = 1/50. This is then used by the "specialized" ramp functions "ramp_R1· · ·R4", which scales "ramp" so that the end point is the standard voltages for $t \in 0 \dots 50$.

These functions are used to first set the crabs to 0.0 for the first 9 revolutions, and in the 10th revolution the ramp starts. As the "ramp" function is defined starting at turn 0, a shift -10 or -11 is applied to the ramps. The ramp is switched off after turn 60/61, leaving the crabs to be operating at the last SET value.

Further, we want to demonstrate a failure in the crab voltage. This is done using an exponential decaying function $V(t) = V_0 \exp(-0.05t)$, which is implemented as three chained functions:

expCore: $f(t) = -0.05t + 0.0$

decay: $g(t) = \exp(f(t)) = \exp(-0.05t + 0.0)$

decayScaled: $h(t) = V_0 \cdot g(f(t)) = V_0 \cdot \exp(f(t)) = \exp(-0.05t + 0.0)$

For the SET, the time t is then shifted by -70 turns, so that the functions are evaluated starting at t=0.

3.3.5 Beam–Beam Element

Description The beam–beam kick, including a separation of the beams, is treated à la Basetti and Erskine [17] and implemented as in MAD [21]. However, a much faster but nevertheless precise calculation using interpolation can be used [22]. For SixTrack version 3 the beam–beam is also available in the 6D form à la Hirata [18]. Lastly, the linear coupling has been considered in 4 and 6 dimensional phase space [19].

Keyword BEAM

Number of data lines variable but at least one

Format

- first data line: *partnum emitnx emitny sigz sige ibeco ibtyp lhc ibbc*
- other data lines: *name ibsix xang xplane*

partnum (float) Number of particles in bunch

emitnx,emitny (floats) Horizontal and vertical normalized emittance respectively $[\mu m \cdot rad]$

sigz,sige (floats) R.m.s. bunch length [m] and r.m.s. energy spread

ibeco (integer) Switch (0 = off; 1 = on) to subtract the closed orbit introduced by the separation of the beams. It is recommended to always subtract it as it is not yet calculated in a selfconsistent manner.

ibtyp (integer) Switch (0 = off; 1 = on) to use the fast beam–beam algorithms developped in collaboration with G.A. Erskine and E. McIntosh. The linear optics are calculated with “exact” beam–beam kicks.

lhc For the LHC with its anti–symmetric IR the separation of the beams in one plane can be calculated by the β–function of the other plane. For flat beams (not anti-symmetric optics) the separation can be loaded from the fort.2 file. (0 = off; 1 = anti-symmetric; 2 = load from file).

ibbc Linear coupling considered in 4D and 6D (0 = off; 1 = on).

name Name of 6D beam–beam element. Beam–beam elements that do not appear will be treated as 4D kicks.

ibsix (integer) Number of slices of the 6D beam–beam kick. If *ibsix* is set to 0 this element is treated as a 4D element.

xang (float) Half crossing angle at this particular element [rad].

xplane (float) Crossing plane angle [rad].

Remark These beam–beam elements have beam–beam have to appear in the single element list (3.2.1.2) (type 20) together with their horizontal and vertical beam–beam separations (see 3.2.1.5).

3.3.6 “Phase Trombone” Element

Description The linear “phase trombone” allows to introduce a change in the tranverse phases without spoiling the linear optics of the rest of the machine, i.e. the Twiss parameters are the same at entrance and exit of the element.

Keyword TROM

Number of data lines 1 line with name and then in blocks of 14 lines with 3 entries each

Format

- first data line: *name*
- second data line: cx, cx', cy
- third data line: cy', cz, cz'
- fourth till 15^{th} $M(6 \times 6)$ matrix

name May contain up to sixteen characters

cx, cx'**, cy,** cy'**, cz,** cz' (floats) 6D closed orbit to be added to the coordinates.

M(6×6) (floats) 6×6 matrix elements

Remark The user has to make sure that the above stated conditions are fulfilled. When using the *mad_6t* [15] converter from MAD8 to SixTrack this is guaranteed to be the case. Note also that the crossterms between the transverse plains are not considered for the time being.

3.4 Organising Tasks

In this section the input data blocks are described, which are used to organise the input structure.

3.4.1 Random Fluctuation Starting Number

Description If besides mean values for the multipole errors (Gaussian) random errors should be considered this input data structure is used to set the start value for the random generator.

Keyword FLUC

Number of data lines 1

Format *izu0 mmac mout mcut* (integers)

izu0 Start value for the random number generator

mmac – *Sorry: disabled for the time being, i.e. mmac is fixed to be 1* – (In the vectorised version the number of different starting seeds can be varied. Each seed is calculated as $k \times izu0$ where k runs from 1 to *mmac* which can not exceed 5 to save storage space (see list of parameters in Appendix B.2).)

mout A binary switch for various purposes, so all options, as described below, can be combined.

- $mout = 0$: multipole errors internally created
- $mout = 1$: multipole errors read–in from external file
  External multipole errors are read–in from file 16 into the array of random values. To activate these values one has to set to a value of 1 the relevant r.m.s.–positions of the corresponding multipole blocks (3.3.1). The systematic components are added as usual and multipoles not found in the fort.16 are treated as for ($mout = 0$). An error is only detected if there are too few sets of multipoles in fort.16.
- $mout = 2$: the geometry and strength file is written to file # 4 in the same format as the input file # 2; the multipole coefficients are written to file # 9; name, misalignments and tilt is written to file # 27 and finally name, random single multipole strength and both random transverse misalignments are written to file # 31.
- $mout = 4$: Name, horizontal and vertical misalignment and also the element tilt are read–in from file # 8.
- $mout = 8$: Name and 3 Random numbers for single kick strength and both random transverse misalignments and also the value of the tilt are read–in from file # 30.

mcut The random distribution can be cut by *mcut* sigma of the distribution. No cuts are applied for *mcut = 0*.

Remarks

1. The RANECU random generator [23] is used as it produces machine independent sequences of random numbers.

2. If the starting point has to be changed or another nonlinear element is to be inserted, this can be done without changing the once chosen random distribution of errors by using the *Organisation of Random Numbers* input block.

3. The description of an accelerator is fully contained in 4 files: fort.2 (geometry), fort.3 (tracking parameters and definition of multipole blocks), fort.16 (multipole errors) and fort.30 (random numbers of the single multipole kick, the horizontal and vertical misalignment and the value of the tilt). This block allows to write out the files # 4, 9, 27, 31 which may serve as the input files # 2, 16, 8 and 30 respectively. The file fort.30 superseeds fort.8 if both files are read in.

3.4.2 Organisation of Random Numbers

Description Working on a lattice for an accelerator often requires to introduce new nonlinear elements. In those cases simply introducing this new element means that the previously chosen random distribution of the errors will be changed and with it often the linear parameters. This input data block is mainly used to avoid this problem by reserving extra random numbers for the new elements. It also allows to change the observation point without affecting the machine. The random values of different nonlinear elements including blocks of multipoles can be set to be equal to allow to vary the number of nonlinear kicks in one magnet which clearly should have the same random distribution for each multipolar kick. Finally multipole sets with different name can be made equal with this input data block.

Keyword ORGA

Number of data lines variable

Format *ele1 ele2 ele3* The data lines can be set in three different ways:

1. Ele1 = "name" where name $\neq$ MULT
   Ele2 = ignored
   Ele3 = ignored
   The nonlinear element or multipole set will have its own set of random numbers.

2. Ele1 = "name1" where name1 $\neq$ MULT
   Ele2 = "name2"
   Ele3 = ignored
   The nonlinear element or multipole block Ele1 has the same random number set as those of Ele2, if it follows Ele2 as the first nonlinear element in the structure list (3.2.3).

3. Ele1 = "MULT"
   Ele2 = "name2"
   Ele3 = "name3"
   The multipole set "name3" is set to the values of the set "name2". random errors are not influenced in this case.

Remarks

1. A simple change of the starting point, by placing a "GO" somewhere in structure, used to change the machine optics as the random numbers were shifted, too. Simply calling this block even without a data line, will always fix the sequence of random numbers to start at the first multipole in the structure.

2. This input data block must follow the definition of the multipole block, otherwise multipoles cannot be set equal (option 3).

3. Do not use the keyword "MULT" in the single element list (3.2.1).

3.4.3 Combination of Elements

Description It is often necessary to use several families of magnetic elements with a certain ratio R of magnetic strength to perform corrections like tune adjustment (3.5.2), chromaticity correction (3.5.3) or resonance compensation (3.5.8). The *Combination of Elements* input block allows such a combination of elements. The maximum number of elements is defined by the parameter NCOM (see Appendix B.2).

Keyword COMB

Number of data lines variable

Format *e0 R1 e1 . . . Rn en*

e0 Reference element which appears in the input of the processing procedure

e1, . . . , en Elements to be combined with *e0*

Rj Ratio of the magnetic strength of element *ej* to that of element *e0*

3.5 Processing

This section comprises all the input blocks that do some kind of pre– or post–processing.

3.5.1 Linear Optics Calculation

Description The linear optics calculation input block is used to make a printout of all linear parameters (magnet lengths, β and α functions, tunes, dispersion and closed orbit) in the horizontal and vertical planes at the end of each element or linear block. The number of elements or blocks can be chosen.

Keyword LINE

Number of data lines variable but at least 1

Format

- first data line: *mode number–of–blocks ilin ntco E_I E_II*
- other data lines: *name(1), . . . , name(nlin)*

mode "ELEMENT" for a printout after each single element (3.2.1); "BLOCK" for a printout after each structure block (3.2.2)

number–of–blocks (integer) The number of the blocks in the structure to which the linear parameter will be printed. If this number is set to zero or is larger than the number of blocks, the complete structure will be calculated.

ilin (integer) Logical switch to calculate the traditional linear optics calculation in 4D ($1 = ilin$) and with the DA approach 6D ($2 = ilin$).

ntco (integer) A switch to write out linear coupling parameters.

- $ntco = 0$: no write–out
- $ntco \neq 0$: write–out of all linear coupled (4D) parameters including the coupling angle. These parameters (name, longitudinal position, the phase advances at that location, 4 β–, α– and γ–functions, 4 angles for coordinates and momenta respectively, plus the coupling angle [rad]) are written in binary format on file # 11.

E_I, E_II (floats) The two eigen–emittances to be chosen to determine the coupling angle. They are typically set to be equal.

names (char) For *nlin* ($\leq nele$) element– and block names the linear parameters are printed whenever they appear in the accelerator structure.

Remarks

1. To make this block work the Tracking Parameter block (3.6.1) has to used as well.
2. When the "ELEMENT 0" option is used a file unit # 34 is written with the longitudinal position, name, element type, multipole strength, β functions and phase advances in the horizontal and vertical phase space respectively. This file is used as input for the "SODD" program [20] to calculate detuning and distortion terms in first and second order. A full program suite can be found at: /afs/cern.ch/group/si/slap/share/sodd
3. If the "BLOCK" option has been used, the tunes may be wrong by a multiple of 1/2. This option is not active in the DA part (*2 = ilin*), which also ignores the (*NTCO*) option.

3.5.2 Tune Variation

Description This input block initializes a tune adjustment with zero length quadrupoles. This is normally done with two families of focusing and defocusing quadrupoles. It may be necessary, however, to have a fixed phase advance between certain positions in the machine. This can be done with this block by splitting the corresponding family into two sub–families which then are adjusted to give the desired phase advance.

Keyword TUNE

Number of data lines 2 or 4

Format

- data lines 1: *name1 Qx iqmod6*
- data lines 2: *name2 Qy*
- data lines 3 and 4, optional: *name3* ΔQ and *name4 name5* respectively

name1, name2 Names of focusing and defocusing quadrupole families respectively (in the single element list (3.2.1.1)

Qx, Qy (floats) Horizontal and vertical tune *including* the integer part

iqmod6 (integer) Logical switch to calculate the tunes in the traditional manner (*1 = iqmod6*) and with the DA approach including the beam-beam kick (*2 = iqmod6*).

name3 Name of the second sub–family, where the first sub–family is one of the above (*name1* or *name2*) This second sub–family replaces the elements of the first sub–family between the positions marked by *name4* and *name5*.

ΔQ Extra phase advance *including* the integer part (horizontal or vertical depending on the first sub–family) between the positions in the machine marked by *name4* and *name5*

name4, name5 Two markers in the machine for the phase advance ΔQ with the elements of the second sub–family between them

Remark The integer has to be included as the full phase advance around the machine is calculated by the program.

3.5.3 Chromaticity Correction

Description The chromaticity can be adjusted to desired values with two sextupole family using this input block.

Keyword CHRO

Number of data lines 2

Format data lines 1: *name1* Q'_x *ichrom*

Format data lines 2: *name2* Q'_y

name1/2 Names (in the single element list (3.2.1.2) of the two sextupole families

Q' Desired values of the chromaticity: $Q' = \frac{\delta Q}{\delta(\frac{\Delta p}{p_o})}$.

ichrom (integer) Logical switch to calculate the traditional chromaticity calculation (*1 = ichrom*) and with the DA approach including the beam-beam kick (*2 = ichrom*).

Remark To make the chromaticity correction work well a small momentum spread is required (DE0 in table (3.1)). It sometimes is required to optimize this spread.

3.5.4 Orbit Correction

Description Due to dipole errors in a real accelerator a closed orbit different from the beam axis is unavoidable. Even after careful adjustment one always will be left over with some random deviation of the closed orbit around the zero position. A closed orbit is introduced by nonzero strengths of b_1 and a_1 components of the multipole block (3.3.1), horizontal and vertical dipole kicks (3.2.1.2) or displacements of nonlinear elements (3.2.4). This input data block allows the correction of a such a random distributed closed orbit using he first two types in a "most effective corrector strategy" [24]. For that purpose correctors have to be denoted by *"HCOR= "* and *"VCOR= "* and monitors by *"HMON= "* and *"VMON= "* for the horizontal and vertical plane respectively. After correction the orbit is scaled to the desired r.m.s. values unless they are zero.

On file unit 28 the horizontal orbit displacement, measured at the horizontal monitors, will be written together with the monitor number, on file unit 29 the same is done for the vertical closed orbit displacement.

Keyword ORBI

Number of data lines variable but at least 1

Format

- first data line: *sigmax sigmay ncorru ncorrep*
- other data lines: *"HCOR= " namec* or *"HMON= " namem* or *"VCOR= " namec* or *"VMON= " namem*

sigmax, sigmay Desired r.m.s.–values of the randomly distributed closed orbit

ncorru Number of correctors to be used

ncorrep Number of corrections
If *ncorrep=0* the correction is iterated until *ITCO* (see table 3.1) iterations or after the both desired r.m.s.–values have been reached.

"HCOR= " namec Horizontal correction element of name *namec*

"HMON= " namem Horizontal monitor for the closed orbit of name *namem*

"VCOR= " namec Vertical correction element of name *namec*

"VMON= " namem Vertical monitor for the closed orbit of name *namem*

Remarks

1. Elements can have only one extra functionality: either horizontal corrector, horizontal monitor, vertical corrector or vertical monitor. If the number of monitors in a plane is smaller than the number of correctors it is likely to encounter numerical problems.
2. The *"HCOR= "*, *"HMON= "*, *"VCOR= "* and *"VMON= "* must be separated from the following name by at least one space.

3.5.5 Decoupling of Motion in the Transverse Planes

Description Skew–quadrupole components in the lattice create a linear coupling between the transverse planes of motion. A decoupling can be achieved with this block using four independent families of skew–quadrupoles, which cancel the off–diagonal parts of the transfer map. As these skew–quadrupoles also influence the tunes an adjustment of the tunes is performed at the same time.

Keyword DECO

Number of data lines 3

Format

- first data line: *name1,name2,name3,name4*
- data lines 2 and 3: *name5 Qx* and *name6 Qy* respectively

name1,2,3,4 Names of the four skew–quadrupole families

name5, name6 Names of focusing and defocusing quadrupole families respectively (in the single element list (3.2.1.1)

Qx, Qy (floats) Horizontal and vertical tune *including* the integer part

Remark A decoupling can also be achieved by compensating skew–resonances (3.5.8). The two approaches, however, are not always equivalent. In the resonance approach the zeroth harmonic is compensated, whilst a decoupling also takes into account the higher–order terms.

3.5.6 Sub–resonance Calculation

Description First order resonance widths of multipoles from second to ninth order are calculated following the approach of Guignard [10]. This includes resonances, which are a multiple of two lower than the order of the multipole. The first order detuning including feed–down from closed orbit is calculated from all multipoles up to to tenth order.

Keyword SUBR

Number of data lines 1

Format *n1 n2 Qx Qy Ax Ay Ip length*

n1, n2 (integers) Lowest and highest order of the resonance

Qx, Qy Horizontal and vertical tune including the integer part

Ax, Ay Horizontal and vertical amplitudes in mm

Ip (integer) Is a switch to change the nearest distance to the resonance $e = nxQx + nyQy$. In cases of structure resonances a change of p by one unit may be useful.

- $ip = 0$: e is unchanged
- $ip = 1$: $(e \pm 1) = nxQx + nyQy - (p \pm 1)$

length Length of the accelerator in meters

3.5.7 Search for Optimum Places to Compensate Resonances

Description To be able to compensate a specific resonance one has to know how a correcting multipole affects the cosine and sine like terms of the resonance width at a given position in the ring. This input data block can be used to find best places for the compensation of up to three different resonances, by calculating the contribution to the resonance width for a variable number of positions. For each position the effect of a fixed and small change of magnetic strength on those resonance widths is tested.

Keyword SEAR

Number of data lines variable but at least 2

Format

- data line 1: *Qx Qy Ax Ay length*
- data line 2: *npos n ny1 ny2 ny3 ip1 ip2 ip3* (integers)
- data lines from 3 on: *name1, . . . , namen*

Qx, Qy Horizontal and vertical tune including the integer part

Ax, Ay Horizontal and vertical amplitudes in mm

length Length of the accelerator in m

npos Number of positions to be checked

n Order of the resonance

ny1, ny2, ny3 Define three resonances of order n via : $nxQx + nyQy = p$ with $|nx| + |ny| = n$

ip1,ip2,ip3 The distance to a resonance is changed by an integer ip for each of the three resonances: $e = nxQx + nyQy - (p + ip)$.

namei i'th name of a multipole of order n , which has to appear in the single element list (3.2.1.2)

3.5.8 Resonance Compensation

Description The input block allows the compensation of up to three different resonances of order n simultaneously the chromaticity and the tunes can be adjusted. For mostly academic interest there is also the possibility to consider sub–resonances which come from multipoles which are a multiple of 2 larger than the resonance order n. However, it must be stated that the sub–resonances depend differently on the amplitude compared to resonances where the order of the resonances is the same as that of the multipoles.

Keyword RESO

Number of data lines 6

Format

- data line 1: *nr n ny1 ny2 ny3 ip1 ip2 ip3* (integers)
- data line 2: *nrs ns1 ns2 ns3* (integers)
- data line 3: *length Qx Qy Ax Ay*
- data line 4: *name1, ..., name6*
- data line 5: *nch name7 name8*
- data line 6: *nq name9 name10 Qx0 Qy0*

nr Number of resonances (0 to 3)

n Order of the resonance, which is limited to $nrco = 5$ (see list of parameters in Appendix B.2).
normal: $3 \leq n \leq nrco$; skew: $2 \leq n \leq nrco$

ny1, ny2, ny3 Define three resonances of order n via : $nxQx + nyQy = p$ with $|nx| + |ny| = n$

ip1, ip2, ip3 The distance to the resonance e can be changed by an integer value:
$e = nxQx + nyQy - (p + ip)$.

nrs Number of sub–resonances (0 to 3)

ns1, ns2, ns3 Order of the multipole with $ns \leq 9$ and $(ns - n)/2 \in \mathbf{N}$

length Length of the machine in meters

Qx, Qy Horizontal and vertical tune including the integer part

Ax, Ay Horizontal and vertical amplitudes in mm

name1, ..., name6 Names (3.2.1.2) of the correction multipoles for the first, second and third resonance

nch (integer) Switch for the chromaticity correction (0 = off, 1 = on)

name7, name8 Names (3.2.1.2) of the families of sextupoles to correct the chromaticity

nq (integer) Switch for the tune adjustment (0 = off, 1 = on)

name9, name10 Names (3.2.1.1) of the families of quadrupoles to adjust the tune

Qx0, Qy0 Desired tune values including the integer part

3.5.9 Differential Algebra

Description This input block initiates the calculation of a one turn map using the LBL Differential Algebra package [1]. The use of this block inhibits post–processing. The same differential algebra tools allow a subsequent normal form analysis (see [16]). A four–dimensional version integrated in SixTrack is available as described in sections 3.5.10 and 3.5.11.

Keyword DIFF.

Number of data lines 1 or 2

Format

- data line 1: *nord nvar preda nsix ncor*
- data line 2: *name(1),...,name(ncor)*

nord (integer) Order of the map

nvar (integer) Number of the variables (2 to 6). $nvar = 2{,}4{,}6$: two– and four–dimensional transverse motion and full six–dimensional phase space respectively. $nvar = 5$: four–dimensional transverse motion plus the relative momentum deviation $\frac{\Delta p}{p_o}$ as a parameter.

preda Precision needed by the DA package, usually set to $preda$ = 1e-38

nsix (integer) switch to calculate a 5×6 instead of a 6×6 map. This saves computational time and memory space, as the machine can be treated up to the cavity as five–dimensional (constant momentum).

- $nsix = 0$: $6x6$ map
- $nsix = 1$: $5x6$ map
  (*nvar* must be set to 6; 6D closed orbit must not be calculated, i.e. *iclo6 = 0* (3.6.2) and the map calculation is stopped once a cavity has been reached and being evaluated.)

ncor (integer) Number of zero–length elements to be additional parameters besides the transverse and/or longitudinal coordinates (i.e. two–, four–, five– or six–dimensional phase space).

name(i) (char) *Ncor* names (3.2.1.2) of zero–length elements (e.g dipole kicks, quadrupole kicks, sextupoles kicks etc.).

Remarks

- For *nsix = 1* the map can only be calculated till a cavity is reached.
- If the 6D closed orbit is calculated, the $5x6$ map can not be done, *nsix* is therefore forced to 0.
- If nvar is set to 5, the momentum dependence is determined without the need for including a fake cavity. With other words: the linear blocks are automatically broken up into single linear elements so that the momentum dependence can be calculated.
- If a DA map is needed at some longitudinal location one just has to introduce an element denoted “DAMAP” at that place in the structure, “DAMAP” has also to appear as a marker (zero length, element type = 0) in the single element list (3.2.1.2). This extra map is written to file # 17.

3.5.10 Normal Forms

Description All the parameters to compute the Normal Form of a truncated one–turn map are given in the *Normal Form* input block. Details on these procedures including the next block 3.5.11 can be found in reference [25].

Keyword NORM

Number of data lines 1

Format

- first data line: *nord nvar*

nord (integer) Order of the Normal Form

nvar (integer) Number of variables

Remarks

- The *Normal Form* input block has to be used in conjunction with the *Differential Algebra* input block that computes the one–turn map of the accelerator.
- The value of the parameter *nord* should not exceed the order specified for the transfer map plus one.
- The value of the parameter *nvar* should be equal to the number of coordinates used to compute the map plus eventually the number of correctors specified in the *Differential Algebra* input block.
- the value 1 for the off–momentum order is forbidden. This case corresponds to the linear chromaticity correction. It is in fact corrected by default when $par1 = 1$ or $par2 = 2$.

3.5.11 Corrections

Description All the parameters to optimise the tune–shift using a set of correctors are given in the *Correction* input block. (For details see reference [25].)

Keyword CORR

Number of data lines 3

Format

- first data line: *ctype ncor*
- second data line: *name(1),...,name(ncor)*
- third data line: *par1,...,par5*

ctype (integer) Correction type :

- ctype = 0 order–by–order correction
- ctype = 1 global correction

ncor (integer) Number of zero–length elements to be used as correctors in the optimisation of the tune–shift.

name(i) (char) *Ncor* names of zero–length elements (e.g sextupoles kicks, octupoles kicks etc.).

par1,...,par5 Parameters for the correction. Their meaning depend on the value of *ctype* and is explained in the following table:

Remarks

- The names of the elements specified in the *Correction* input block should be grouped according to the multipole type: first sextupoles, then octupoles ... etc.
- In case of order–by–order corrections, at least one of the quantities $par1, par2$ has to be zero, i.e. the correction of tune–shift terms depending on both amplitude and momentum is not allowed (as stated in the previous section).

	par1	par2	par3	par4	par5
variable type	integer	integer	real	real	real
ctype = 0	tune–shift order ≤ 2	off–momentum order ≤ 3	0.0	0.0	0.0
ctype = 1	$N_{min} \geq 2$	$N_{max} \leq 3$	α_H	α_V	δ_0

3.5.12 Post–processing

Description It has been seen in the past that the tracking data hold a large amount of information which should be extracted for a thorough understanding of the nonlinear motion. It is therefore necessary to store the tracking data turn by turn and post–process it after the tracking has been finished. The following quantities are calculated:

1. **Lyapunov exponent analysis** This allows to decide if the motion is of regular or chaotic nature, and, in the later case, that the particle will ultimately be lost. This is done with the following procedure:
   (a) Start the analysis where the distance in phase space of the two particles reaches its minimum.
   (b) Study the increase in a double logarithmic scale so that the slope in a regular case is always one, while a exponential increase stays exponential when we have chaos.
   (c) Average the distance in phase space to reduce local fluctuations, as we are interested in a long range effect.
   (d) Make a weighted linear fit with an increasing number of averaged values of distance in phase space, so that an exponential increase results in a slope that is larger than one and is increasing. (The weighting stresses the importance of values at large turn numbers).
2. **Analysis of the tunes** This is done either by the averaged phase advance method leading to very precise values of the horizontal and vertical tunes. A FFT analysis is also done. With the second method one can evaluate the relative strength of resonances, rather than achieve a precise tune measurement. In both cases the nearby resonances are determined.
3. **Smear** The smear of the horizontal and vertical emittances and the sum of the emittances are calculated in case of linearly coupled and un–coupled motion.
4. **Nonlinear Invariants** A rough estimate of the nonlinear invariants are given.
5. **Plotting** The processed tracking data can be plotted in different ways:
   (a) The distance of phase space as a function of amplitude
   (b) Phase space plots
   (c) Stroboscoped phase space
   (d) FFT amplitudes
6. **Summary** The post–processing results for a complete tracking session with varying initial parameters are summarised in a table at the end of the run.

Keyword POST

Number of data lines 4

Format

- data line 1: *comment title*
- data line 2: *iav nstart nstop iwg dphix dphiy iskip iconv imad cma1 cma2* (general parameters)
- data line 3: *Qx0 Qy0 ivox ivoy ires dres ifh dfft* (parameters for the tune calculation)
- data line 4: *kwtype itf icr idis icow istw iffw nprint ndafi* (integer parameters for the plotting)

iav (integer) Averaging interval of the values of the distance in phase space. Typically a tenth of the total turn number should be used as this interval.

nstart, nstop (integers) Start and stop turn number for the analysis of the post–processing (0 0 = all data used).

iwg (integer) Switch for the weighting of the slope calculation of the distance in phase space (0 = off, 1 = on).

dphix, dphiy Horizontal and vertical angle interval in radians that is used to stroboscope phase space. This stroboscoping of one of the two phase space projections is done by restricting the angle in the other phase space respectively to lie inside ± *dphix* or ± *dphiy*.

iskip (integer) This parameter allows to reduce the number of data to be processed: only each *iskip* sample of data will be used.

iconv (integer) If *iconv* is set to 1 the tracking data are not normalised linearly. Sometimes it is necessary to compare normalised to unnormalised data as the later will be found in the real machine.

imad (integer) This parameters is useful when MAD data shall be analysed (*imad* set to one).

cma1, cma2 (floats) To improve the Lyapunov analysis for MAD data and in the case that the motion is 6D but the 6D closed orbit is not calculated the off–momentum and the path–length difference ($\sigma = s - v_o \times t$) can be scaled with *cma1* and *cma2* respectively (see also 3.6.3). Please set both to 1. when the 6D closed orbit is calculated.

Qx0, Qy0 (floats) Values of the horizontal and vertical tune respectively (integer part) to be added to the averaged phase advance and to the Q values of the FFT analysis.

ivox, ivoy (integers) The tunes from the average phase advance are difficult to be calculated when this phase advance is strongly changing from turn to turn and when the tune is close to 0.5, as then the phase may become negative leading to a deviation of one unit. This problem can partly be overcome by setting these switches in the following way:

- tune close to an integer: *ivox, ivoy* = 1
- tune close to half an integer: *ivox, ivoy* = 0

ires, dres (integer,float) For the calculated tune values from the average phase advance method and the FFT–routine the closest resonances are searched up to *ires*'th order and inside a maximum distance to the resonance *dres*, so that $nxQx + nyQy < dres$ and $nx + ny \leq ires$.

ifh, dfft (integer,float) For the FFT analysis the tune interval can be chosen with *ifh*. To find resonances with the FFT spectrum, all peaks below a fraction *dfft* of the maximum peak are accepted.

- *ifh* $= 0 : 0 \leq Q \leq 1$
- *ifh* $= 1 : 0 \leq Q \leq 0.5$
- *ifh* $= 2 : 0.5 \leq Q \leq 1$

kwtype – Disabled, set to 0 – (Terminal type, e.g. 7878 for the Pericom graphic terminals. For details, consult the HPLOT manual [8].)

itf Switch to get PS–file of plots

- *itf* $= 0$: off
- *itf* $= 1$: on

icr – Disabled, set to 0 – (Switch to stop after each plot (0 = no stop, 1 = stop after each plot).

idis, icow, istw, iffw Switches (0 = off) to select the different plots. If all values are set to zero, the HBOOK/HPLOT routine will not be called.

- *idis* $= 1$: plot of distance in phase space
- *icow* $= 1$: a set of plots of projections of the six–dimensional phase space and the energy E versus the turn number
- *istw* $= 1$: plot of the stroboscoped phase space projection by restricting the phase in the other phase space projection
- *iffw* $= 1$: plots of the horizontal and vertical FFT spectrum with linear amplitude scale
- *iffw* $= 2$: plots of the horizontal and vertical FFT spectrum with logarithmic amplitude scale

nprint Switch to stop the printing of the post–processing output to unit 6 (0 = printing off, 1 = printing on).

ndafi Number of data–files to be processed (units : from 90 to (90–ndafi+1)).

Remarks

1. The post–processing can be done in two ways :
   (a) directly following a tracking run by adding this input block to the input blocks of the tracking
   (b) as a later run where the tracking parameter file (unit # 3) consists of only the *Program Version* input block 3.1.1 (using the *FREE* option) and of this input block specifying the post–processing parameters followed by *ENDE* as usual
2. The HBOOK/HPLOT routines are only used at the start of the main program for initialisation and termination. The actual plots are done in the post–processing subroutine. The routines are activated only if at least one of the plotting parameters (*idis, icow, istw, iffw*) is set to one.

3.6 Initial Conditions for Tracking

Description For the study of nonlinear system the choice of initial conditions is of crucial importance. The input structure for the initial conditions was therefore organise in such a way as to allow for maximum flexibility. SixTrack is optimised to reach the largest possible number of turns. In order to derive the Lyapunov exponent and thereby to distinguish between regular and chaotic motion, the particle has a close by companion particle. Moreover, experience has shown that varying only the amplitude while keeping the phases constant is sufficient to understand the nonlinear dynamics, as a subsequent detailed post–processing allows to find the dependence of the parameter of interest on these phases.

3.6.1 Tracking Parameters

Description All tracking parameters are defined with this input block, the initial coordinates are generally set here, too. A fine tuning of the initial condition is done with Initial Coordinates block (3.6.2) and the parameters for the synchrotron oscillation are given in block (3.6.3)

Keyword TRAC

Number of data lines 3

Format

- data line 1: *numl numlr napx amp(1) amp0 ird imc*
- data line 2: *idy(1) idy(2) idfor irew iclo6* (integers)
- data line 3: *nde(1) nde(2) nwr(1) nwr(2) nwr(3) nwr(4) ntwin ibidu iexact* (integers)

Missing variables on line 1

numl (integer) Number of turns in the forward direction

numlr (integer) Number of turns in the backward direction

napx (integer) Number of amplitude variations

amp(1), amp0 (floats) Start and end amplitude (any sign) in the horizontal phase space plane for the amplitude variations. The vertical amplitude is calculated using the ratio between the horizontal and vertical emittance set in the *Initial Coordinates* block (3.6.2), where the initial phase in phase space are also set. Additional information can be found in the *Remarks*.

ird (integer) Switch for the type of amplitude variation. In case $napx = 1$ the amplitude nstart is used.

- $ird = 0$: amplitudes are varied between the amplitudes *amp(1)* and *amp0* with equal increments:
$$delta = (amp0 - amp(1))/(napx - 1)$$
- $ird = 1$: amplitude variation to find an estimate for the short term dynamic aperture. The amplitude is increased or decremented corresponding to stable motion or particle loss respectively. The change of amplitude is reduced each iteration $i \leq (napx - 1)$ to:
$$delta = (amp0 - amp(1))/2^i$$

imc (integer) Number of variations of the relative momentum deviation $\frac{\Delta p}{p_o}$. The maximum value of the relative momentum deviation $\frac{\Delta p}{p_o}$ is taken from that of the first particle in the *Initial Coordinates* block (3.6.2). The variation will be between $\pm\frac{\Delta p}{p_o}$(max) in steps of $\frac{\Delta p}{p_o}$(max) / (*imc–1*).

idy(1), idy(2) A tracking where one of the transversal motion planes shall be ignored is only possible when all coupling terms are switched off. The part of the coupling that is due to closed orbit and other effects can be turned off with these switches.

- *idy(1), idy(2)* $= 1$: coupling on
- *idy(1), idy(2)* $= 0$: coupling to the horizontal and vertical motion plane respectively switched off

idfor Usually the closed orbit is added to the initial coordinates. This can be turned off using *idfor*, for instance when a run is to be prolonged.

- *idfor* $= 0$: closed orbit added
- *idfor* $= 1$: initial coordinates unchanged
- *idfor* $= 2$: prolongation of a run, taken the initial coordinates from unit # 13

irew To reduce the amount of tracking data after each amplitude and relative momentum deviation iteration $\frac{\Delta p}{p_o}$ the binary output units 90 and lower (see Appendix C) are rewound. This is always done when the post–processing is activated (3.5.12). For certain applications it may be useful to store all data. The switch *irew* allows for that.

- *irew* $= 0$: unit 90 (and lower) rewound
- *irew* $= 1$: all data on unit 90 (and lower)

iclo6 This switch allows to calculate the 6D closed orbit using the differential algebra package. It is ignored in the regular tracking versions. It is active in all versions that link to the Differential Algebra package. This 6D closed orbit can be calculated from any longitudinal position contrary to earlier versions.

- *iclo6* $= 0$: switched off
- *iclo6* $= 1$: calculated
- *iclo6* $= 2$: calculated and added to the initial coordinates (3.6.2).
- *iclo6* $= 5$ or $=6$: like for *1* and *2* but in addition a guess closed orbit is read (in free format) from file unit # 33.

nde(1) Number of turns at flat bottom, useful for energy ramping.

nde(2) Number of turns for the energy ramping. *numl–nde(2)* gives the number of turns on the flat top. For constant energy with $nde(1) = nde(2) = 0$ the particles are considered to be on the flat top.

nwr(1) Every *nwr(1)*'th turn the coordinates will be written on unit 90 (and lower) in the flat bottom part of the tracking.

nwr(2) Every *nwr(2)*'th turn the coordinates in the ramping region will be written on unit 90 (and lower).

nwr(3) Every *nwr(3)*'th turn at the flat top a write out of the coordinates on unit 90 (and lower) will occur. For constant energy this number controls the amount of data on unit 90 (and lower), as the particles are considered on the flat top.

nwr(4) In cases of very long runs it is sometimes useful to save all coordinates for a prolongation of a run after a possible crash of the computer. Every *nwr(4)*'th turn the coordinates are written to unit 6.

ntwin For the analysis of the Lyapunov exponent it is usually sufficient to store the calculated distance of phase space together with the coordinate of the first particle (*ntwin* set to one). You may want to improve the 6D calculation of the distance in phase space with *sigcor, dpscor* (see 3.6.2) when the 6D closed orbit is not calculated with *iclo6* $\neq 2$. If storage space is no problem, one can store the coordinates of both particles (*ntwin* set to two). The distance in phase space is then calculated in the post–processing procedure (see 3.5.12). This also allows a subsequent refined Lyapunov analysis using differential–algebra and Lie–algebra techniques ([26]).

ibidu Switch to creat or read binary dump of the full accelerator decription on file # 32. The parameters relevant to tracking, i.e. *numl, amp0, amp(1), amp(2), damp, chi0, chid, rat,* x_1, x_1', y_1, y_1', σ_1, $\frac{\Delta p}{p_{o1}}$, x_2, x_2', y_2, y_2', σ_2, $\frac{\Delta p}{p_{o2}}$, *time0, time1*, are to be given via the tracking parameter file # 3.

- *ibidu* $= 1$: write dump
- *ibidu* $= 2$: read dump

iexact Switch to enable exact solution of the equation of motion into tracking and 6D (no 4D) optics calculations.

- *ibidu* $= 0$: approximated equation (e.g. $x' \simeq \frac{P_x}{P_0(1+\delta)}$, $y' \simeq \frac{P_y}{P_0(1+\delta)}$);
- *ibidu* $= 1$: exact equation (e.g $x' \simeq \frac{P_x}{P_0\sqrt{(1+\delta)^2-P_x^2-P_y^2}}$, $y' \simeq \frac{P_y}{P_0\sqrt{(1+\delta)^2-P_x^2-P_y^2}}$).

Remarks

1. This input data block is usually combined with the *Initial Coordinates* input block (3.6.2) to allow a flexible choice of the initial coordinates for the tracking.
2. For a prolongation of a run the following parameters have to be set :
   - in this input block : idfor = 1
   - in the *Initial coordinates* input block :
     (a) itra = 0
     (b) take the end coordinates of the previous run as the initial coordinates (including all digits) for the new run.
3. A feature is installed for a prolongation of a run by using *idfor = 2* and reading the initial data from unit # 13. The end coordinates are now written on unit # 12 after each run. Intermediate coordinates are also written on unit # 12 in case the turn number *nwr(4)* is exceeded in the run. The user takes responsibility to transfer the required data from unit # 12 to unit # 13 if a prolongation is requested.
4. Some illogical combinations of parameters have been suppressed.
5. The initial coordinates are calculated using a proper linear 6D transformation: *amp(1)* is still the maximum horizontal starting amplitude (excluding the dispersion contribution) from which the emittance of mode 1 e_I is derived, *rat* (see 3.6.2) is the ratio of e_{II}/e_I of the emittances of the two modes. The momentum deviation $\frac{\Delta p}{p_{o1}}$ is used to define a longitudinal amplitude. The 6 normalized coordinates read:
   - horizontal:

     $\sqrt{e_I} = \frac{amp(1)}{\sqrt{\beta_{xI}+|rat|\times\beta_{xII}}}$,

     0.
   - vertical:

     $sign(rat) \times \sqrt{e_{II}}$, with $e_{II} = |rat| \times e_I$,

     0.
   - longitudinal:

     0.,

     $\frac{\Delta p}{p_{o1}} \times \sqrt{\beta_{sIII}}$

   and are then transformed with the 6D linear transformation into real space. Note that results may differ from those of older versions.

3.6.2 Initial Coordinates

Description The *Initial Coordinates* input block is meant to manipulate how the initial coordinates are organise, which are generally set in the tracking parameter block (3.6.1). Number of particles, initial phase, ratio of the horizontal and vertical emittances and increments of 2 × 6 coordinates of the two particles, the reference energy and the starting energy for the two particles.

Keyword INIT

Number of data lines 16

Format

- first data line: *itra chi0 chid rat iver*
- data lines 2 to 16: *15 initial coordinates in table* 3.5

itra (integer) Number of particles

- *itra* = 0 : Amplitude values of tracking parameter block (3.6.1) are ignored and coordinates of data line 2–16 are taken. *itra* is set internally to 2 for tracking with two particles. This is necessary in case a run is to be prolonged.
- *itra* = 1 : Tracking of one particle, twin particle ignored
- *itra* = 2 : Tracking the two twin particles

chi0 Starting phase of the initial coordinate in the horizontal and vertical phase space projections

chid Phase difference between first and second particles

rat Denotes the emittance ratio (e_{II}/e_I) of horizontal and vertical motion. For further information see the *Remarks* of input block (3.6.1).

iver In tracking with coupling it is sometimes desired to start with zero vertical amplitude which can be painful if the emittance ratio *rat* is used to achieve it. For this purpose the switch *iver* has been introduced:

- *iver* = 0 : Vertical coordinates unchanged
- *iver* = 1 : Vertical coordinates set to zero.

Remarks

1. These 15 coordinates are taken as the initial coordinates if *itra* is set to zero (see above). If *itra* is 1 or 2 these coordinates are added to the initial coordinates generally defined in the tracking parameter block (3.6.1). This procedure seems complicated but it allows freely to define the initial difference between the two twin particles. It also allows in case a tracking run should be prolonged to continue with precisely the same coordinates. This is important as small difference may lead to largely different results.
2. The reference particle is the particle in the centre of the bucket which performs no synchrotron oscillations.
3. The energy of the first and second particles is given explicitly, again to make possible a continuation that leads precisely to the same results as if the run would not have been interrupted.
4. There is a refined way of prolonging a run, see the *Tracking Parameters* input block (3.6.1).

3.6.3 Synchrotron Oscillation

Description The parameters needed for treating the synchrotron oscillation in a symplectic manner are given in the *Synchrotron Oscillation* input block.

Keyword SYNC

Number of data lines 2

Format

- first data line: *harm alc u0 phag tlen pma ition dppoff*
- second data line: *dpscor sigcor*

harm Harmonic number

alc Momentum compaction factor, used here only to calculate the linear synchrotron tune Q_S.

u0 Circumference voltage in [MV]

phag Acceleration phase in degrees

tlen Length of the accelerator in meters

pma rest mass of the particle in MeV/c^2

ition (integer) Transition energy switch

- *ition* = 0 for no synchrotron oscillation (energy ramping still possible)
- *ition* = 1 for above transition energy
- *ition* = –1 for below transition energy

dppoff Offset Relative Momentum Deviation $\frac{\Delta p}{p_o}$: a fixpoint with respect to synchrotron oscillations. It becomes active when the 6D closed orbit is calculated (see item *iclo6* in section 3.6.1).

dpscor, sigcor Scaling factor for relative momentum deviation $\frac{\Delta p}{p_o}$ and the path length difference ($\sigma = s - v_o \times t$) respectively. They can be used to improve the calculation of the 6D distance in phase space, but is only used when *ntwin = 1* in the tracking parameter input block (3.6.1). Please set to 1 when the 6D closed is calculated.

3.7 Extra output files

Write

3.7.1 Dumping of beam population

Descrioption

Keyword

Format

Examples

Table 3.5: Initial Coordinates of the 2 Particles

data line	contents
2	x_1 [mm] coordinate of particle 1
3	x_1' [mrad] coordinate of particle 1
4	y_1 [mm] coordinate of particle 1
5	y_1' [mrad] coordinate of particle 1
6	path length difference 1 ($\sigma_1 = s - v_o \times t$) [mm] of particle 1
7	$\frac{\Delta p}{p_{o1}}$ of particle 1
8	x_2 [mm] coordinate of particle 2
9	x_2' [mrad] coordinate of particle 2
10	y_2 [mm] coordinate of particle 2
11	y_2' [mrad] coordinate of particle 2
12	path length difference ($\sigma_2 = s - v_o \times t$) [mm] of particle 2
13	$\frac{\Delta p}{p_{o2}}$ of particle 2
14	energy [MeV] of the reference particle
15	energy [MeV] of particle 1
16	energy [MeV] of particle 2

Conclusions

Programs with large input structures like SixTrack tend to be far from perfect, even though a cumbersome chase for program bugs and a lot of polishing on the input structure has been performed. Plenty of comments and suggestions are therefore needed to further improve the program.

Chapter 4

Acknowledgement

I would like to thank my colleagues at DESY and CERN to help to find nasty bugs and for a thorough check of the program. I would like to thank Mikko Vaenttinen who helped to vectorise the program. He also did most of the typing of the manuscript. Moreover, I want to express my gratitude to F. Zimmermann who helped to finish the differential–algebra part in endless night sessions. Additions concerning Normal Forms have been contributed by M. Giovannozzi. J. Miles helped with the calculation of the 6D Courant–Snyder matrix and its use to transform the tracking data in the post–processing. W. Herr is thanked for providing a software package used for the orbit correction. L.H.A. Leunissen extracted and adapted the 6D beam–beam code of Hirata [18].

Appendix A

List of Keywords

Table A.1: List of Keywords

#	Keyword	Input–data–block		Short Description	§	Page
		Title	**# of Data–lines**			
1	BEAM	BEAM–BEAM Element	1	4-6D including Beam Separation & Linear Coupling	3.3.5	22
2	BLOC	Block–definition	variable + 1	Blocks of Linear Elements	3.2.2	12
3	BLOCK			Linear Parameters for each Structure Element	3.5.1	26
4	CAV			Cavity in the Structure Input Block	3.2.3	13
5	CHRO	Chromaticity Correction	2	Correcting Chromaticity with Sextupoles	3.5.3	27
6	CORR	Tune–shift Corrections	3	Correction of Nonlinear Tune–Shift	3.5.11	33
7	COMB	Combination of Elements	variable	Combining Different Elements for a Correction	3.4.3	25
8	COMM	Comment Line	1	Additional Comments	3.1.3	7
9	DAMAP			Location for a Printout of a DA map	3.5.9	31
10	DECO	Decoupling	3	Compensation of Linear Coupling	3.5.5	29
11	DIFF	Differential Algebra	1	Calculating a One–turn Map with Differential Algebra	3.5.9	31
12	DISP	Displacement of Elements	variable	Displacing Nonlinear Elements	3.2.4	13
13	EL			Elliptical Aperture Limitation	3.3.2	15
14	ELEMENT			Linear Parameters after each Single Element	3.5.1	26
15	ENDE			End of SixTrack Input Structure		
16	FLUC	Random Fluctuation Starting Number	1	Seed for the Random Generator	3.4.1	24
17	FREE	1^{st} Program Version	0	Free Format Input from one File	3.1.1	6
18	GEOM	2^{nd} Program Version	0	Input of Machine Geometry in extra File	3.1.1	6
19	GO			Start of Tracking in the Structure Input	3.2.3	13
20	“HCOR= ”			Specifies an Horizontal Orbit Corrector Element (Dipole or Multipole)	3.5.4	28
21	“HMON= ”			Specifies an Horizontal Orbit Monitor	3.5.4	28

#	Keyword	Input–data–block		Short Description	§	Page
		Title	**# of Data–lines**			
22	INIT	Initial Coordinates	16	Setting up of the Initial Coordinates	3.6.2	40
23	ITER	Iteration Errors	4	# of Iterations and Precision for Correction Routines	3.1.4	7
24	LIMI	Aperture Limitation	variable	Collimators that stop the Program when being hit	3.3.2	15
25	MULT	Multipole skew Coefficients	max. 11	Multipole Coefficients normal and up to 10^{th} order	3.3.1	14
				Combination of Different Multi–poles in the ORGA Input Block	3.4.2	25
26	NEXT			Last Line of each Input Data Block	3.5.4	28
27	NORM	Normal Form	1	Normal Form Operations on Maps	3.5.10	32
28	ORBI	Orbit Adjustment	variable	Adjusting Orbit to desired Sigma Values	3.5.4	28
29	ORGA	Organisation of Random Numbers	variable + 1	Arranging Random Errors and Multipole sets	3.4.2	25
30	POST	Post–processing	3	Post–processing of the Tracking Data	3.5.12	34
31	PRIN	Printout Selection	0	Initiates the Printing of the Input Data	3.1.2	6
32	RE			Rectangular Aperture Limitation	3.3.2	15
33	RESO	Resonance Compensation	6	Compensation of up to 3 Different Resonances	3.5.8	30
34	RIPP	Power Supply Ripple	variable	Invokes a Sinusoidal Tune Variation	3.3.3	15
35	SEAR	Search for Resonance Compensation Positions	variable	Evaluating Longitudinal Positions for a Resonance Compensation	3.5.7	30
36	SING	Single Elements	variable	Magnet Parameters of Single Elements	3.2.1	8
37	STRU	Structure Input	variable	Structure of Linear Blocks and Nonlinear Elements	3.2.3	13
38	SUBR	Sub–resonance Calculation	1	Calculation of 1^{th} Order Reso–nances up to 9^{th} Multipole Order	3.5.6	29
39	SYNC	Synchrotron Oscillations	2	Parameters concerning Synchrotrons Oscillation	3.6.3	41
40	TRAC	Tracking Parameters	3	All major Tracking Parameters for the transversal Motion Plane	3.6.1	37
41	TUNE	Tune Variation	2 or 4	Adjusting the Horizontal and Vertical Tunes	3.5.2	27
42	TROM	“Phase Trombone” element	mult. of 14	Phase Shift Transparent for Linear Optics	3.3.6	23
43	“VCOR= ”			Specifies an Vertical Orbit Corrector Element (Dipole or Multipole)	3.5.4	28
44	“VMON= ”			Specifies an Vertical Orbit Monitor	3.5.4	28

remove RIPP (or note obsolete), add DYNK and DUMP (alphabetic?)

Appendix B

List of Default Values

B.1 Default Tracking Parameters

Some of the parameters for tracking are set to non–zero values. This is done for instance to avoid as much as possible program errors such as division by zero due to an erroneous input. The default values for the *Iteration Errors* (3.1.4) see table 3.1.

Table B.1: Default Tracking Parameters

#	**Description**	**Value**	§	**Page**
1	General Aperture Limitations (horizontal and vertical)	1000 mm	3.3.2	15
2	Starting in the Accelerator Structure at Element Number	1	3.2.3	13
3	Number of Turns in the forward Direction	1	3.6.1	37
4	Initial horizontal Amplitude	0.001 mm		
5	Horizontal and vertical Phase Space Coupling Switches on	1		
6	Flat Bottom, Ramping and Flat Top Printout after Turn Number	1		
7	Printout of Coordinates (file 6) after Turn Number	10000		
8	Kinetic Energy [MeV] of the Reference Particle	10^{-6}	3.6.2	40
9	Harmonic Number	1	3.6.3	41
10	Momentum Compaction Factor	0.001		
11	Length of the Machine	1 km		
12	Mass of the Particle (Proton)	938.2723128 MeV/c^2		
13	Momentum Correction Factor for Distance in Phase Space	1		
14	Path–length Correction Factor for Distance in Phase Space	1		
15	Averaging Turn Interval for Post–processing	1	3.5.12	34

B.2 Default Size Parameters

For large machines the arrays holding the machine parameters might have to be increased. The size of each of the dimensions of the arrays is therefore defined as a parameter. The default values are adjusted to allow the treatment of a full LHC lattice: the tracking version uses 50 Mb and the DA version 400 Mb.

Table B.2: Default Size Parameters

#	Description	Value	Name	§	Page
1	Maximum Number of Coordinates used in the Correction Routines	6	MPA		
2	Number of Single Elements	750	NELE	3.2.1	8
3	Number of Blocks of Linear Elements	160	NBLO	3.2.2	12
4	Number of Linear Elements per Block	100	NELB		
5	Total Number of Elements in the Structure	15000	NBLZ	3.2.3	13
6	Number of Accelerator Super–periods	16	NPER		
7	Total Number of Random Values	300000	NZFZ	3.4.1	24
8	Number of Random Values for the basic Set of Nonlinear Elements	280000	NRAN		
9	Number of Random Values for inserted Nonlinear Elements	20000		3.4.2	25
10	Number of Random Values for each Inserted Nonlinear Element Number of Nonlinear Elements that can be inserted	500 20	MRAN		
11	Limit Number of Particles for Vectorisation	64	NPART		
12	Maximum Number of Elements for Combined Tasks	100	NCOM	3.4.3	25
13	Maximum Resonance Compensation Order	5	NRCO	3.4.3	25
14	Total Number of Data for Processing	20000	NPOS	3.5.12	34
15	Number of Intervals for Calculation of Lyapunov–Exponents	10000	NLYA		
16	Number of Intervals for Calculation of Invariants	1000	NINV		
17	Number of Data for Plotting	20000	NPLO		
18	Maximum Pole Order of Multipole Block	11	MMUL	3.3.1	14
19	Maximum Number of extra Parameters of the DA Map	10	MCOR	3.5.9	31
20	Maximum Order of DA Calculation	15	NEMA	3.5.9	31
21	Maximum Number of Monitors for Micado Closed Orbit Correction	600	NMON1	3.5.4	28
22	Maximum Number of Correctors for Micado Closed Orbit Correction	600	NCOR1	3.5.4	28
23	Maximum Number of Beam–Beam Elements	160	NBB	3.3.5	22
24	Maximum Number of Slices for 6D Beam–Beam Kick	15	MBEA	3.3.5	22
25	Maximum Number of “Phase Trombone” Elements	20	NTR	3.2.1.6	11

Appendix C

Input and Output Files

The program uses a couple of files for its input and output procedures.

Table C.1: List of Input and Output Files.

File Unit	**Input**	**Output**	**File Type**	**Contents**
2	X		Ascii	Geometry and Strength Parameters
3	X		Ascii	Tracking Parameters
4		X	Ascii	Geometry and strength Parameters (format as file # 2)
6		X	Ascii	Input Parameters and Analysis of Data
8	X		Ascii	Name, hor., ver. Misalignment and Tilt
9		X	Ascii	Internally used multipoles Format: $a16,\ 2 \times \{6 \times (1p, 3d23.15),\ (1p, 2d23.15)\}$
10	X	X	Ascii	Summary of Post–processing (auxiliary)
11		X	Binary	This file is used to dump linear coupling parameters at locations of choice
12		X	Ascii	End Coordinates of both Particles Format: $(15 \times F10.6)$
13	X		Ascii	Start Coordinates for a Prolongation
14		X	Ascii	Horizontal FFT Spectrum for detailed Analysis; Format: $(2 \times F10.6)$
15		X	Ascii	Vertical FFT Spectrum for detailed Analysis; Format: $(2 \times F10.6)$

File Unit	Input	Output	File Type	Contents
16	X		Ascii	External multipole errors Format: $a16,\ 2 \times \{6 \times (1p, 3d23.15),\ (1p, 2d23.15)\}$
17		X	Ascii	Additional Map at location of interest
18		X	Ascii	One–Turn Map with Differential Algebra
19	X	X	Ascii	Internal use for Differential Algebra
20		X	Meta–file	PS–file of selected Plots
21		X	Ascii	Factorisation of the one–turn map
22		X	Ascii	Transformation in the Normal Form coordinates
23		X	Ascii	Hamiltonian in action variables
24		X	Ascii	Tune–shift in action coordinates
25		X	Ascii	Tune–shift in Cartesian coordinates
26		X	Ascii	NAGLIB log–file
27		X	Ascii	Name, hor., ver. Misalignment and Tilt
28		X	Ascii	Horizontal closed orbit displacement, measured at monitors
29		X	Ascii	Vertical closed orbit displacement, measured at monitors
30	X		Ascii	Name, Random strength, misalignments and tilt
31		X	Ascii	Name, Random strength, misalignments and tilt
32	X	X	Binary	Binary dump of full accelerator description
33	X		Ascii	Guess values for 6D closed orbit search

File Unit	**Input**	**Output**	**File Type**	**Contents**
34		X	Ascii	Multipole strength and linear lattice parameters [20]
90 – k		X	Binary	Tracking Data $0 <= k <= 31$
98		X	Ascii	6D coordinates at Cavity (1p,6(2x,e25.18))

Update for dynk (input and output). Also update for Checkpoint/restart (92/93/95?) and collimation?

Appendix D

Data Structure of the Data–Files

A common data structure for the programs MAD and SixTrack is agreed on. Besides some minor differences this allows a straightforward post–processing of data from either program. Each binary data–file has a header which holds a description of the run with comments, tracking parameters and 50 additional parameters for future purposes, six of which are already specified in SixTrack.

Table D.1: Header of the Binary Data–Files

Data Type	**Bytes**	**Description**
Character	80	General title of the run
Character	80	Additional title
Character	8	Date
Character	8	Time
Character	8	Program name
Integer	4	First particle in the file
Integer	4	Last particle in the file
Integer	4	Total number of particles
Integer	4	Code for dimensionality of phase space 1,2,4 are hor., vert. and longitudinal respectively
Integer	4	Projected number of turns
Float	8	Horizontal Tune
Float	8	Vertical Tune
Float	8	Longitudinal Tune
Float	6 * 8	Closed Orbit vector
Float	6 * 8	Dispersion vector
Float	36 * 8	Six–dimensional transfer map
—— 50 additional parameters ——		
Float	8	Maximum number of different seeds
Float	8	Actual seed number
Float	8	Starting value of the seed
Float	8	Number of turns in the reverse direction (IBM only)
Float	8	Correction–factor for the Lyapunov $(\sigma = s - v_o \times t)$
Float	8	Correction–factor for the Lyapunov $(\frac{\Delta p}{p_o})$
Float	8	Start turn number for ripple prolongation
Float	43 * 8	Dummies

Following this header the tracking data are written in n samples of nine numbers preceded by the turn number. In the MAD format the number of samples n is not restricted, whilst SixTrack writes only up to two samples for the two particles for the Lyapunov–exponent method. Up to 64 particles (two per file) can be treated in the vectorised version of SixTrack.

Table D.2: Format of the Binary Data

Data Type	**Bytes**	**Description**
Integer	4	Turn number
—— One or two samples of 9 values are following ——		
Integer	4	Particle number
Float	8	Angular distance in phase space (<= 1)
Float	8	x (mm)
Float	8	x' (mrad)
Float	8	y (mm)
Float	8	y' (mrad)
Float	8	Path–length ($\sigma = s - v_o \times t$) (mm)
Float	8	Relative momentum deviation $\frac{\Delta p}{p_o}$
Float	8	Energy (MeV)

Some of the post–processing data are written in Ascii–format on file # 10. This can be used for instance for plotting purposes. Each time the post–processing routine is called 60 double precision numbers (some of them still dummy) are added to the file.

The file with the errors (in: fort.16, out: fort.9) has the following format: first line – name of element; line 2–7 – normal multipoles order 1–18; line 8 – normal multipoles of order 19 and 20; line 9–14 – skew multipoles order 1–18; line 15 – skew multipoles of order 19 and 20. The strength definition is according to block 3.3.1 and to be effective in fort.3 the random values of the corresponding multipole block have to be set to 1.0. A word of caution: when writing on file fort.9 the *total* multipole strength is used, i.e. systematic and random part combined. File fort.16 and fort.9 might therefore be different. When using fort.9 as input (fort.16) the systematic part in fort.3 has to be set to 0.0.

Misalignment and tilt are in file # 8 and # 27 as input and output respectively. The format is (a16,2x,1p,2d14.6,d17.9), i.e. name, horizontal misalignment, vertical misalignment and tilt. The misalignment is in units of [mm] the tilt in units of [mrad]. The files # 30 (in) and # 31 (out) have the random single nonlinear element kick, misalignments and tilt in the format: (a8,1p,d19.11,2d14.6,d17.9). Misalignment and tilt in file fort.8 or fort.30 is automatically activated while the random strength (strength definition same as in block 3.2.1) needs an entry in the fourth column in the geometry file fort.2. File # 28 and # 29 hold integer counter and closed orbit displacement at a horizontal or vertical monitor respectively.

Table D.3: Post–processing Data

# of Column	Description
1	Maximum turn number
2	Stability Flag (0=stable, 1=lost)
3	Horizontal Tune
4	Vertical Tune
5	Horizontal β–function
6	Vertical β–function
7	Horizontal amplitude 1^{st} particle
8	Vertical amplitude 1^{st} particle
9	Relative momentum deviation $\frac{\Delta p}{p_o}$
10	Final distance in phase space
11	Maximum slope of distance in phase space
12	Horizontal detuning
13	Spread of horizontal detuning
14	Vertical detuning
15	Spread of vertical detuning
16	Horizontal factor to nearest resonance
17	Vertical factor to nearest resonance
18	Order of nearest resonance
19	Horizontal smear
20	Vertical smear
21	Transverse smear
22	Survived turns 1^{st} particle
23	Survived turns 2^{nd} particle
24	Starting seed for random generator
25	Synchrotron tune
26	Horizontal amplitude 2^{nd} particle
27	Vertical amplitude 2^{nd} particle
28	Minimum horizontal amplitude
29	Mean horizontal amplitude
30	Maximum horizontal amplitude
31	Minimum vertical amplitude
32	Mean vertical amplitude
33	Maximum vertical amplitude
34	Minimum horizontal amplitude (linear decoupled)
35	Mean horizontal amplitude (linear decoupled)
36	Maximum horizontal amplitude (linear decoupled)
37	Minimum vertical amplitude (linear decoupled)
38	Mean vertical amplitude (linear decoupled)
39	Maximum vertical amplitude (linear decoupled)
40	Minimum horizontal amplitude (nonlinear decoupled)
41	Mean horizontal amplitude (nonlinear decoupled)
42	Maximum horizontal amplitude (nonlinear decoupled)
43	Minimum vertical amplitude (nonlinear decoupled)
44	Mean vertical amplitude (nonlinear decoupled)
45	Maximum vertical amplitude (nonlinear decoupled)
46	Emittance Mode I
47	Emittance Mode II
48	Secondary horizontal β–function
49	Secondary vertical β–function
50	Q'_x
51	Q'_y
52 – 58	Dummy
59 – 60	Internal use

As an option the 4D linear parameters can be dumped to file # 11 when the linear optics block 3.5.1 is activated. This can be used for instance for a post–processing of linear coupling. 25 values are written in a binary format.

Table D.4: 4D Linear Parameters

# of Column	**Description**
1	Name of the element
2	Longitudinal Position [m]
3	Horizontal phase advance
4	Vertical phase advance
5	Primary horizontal β–function [m]
6	Secondary horizontal β–function [m]
7	Secondary vertical β–function [m]
8	Primary vertical β–function [m]
9	Primary horizontal α–function [rad]
10	Secondary horizontal α–function [rad]
11	Secondary vertical α–function [rad]
12	Primary vertical α–function [rad]
13	Primary horizontal γ–function [m]
14	Secondary horizontal γ–function [m]
15	Secondary vertical γ–function [m]
16	Primary vertical γ–function [m]
17	Primary horizontal phase of x–coordinate [pi]
18	Secondary horizontal phase of x–coordinate [pi]
19	Secondary vertical phase of y–coordinate [pi]
20	Primary vertical phase of y–coordinate [pi]
21	Primary horizontal phase of x'–coordinate [pi]
22	Secondary horizontal phase of x'–coordinate [pi]
23	Secondary vertical phase of y'–coordinate [pi]
24	Primary vertical phase of y'–coordinate [pi]
25	Coupling angle [pi]
26	D_x [mm]
27	D'_x [mrad]
28	D_y [mm]
29	D'_y [mrad]

When external multipole errors are read–in (see section 3.4.1) the program expects a complete list of magnet errors on file # 16. The format of each set of multipole errors is given in table D.5. The definition of the multipole coefficients should be as described in section 3.3.1.

Table D.5: Format of file with external errors # 16 and internal errors written to # 9

# of Row	**Description**
1	Name of multipole set
2	B_1 B_2 B_3
3	B_4 B_5 B_6
4	B_7 B_8 B_9
5	B_{10} B_{11} B_{12}
6	B_{13} B_{14} B_{15}
7	B_{16} B_{17} B_{18}
8	B_{19} B_{20}
9	A_1 A_2 A_3
10	A_4 A_5 A_6
11	A_7 A_8 A_9
12	A_{10} A_{11} A_{12}
13	A_{13} A_{14} A_{15}
14	A_{16} A_{17} A_{18}
15	A_{19} A_{20}

With the parameter "mout" set to 2 or 3 in the "Random Fluctuation" block (3.4.1) the internally used multipoles are written to file # 9 in the same format as above. This file can therefore be used as an input fort.16 file for a subsequent run.

The file # 34 is written when the "Linear Optic Block" (see section 3.5.1) is invoked with the "ELEMENT 0" option.

Table D.6: Format of file # 34 for detuning and distortion calculation with external program "SODD" [20]

# of Column	**Description**
1	Longitudinal position [m]
2	Type "n" of Multipole ($n > 0$ => erect, $n < 0$ => skew)
3	Multipole strength [$\mathrm{mrad} \cdot \mathrm{mm}^{(1-\lvert n \rvert)}$]
4	Horizontal β–function [m]
5	Vertical β–function [m]
6	Horizontal phase advance
7	Vertical phase advance

The last line serves as the end of the structure: Length of the accelerator, fake name "END", fake type "100", β functions and phase advances at the end of the accelerator for the horizontal and vertical plane respectively.

Appendix E

Tracking Examples

A simple tracking example is shown with its input file (E.1), its output file (E.2) and some corresponding plots in (E.3).

E.1 Input Example

For the description of the different input blocks see chapter 3.

```
FREE FORMAT   TITLE: EXAMPLE
PRINTOUT OF INPUT PARAMETERS-----------------------------------------
NEXT-----------------------------------------------------------------
SINGLE ELEMENTS------------------------------------------------------
B       0  0.0000000      0.000000      50.0000
QD2     2  0.0000000      0.009536      0.77000
QF2     2  0.0000000     -0.009536      0.77000
MU     11  1.0000000      1.000000      0.00000
SEX     3  0.0500000      0.000000      0.00000
NEXT-----------------------------------------------------------------
BLOCK DEFINITIONS----------------------------------------------------
  1  1
 B1   QD2 B QF2
 B2   QF2 B QD2
NEXT-----------------------------------------------------------------
STRUCTURE INPUT------------------------------------------------------
          MU  B1 SEX B2
NEXT-----------------------------------------------------------------
MULTIPOLE COEFFICIENTS-----------------------------------------------
MU          10.0    3.5765
        0.0000     0.0000    0.0000    0.0000
        0.0000     0.0000    0.0000    0.0000
        0.405E-3   0.0000    0.0000    0.0000
        -.5E-5     0.0000    0.0000    0.0000
        -.56E-4    0.0000    0.0000    0.0000
        0.0000     0.0000    0.0000    0.0000
        0.3E-5     0.0000    0.0000    0.0000
        0.0000     0.0000    0.0000    0.0000
        -.1E-5     0.0000    0.0000    0.0000
NEXT-----------------------------------------------------------------
TRACKING PARAMETERS--------------------------------------------------
 10000   0   2    11.0    11.5   0    1
   1       1       0   0
      0         0         1         1         1    50000 2
NEXT-----------------------------------------------------------------
INITIAL COORDINATES--------------------------------------------------
        2   0.       0.        1.
                      0.
                      0.
                      0.
                      0.
                      0.
                      0.
                      0.
                      0.000001
                      0.
                      0.
                      0.
                      0.
                 450000.
                 450000.
                 450000.
NEXT-----------------------------------------------------------------
ITERATION-ACCURACY---------------------------------------------------
       50 1D-14 1D-15
    10 1D-10 1D-10
    10 1D-5 1D-6
    1D-8  1D-12 1D-10
NEXT-----------------------------------------------------------------
POSTPROCESSING-------------------------------------------------------
EXAMPLE
  1000    0 0  1    .08 .08 1
   0.     0.  1  1  20  .005   1 .10
  7878   1   0    1 1 1 1
NEXT
ENDE=================================================================
```

E.2 Output Example

The preprocessing part is shown first. Followed by the initial coordinates and the final coordinates for a regular (right side) and chaotic (left side) case.

```
---------------------------------------------------------------------------------------------------------------------------

        OOOOOOOOOOOOOOOOOOOOO
        OO                 OO
        OO  PREPROCESSING  OO
        OO                 OO
        OOOOOOOOOOOOOOOOOOOOO

---------------------------------------------------------------------------------------------------------------------------

   ---- ENTRY CLORB ----/DPP= 0.00000 /CLOX/   0.00000   0.00000 /CLOZ/   0.00000   0.00000 /ITERAT.=  2/ ACCURACY= 0.000000D+00
   ---- ENTRY CLORB ----/DPP= 0.00000 /CLOX/   0.00000   0.00000 /CLOZ/   0.00000   0.00000 /ITERAT.=  2/ ACCURACY= 0.000000D+00
   ---- ENTRY ORBIT ----/NO MONITORS SPECIFIED
   ---- ENTRY CLORB ----/DPP= 0.00000 /CLOX/   0.00000   0.00000 /CLOZ/   0.00000   0.00000 /ITERAT.=  2/ ACCURACY= 0.000000D+00
   ---- ENTRY CLORB ----/DPP= 0.00000 /CLOX/   0.00000   0.00000 /CLOZ/   0.00000   0.00000 /ITERAT.=  2/ ACCURACY= 0.000000D+00

---------------------------------------------------------------------------------------------------------------------------

        REL. MOMENTUM DEVIATION=   0.00000
      ========================================

        TRACKING FOR CONSTANT MOMENTUM DEVIATION

              ------ NO ACCELERATION ------

                  TUNE           CLO           CLOP           BET0           ALF0
          X    0.1222386779 0.00000000E+00 0.00000000E+00   92.957545511   0.0000000000
          Z    0.1222386779 0.00000000E+00 0.00000000E+00  203.581213058   0.0000000000

    ---- INITIAL COORD. OF TWIN-TRAJECTORIES

          11.50000000000000000000000000000000          11.00000000000000000000000000000000
          -0.00000000000000004942488456867054          -0.00000000000000004727597654394574
          17.01862094259769260000000000000000          16.27868090161518210000000000000000
           0.00000000000000001669894918209964           0.00000000000000001597290791331270
           0.00000000000000000000000000000000           0.00000000000000000000000000000000
           0.00000000000000000000000000000000           0.00000000000000000000000000000000
          11.50000000000000000000000000000000          11.00000000000000000000000000000000
           0.00000099999999999505730600000000           0.00000099999999999527224000000000
          17.01862094259769260000000000000000          16.27868090161518210000000000000000
           0.00000000000000001669894918209964           0.00000000000000001597290791331270
           0.00000000000000000000000000000000           0.00000000000000000000000000000000
           0.00000000000000000000000000000000           0.00000000000000000000000000000000
      450000.00000000000000000000000000000000      450000.00000000000000000000000000000000
      449999.99999999999710000000000000000000      449999.99999999999710000000000000000000
      449999.99999999999710000000000000000000      449999.99999999999710000000000000000000

---------------------------------------------------------------------------------------------------------------------------

        OOOOOOOOOOOOOOOO
        OO            OO
        OO  TRACKING  OO
        OO            OO
        OOOOOOOOOOOOOOOO

---------------------------------------------------------------------------------------------------------------------------

    NUMBER OF REVOLUTION       10000

          -2.48768615571860296000000000000000           6.41350525115668169000000000000000
           0.07575609332820337770000000000000          -0.08929290763721121900000000000000
         -15.96950708456677390000000000000000          -6.60135941908570345000000000000000
           0.08126481209989279420000000000000           0.08135693924485626340000000000000
           0.00000000000000000000000000000000           0.00000000000000000000000000000000
           0.00000000000000000000000000000000           0.00000000000000000000000000000000
          10.71699211682154410000000000000000           6.40774254543778388000000000000000
          -0.07967757286602571150000000000000          -0.08930614360732422320000000000000
           5.97384177190877108000000000000000          -6.60868117515426112000000000000000
          -0.07540770872522122030000000000000           0.08135396639582599680000000000000
           0.00000000000000000000000000000000           0.00000000000000000000000000000000
           0.00000000000000000000000000000000           0.00000000000000000000000000000000
      450000.00000000000000000000000000000000      450000.00000000000000000000000000000000
      449999.99999999999710000000000000000000      449999.99999999999710000000000000000000
      449999.99999999999710000000000000000000      449999.99999999999710000000000000000000

        ******************************
        **** ALL PARTICLES STABLE ****
        ******************************
```

Finally part of the post–processing for the two particles are shown (chaotic on the left and regular on the right respectively) and a summary of the post–processing is given.

```
---------------------------------------------------------------------------------------------------------------------------

         OOOOOOOOOOOOOOOOOOOOOO
         OO                  OO
         OO  POSTPROCESSING  OO
         OO                  OO
         OOOOOOOOOOOOOOOOOOOOOO

---------------------------------------------------------------------------------------------------------------------------

         ANALYSING THE INCREASE OF THE DISTANCE IN PHASE-SPACE
         -----------------------------------------------------

TURNS        DISTANCE            SLOPE         RESIDUAL       TURNS        DISTANCE             SLOPE          RESIDUAL
-----------------------------------------------------------------------------------------------------------------------
 2000     0.2253779764D-03    0.4554898739    0.0000000000     2000     0.9092427661D-04    0.3871969581    0.0000000000
 3000     0.1081182799D-02    1.3754730225    0.3947801590     3000     0.2666317995D-03    0.9350422025    0.1414830685
 4000     0.6399160375D-02    3.0341444016    2.1449279785     4000     0.3394974411D-03    0.9019818306    0.0706214905
 5000     0.4981834333D-02    2.2218360901    2.3199357986     5000     0.5020486718D-03    1.0834455490    0.0927543640
 6000     0.8028940085D-01    4.1418428421   10.5150909424     6000     0.5503330639D-03    0.9862068892    0.0943765640
 7000     0.3407768847D+00    6.2231464386   25.0305175781     7000     0.7196859601D-03    1.0560836792    0.0917263031
 8000     0.4788764947D+00    6.7520313263   22.5305938721     8000     0.7402482154D-03    0.9828781486    0.1097971201
 9000     0.4507363285D+00    6.2743434906   21.2371520996     9000     0.9506629146D-03    1.0507984161    0.1310729980
10000     0.6438836450D+00    5.8023786545   21.1673889160    10000     0.9737567472D-03    1.0118942261    0.1297397614
------------------------------------------------------------------------------------------------------------------------


     AVERAGED PHASE-ADVANCE
     ----------------------

X-PHASE :    0.1175695955   +/_   0.0003315228                X-PHASE :    0.1189145300   +/_   0.0001260177
Z-PHASE :    0.1196627812   +/_   0.0001993022                Z-PHASE :    0.1211689637   +/_   0.0001953892

S-PHASE :    0.0000000000   +/_   0.0000000000                S-PHASE :    0.0000000000   +/_   0.0000000000

TART-QX :   0.1222386779   CHANGE IN X : -.4669082376D-02      START-QX :   0.1222386779   CHANGE IN X : -.3324147915D-02
TART-QZ :   0.1222386779   CHANGE IN Z : -.2575896685D-02      START-QZ :   0.1222386779   CHANGE IN Z : -.1069714140D-02

     THE AVERAGED PHASE-ADVANCES ARE CLOSER THEN 0.5000D-02 TO THE FOLLOWING RESONANCES UP TO  20 ORDER
     --------------------------------------------------------------------------------------------------

     NX * QX   +   NZ * QZ   -      P      =      DELTA          NX * QX   +   NZ * QZ   -      P      =      DELTA
     ----------------------------------------------------          ----------------------------------------------------
        1           -1            0.0         -.2093D-02             1           -1            0.0         -.2254D-02
        2           -2            0.0         -.4186D-02             2           -2            0.0         -.4509D-02
       14            3            2.0         0.4963D-02
       15            2            2.0         0.2869D-02
       16            1            2.0         0.7763D-03
       17            0            2.0         -.1317D-02
       18           -1            2.0         -.3410D-02




---------------------------------------------------------------------------------------------------------------------------
        SUMMARY OF THE POSTPROCESSING

---------------------------------------------------------------------------------------------------------------------------
       |           |           |            |          |NORMALIZED| SLOPE  |             |         |                     |
 TURN  |   LINEAR  |   BETA-   | AMPLITUDES | MOMENTUM |PHASESPACE| OF THE |  NONLINEAR  | NEAREST |      SMEAR OF       |
NUMBER |   TUNES   | FUNCTIONS |            | DEVIATION| DISTANCE |DISTANCE|  DETUNING   | RESONANCE|   THE EMITTANCES    |
---------------------------------------------------------------------------------------------------------------------------
       |           |    [M]    |    [MM]    |          |          |        |             |     |ORD.|   [%]   |      [%]  |
---------------------------------------------------------------------------------------------------------------------------
  10000|X   0.12224|X   92.9575|X  11.000000|0.0000D+00|0.9738D-03|  1.0119|X -.33241D-02|X   1|  2 |X  26.423|X+Z   5.664|
       |Z   0.12224|Z  203.5812|Z  16.278681|          |          |        |+/- 0.126D-03|Z  -1|    |Z  34.450|           |
       |QS 0.000000|           |            |          |          |        |Z -.10697D-02|     |    |         |           |
       |           |           |            |          |          |        |+/- 0.195D-03|     |    |         |           |
---------------------------------------------------------------------------------------------------------------------------
  10000|X   0.12224|X   92.9575|X  11.500000|0.0000D+00|0.6439D+00|  5.8024|X -.46691D-02|X  16| 17 |X  45.228|X+Z   6.683|
       |Z   0.12224|Z  203.5812|Z  17.018621|          |          |        |+/- 0.332D-03|Z   1|    |Z  33.745|           |
       |QS 0.000000|           |            |          |          |        |Z -.25759D-02|     |    |         |           |
       |           |           |            |          |          |        |+/- 0.199D-03|     |    |         |           |
```

E.3 Plot Example

In figure E.1 a typical example of the evolution of the distance in phase space is shown of a regular and chaotic particle. Figure E.2 and figure E.3 show the corresponding horizontal phase space and the physical phase space projections respectively. An example of the stroboscoped phase space is shown in figure E.4, where the motion in the chaotic case is beyond a “separatrix” in the four–dimensional phase space. Even in the FFT (figure E.5) one can see the effect of chaotic behaviour: it leads to a widening of the lines of the spectrum.

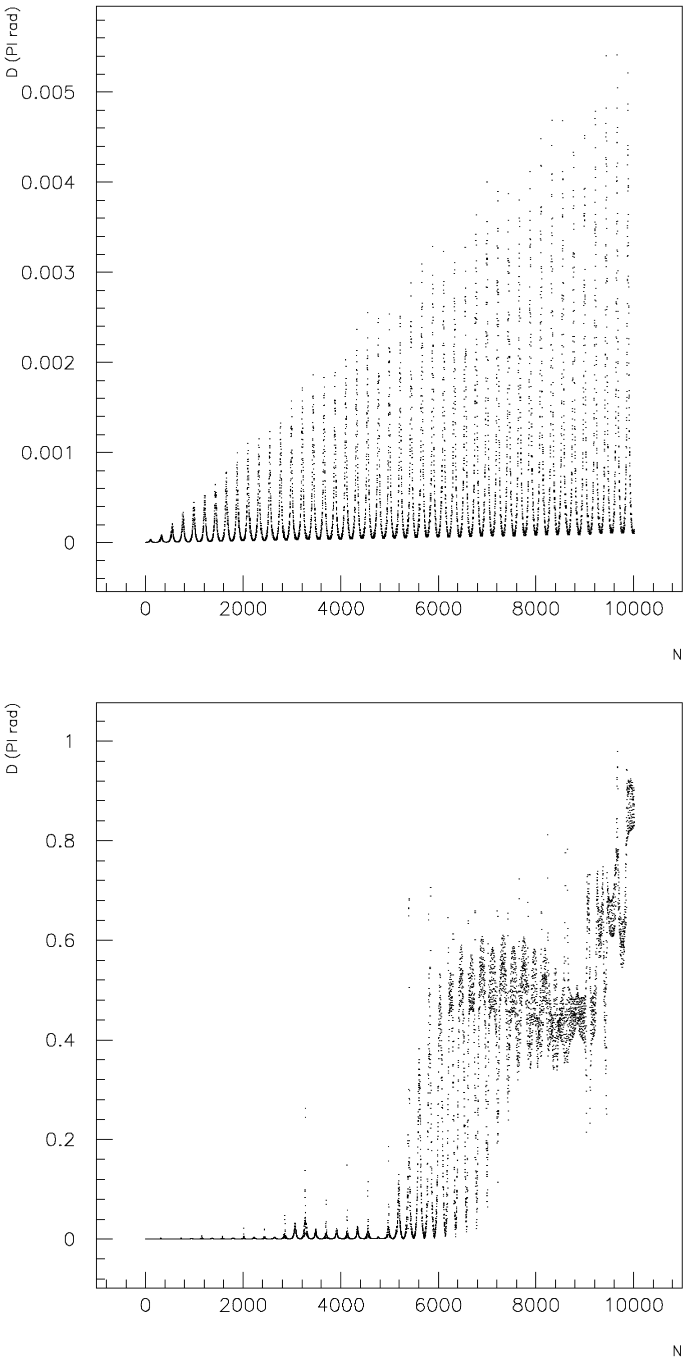

Figure E.1: Evolution of the Distance of Phase Space for Regular (upper part) and Chaotic (lower part) Motion.

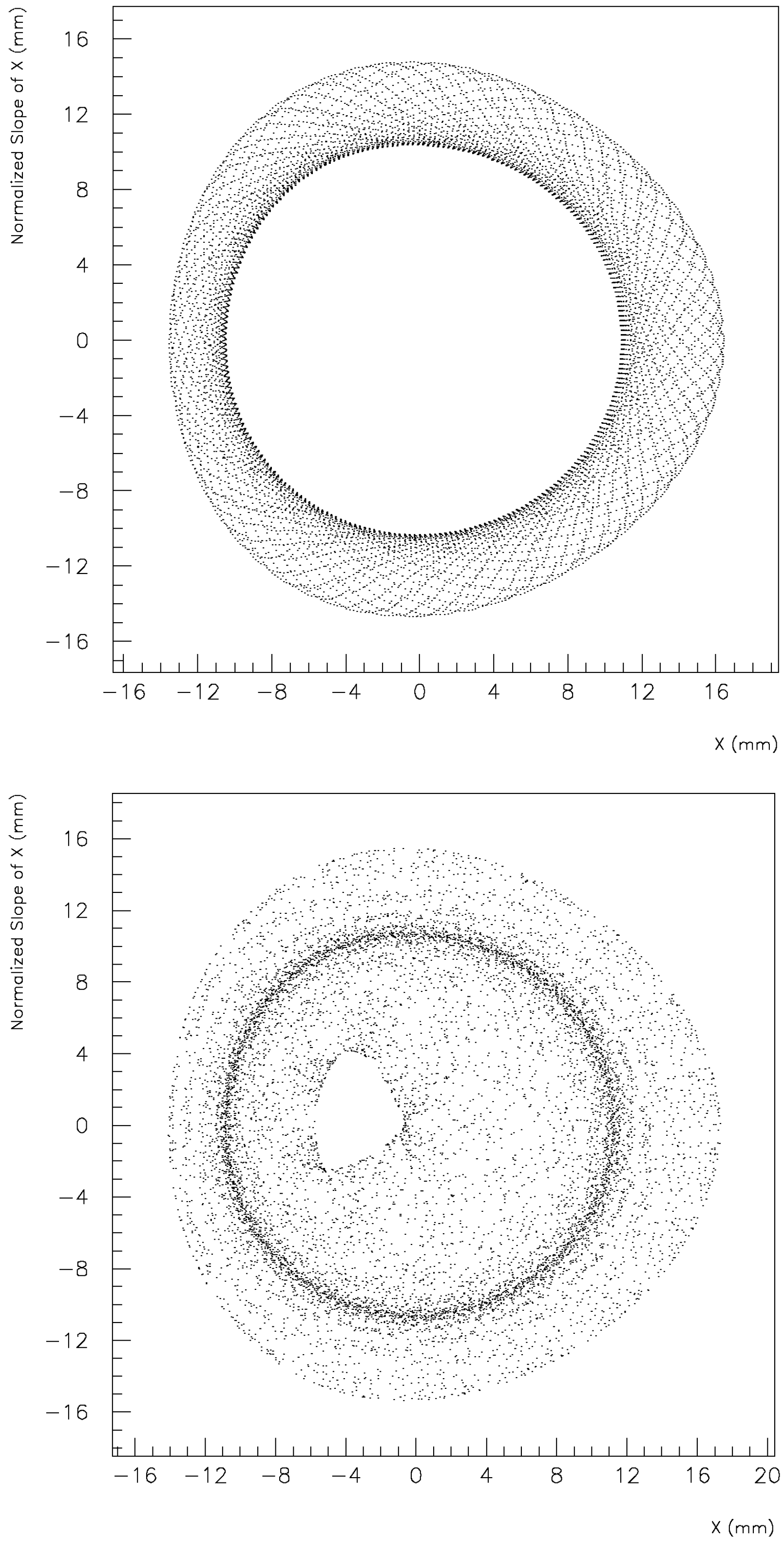

Figure E.2: Horizontal Phase Space Projections for the Regular (upper part) and the Chaotic (lower part) Cases.

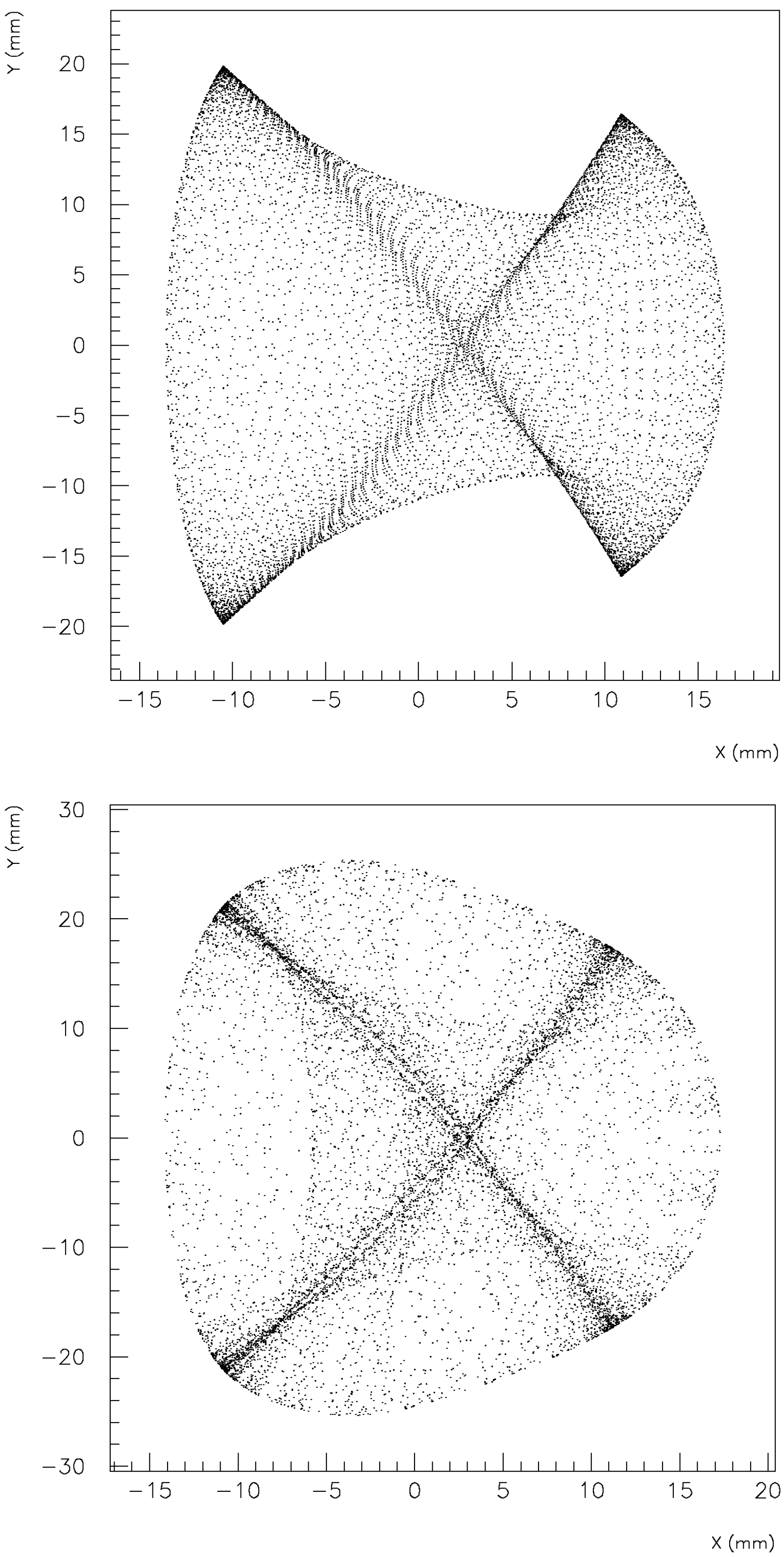

Figure E.3: Physical Phase Space Projections for the Regular (upper part) and the Chaotic (lower part) Cases.

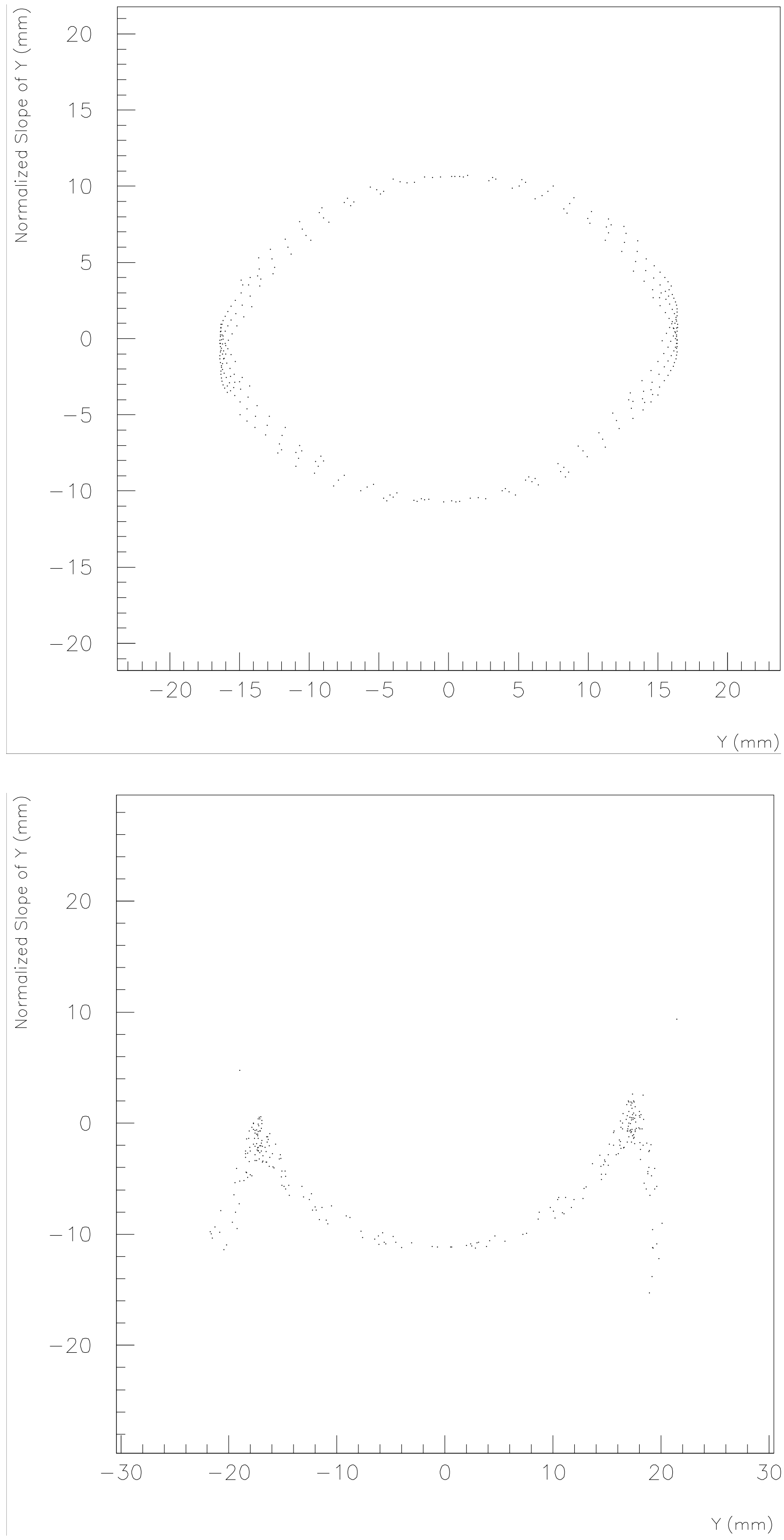

Figure E.4: Stroboscoped Vertical Phase Space Projections for the Regular (upper part) and the Chaotic (lower part) Cases respectively. The regular motion stays inside a “separatrix” with two unstable fix–points visible, while the chaotic motion is clearly outside this “separatrix”.

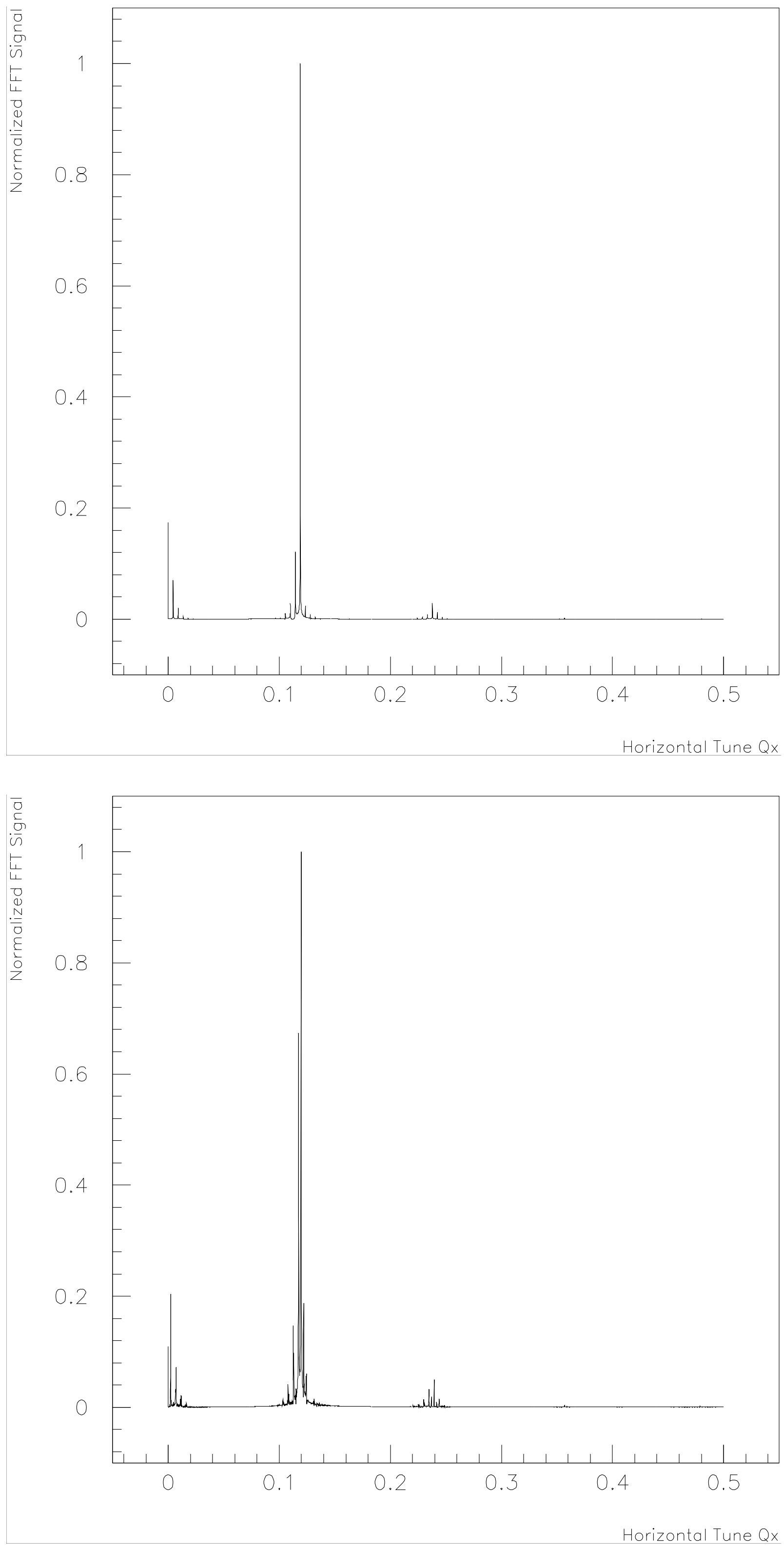

Figure E.5: Horizontal FFT–Analysis for the Regular (upper part) and the Chaotic (lower part) Cases.

Bibliography

[1] LBL diffential algebra package and LieLib routines courtesy of É. Forest.

[2] G. Ripken and F. Schmidt, "A symplectic six–dimensional thin–lens formalism for tracking", CERN SL 95–12 (AP)(1995), DESY 95–063 (1995). G. Ripken and F. Schmidt, "Construction of Nonlinear Symplectic Six–Dimensional Thin–Lens Maps by Exponentiation", DESY 95–189 (1995), http://cern.ch/Frank.Schmidt/report/ripken2.pdf; D.P. Barber, K. Heinemann, G. Ripken and F. Schmidt, "Symplectic Thin - Lens Transfer Maps for SixTrack: Treatment of Bending Magnets in Terms of the Exact Hamiltonian", DESY 96–156 (1995), http://cern.ch/Frank.Schmidt/report/ripken3.pdf.

[3] A. Wrulich, "RACETRACK, A computer code for the simulation of nonlinear motion in accelerators", DESY 84–026 (1984).

[4] B. Leemann and É. Forest, "Brief description of the tracking codes FASTRAC and THINTRAC", SSC Note SSC–133.

[5] G. Ripken, "Nonlinear canonical equations of coupled synchro–betatron motion and their solution within the framework of a nonlinear 6–dimensional (symplectic) tracking program for ultra–relativistic protons", DESY 85–084 (1985).

[6] D.P. Barber, G. Ripken and F. Schmidt, "A nonlinear canonical formalism for the coupled synchro–betatron motion of protons with arbitrary energy", DESY 87–036 (1987); G. Ripken and F. Schmidt, "A symplectic six–dimensional thin–lens formalism for tracking", CERN/SL/95–12 (AP), DESY 95–063 (1995), http://cern.ch/Frank.Schmidt/report/ripken.pdf; K. Heinemann,

[7] R. Brun and D. Lienart, "HBOOK User Guide", CERN Program Library Y250 (1987).

[8] R. Brun and N.C. Somon, "HPLOT User Guide", CERN Program Library Y251 (1988).

[9] R. Bock, R. Brun, O. Couet, N.C. Somon, C.E. Vandoni and P. Zanarini, "HIGZ User Guide", CERN Program Library Q120.

[10] G. Guignard, "A general treatment of resonances in accelerators", CERN 78–11 (1978).

[11] M. Berz, "Differential algebra description of beam dynamics to very high orders", Particle Accelerators, 1989, Vol. 24, pp. 109–124.

[12] M. Berz, "DAFOR – Differential Algebra Precompiler Version 3, Reference Manual", MSUCL–755 (1991).

[13] F. Schmidt and M. Vaenttinen, "Vectorisation of the single particle tracking program SixTrack", CERN SL Note 90–20 (1990) (AP).

[14] F. Schmidt, "Untersuchungen zur dynamischen Akzeptanz von Protonenbeschleunigern und ihre Begrenzung durch chaotische Bewegung", DESY HERA 88–02, (1988).

[15] H. Grote, "A MAD–SixTrack interface", SL Note 97–02 (AP).

[16] M. Berz, É. Forest and J. Irwin, "Normal form methods for complicated periodic systems: a complete solution using differential algebra and lie operators", Particle Accelerators, 1989, Vol. 24, pp. 91–107.

[17] M. Bassetti and G.A. Erskine, "Closed expression for the electrical field of a two–dimensional Gaussian charge", CERN–ISR–TH/80–06.

[18] K. Hirata, H. Moshammer, F. Ruggiero and M. Bassetti, "Synchro-Beam interaction", CERN SL-AP/90-02 (1990) and Proc. Workshop on Beam Dynamics Issues of High-Luminosity Asymmetric Collider Rings, Berkeley, 1990, ed. A.M. Sessler (AIP Conf. Proc. 214, New York, 1990), pp. 389-404;
K. Hirata, H. Moshammer and F. Ruggiero, "A symplectic beam-beam interaction with energy change", KEK preprint 92-117 A (1992) and Part. Accel. 40, 205-228 (1993);
K. Hirata, "BBC User's Guide; A Computer Code for Beam-Beam Interaction with a Crossing Angle, version 3.4", SL-Note 97-57 AP.

[19] L.H.A. Leunissen, F. Schmidt and G. Ripken, "6D Beam–Beam Kick including Coupled Motion", LHC Project Report 369, http://cern.ch/Frank.Schmidt/report/ripken_new.pdf.

[20] F. Schmidt, "SODD:
A Computer Code to calculate Detuning and Distortion Function Terms in First and Second Order", CERN SL/Note 99–009 (AP), http://cern.ch/Frank.Schmidt/report/sodd_manual.pdf.

[21] H. Grote and F.C. Iselin, "The MAD program (Methodical Accelerator Design), Version 8.10, User's Reference Manual", CERN SL 90–13 (AP) (Rev. 4)
http://cern.ch/Hans.Grote/mad/mad8/doc/mad8_user.ps.gz.

[22] private communication.

[23] F. James, "A review of pseudo–random number generators", to be published in Computer Physics Communication.

[24] B. Autin and Y. Marti, "Closed Orbit Correction of A.G. Machines Using a Small Number of Magnets", CERN–ISR–MA/73–17.

[25] M. Giovannozzi, "Description of software tools to perform tune–shift correction using normal forms", CERN SL Note 93–111 (AP).

[26] F. Schmidt, F. Willeke and F. Zimmermann, "Comparison of methods to determine long–term stability in proton storage rings", 1991, Particle Accelerators, Vol. 35, pp. 249–256.

[27] K. Sjobak, H. Burkhardt, R.D. Maria, A. Mereghetti and A. Santamaria, "General functionality for turn-dependent element properties in SixTrack" 2015, Procedings of IPAC'13, Richmond, VA, USA, May 2015.

[28] S. Russenschuck, "Field computation for Accelerator Magnets", Wiley-VCH, 2010

[29] P. Burla, Q. King and J.G. Pett, "Optimisation of the current ramp for the LHC", Proceedings of the 1999 Particle Accelerator Conference, New York, 1999.