The LSA Optics Model

D. Ondreka

Version 0.2
March 30, 2017

This document describes the optics model used in the FAIR settings management system based on LSA. It includes a description of the data on transverse optics stored in the LSA database as well as the usage of this data for the calculation of settings in the machine model and in beam based applications relying on optics data such as orbit and trajectory control, optics correction, and others.

1. Optics Model Used in Settings Management

1.1. Definition of optic by strengths

A particular optic λ is characterized by a set of strengths for all elements of the lattice. More precisely, the presently employed generic machine model assumes that those elements are magnets described by the integral strength $k_{n_e}\ell_e^\lambda$ of a single component n_e in the circular multipole expansion of the magnetic field of the element e.[1]

The values $k_{n_e}\ell_e^\lambda$ are referred to as *theory strengths* of the optics λ. Once the theory strengths for all elements e are known, all properties of the optics of the ring are uniquely determined.

The optics model employed in the settings management requires the theory strengths for the calculation of settings. The strengths of lattice dipoles and quadrupoles are mandatory, whereas the strengths of other elements, like sextupoles for chromaticity correction, are optional and will be assumed to be zero if not present. In practice, for most rings the optics will usually be defined by just specifying the strengths of dipoles and quadrupoles, leaving the calculation of strengths for other elements to the machine model or an application.

The strength values for any optic defined in the LSA database are stored in the table `OPTIC_STRENGTHS`. Assuming a ring with a single power converter for the lattice dipoles, represented by the LSA device `<D>`, and f quadrupole families with devices `<Q#c>` representing the power converter of the quadrupole family with index c, Table 1 displays the connection between the LSA device and the corresponding strength value.

In general, a property P of the linear optics is a function of the strengths of all lattice quadrupoles. For a particular optic λ, the value P^λ is therefore defined as:

$$P^\lambda \equiv P\left(k_{n_e}\ell_e^\lambda\right) \tag{1}$$

[1] Note that skew multipole components are rarely used for the lattice magnets of a ring.

Table 1: Mandatory integral strength values stored in the table OPTIC_STRENGTHS.

Integral strength	LSA Device	Symbol
dipole angle	<D>	$k_0\ell_1^\lambda$
strength of quadrupole family 1	<Q#1>	$k_1\ell_1^\lambda$
$\vdots$	$\vdots$	$\vdots$
strength of quadrupole family f	<Q#f>	$k_1\ell_f^\lambda$

Table 2: Mandatory global properties stored in the table OPTIC_PARAMETERS.

Property	LSA Parameter	Symbol
horizontal tune	<BEAM>/QH_THEO	Q_h^λ
vertical tune	<BEAM>/QV_THEO	Q_v^λ
horizontal chromaticity	<BEAM>/CH_THEO	${Q'}_h^\lambda$
vertical chromaticity	<BEAM>/CV_THEO	${Q'}_v^\lambda$
momentum compaction factor	<BEAM>/ALPHAC	α_c^λ
RMS dispersion at BPMs	<BEAM>/ALPHADR	$\alpha_{\delta R}^\lambda$

The quantities P^λ are referred to as *theory properties* of the optics λ. Examples of such properties are the transverse tunes and chromaticities.

The optics model employed in the settings management requires a number of global properties of the linear optics for the calculation of settings. These properties are represented in the machine model as LSA parameters. Their values for any optic defined in the LSA database is stored in the table OPTIC_PARAMETERS. Table 2 lists all global properties used in the machine model.

1.2. Tunes and tune changes

The transverse tunes Q_h and Q_v are global properties of the optic (see 1). In general, then, they depend on the theory strengths of all elements. However, for the purpose of applying changes to the tunes, the model employed in the settings management considers the tunes to be a function of the strengths $k_1\ell_c$ of the f lattice quadrupoles only, with all other strengths held fixed:

$$Q_{h,v} = Q_{h,v}\left(k_1\ell_1, \ldots, k_1\ell_f\right) \tag{2}$$

The theory tunes $Q_{h,v}^\lambda$ are given by evaluating this function for the theory strengths $k_1\ell_c^\lambda$:

$$Q_{h,v}^\lambda \equiv Q_{h,v}\left(k_1\ell_1^\lambda, \ldots, k_1\ell_f^\lambda\right) \tag{3}$$

If we want to change the tunes from their theory values by an increment $\delta Q_{h,v}$, we have to change the quadrupole strengths. In the ideal machine, this change is described by a set of quadrupole strength changes $\delta k_1\ell_c$ satisfying:

$$Q^\lambda_{h,v} + \delta Q_{h,v} = Q_{h,v}\left(k_1\ell_1^\lambda + \delta k_1\ell_1, \dots, k_1\ell_f^\lambda + \delta k_1\ell_f\right) \tag{4}$$

The relation between $\delta Q_{h,v}$ and $\delta k_1\ell_c$ is non-linear and in general an ion optical simulation tool is required to calculate the strength changes. However, if the tune changes are small enough, a linear approximation can be used:

$$Q^\lambda_{h,v}+\delta Q_{h,v} = Q_{h,v}\left(k_1\ell_1^\lambda, \dots, k_1\ell_f^\lambda\right)+\sum_{c=1}^{f}\frac{\partial Q_{h,v}}{\partial k_1\ell_c}\left(k_1\ell_1^\lambda, \dots, k_1\ell_f^\lambda\right)\delta k_1\ell_c+O\left(\delta k_1\ell_c^2\right) \tag{5}$$

We now define the *tune response matrix* T^λ by the partial derivatives:

$$T^\lambda_{kc} \equiv \frac{\partial Q_k}{\partial k_1\ell_c}\left(k_1\ell_1^\lambda, \dots, k_1\ell_f^\lambda\right) \quad k = h, v \tag{6}$$

Note that this matrix depends on the optic λ, as indicated.

The calculation of strength changes from tune changes requires the inversion of this relation. Since in general there may be more than two families of lattice quadrupoles, we describe this inversion by a pseudo-inverse $\overline{T}^\lambda$ of the tune response. Given this pseudo-inverse, the quadrupole strength changes can be calculated as follows:

$$\delta k_1\ell_c = \sum_{k=h,v}\left(\overline{T}^\lambda\right)_{ck}\delta Q_k \tag{7}$$

In the simplest case, the number of independent quadrupole families is just $f = 2$. Then, the pseudo-inverse is equal to the true inverse of the tune response matrix, i.e. $\overline{T}^\lambda = (T^\lambda)^{-1}$. When $f > 2$, additional constraints may be used to define $\overline{T}^\lambda$ uniquely. For instance, in the SIS100 proton optics with shifted γ_t, an additional constraint is set to ensure that γ_t remains unchanged when changing the tune. Of course, it is also possible to define the pseudo-inverse by using a singular value decomposition, with or without additional constraints.

In practice, the pseudo-inverse $\overline{T}^\lambda$ is frequently calculated directly by using an ion optical simulation tool to calculate the quadrupole strengths for a small tune change $\overline{\delta Q}$ in either h or v and then approximating the partial derivatives by the difference quotients:

$$\overline{T}^\lambda_{ck} = \frac{\overline{k_1\ell_c} - k_1\ell_c^\lambda}{\overline{\delta Q}} \tag{8}$$

Here, $\overline{k_1\ell_c}$ is the solution, determined using an ion optical simulation program, of two or more implicit equations. For instance, if $k = h$:

$$Q^\lambda_h + \overline{\delta Q} = Q_h\left(\overline{k_1\ell_1}, \dots, \overline{k_1\ell_f}\right) \tag{9}$$
$$Q^\lambda_v = Q_v\left(\overline{k_1\ell_1}, \dots, \overline{k_1\ell_f}\right) \tag{10}$$
$$0 = C_i\left(\overline{k_1\ell_1}, \dots, \overline{k_1\ell_f}\right) \tag{11}$$

The C_i summarize any additional constraints satisfied by the solution $\overline{k_1\ell_c}$, which are taken into account in the calculation using the ion optical simulation tool. Note that,

Table 3: Mandatory knob values required for the calculation of quadrupole strength changes from tune changes, stored in the table `KNOBS`.

Knob	Knob symbol	Component	Comp. symbol	Value
`<BEAM>/DQH`	δQ_h	`<Q#1>/DKL`	$\delta k_1\ell_1$	$\overline{T}^{\lambda}_{1h}$
⋮	⋮	⋮	⋮	⋮
`<BEAM>/DQH`	δQ_h	`<Q#f>/DKL`	$\delta k_1\ell_f$	$\overline{T}^{\lambda}_{fh}$
`<BEAM>/DQV`	δQ_v	`<Q#1>/DKL`	$\delta k_1\ell_1$	$\overline{T}^{\lambda}_{1v}$
⋮	⋮	⋮	⋮	⋮
`<BEAM>/DQV`	δQ_v	`<Q#f>/DKL`	$\delta k_1\ell_f$	$\overline{T}^{\lambda}_{fv}$

if such constraints are applied, it is in general necessary that these constraints be also satisfied by the theory values, i.e. we must have:

$$0 = C_i\left(k_1\ell_1^{\lambda}, \ldots, k_1\ell_f^{\lambda}\right) \tag{12}$$

Otherwise, the theory optics and the changed optics will differ in their behaviour with respect to the constraints.

The optics model employed in the settings management represents the tune changes by two parameters of type `DTUNE`, connected to the beam device `<BEAM>` of the particular ring particle transfer and called `<BEAM>/DQH` and `<BEAM>/DQV`. The changes in the quadrupole strengths are represented by f parameters of type `DKL`, labeled `<Q#c>/DKL`, where `<Q#c>` denotes the device representing the power converter of the lattice quadrupole family with component index c.

The values of the pseudo-inverse response matrix $\overline{T}^{\lambda}_{ck}$ are stored in the table `KNOBS`, with the tune change parameters used as knobs and the strength change parameters used as components. Table 3 displays the connection of the symbols used in the mathematical description with the corresponding entries in the table `KNOBS`.

1.3. Chromaticities and chromaticity changes

The natural chromaticities Q'_h and Q'_v of a circular accelerator are global properties of the optic (see 1). In general, they will depend on the strengths of quadrupoles and sextupoles, where the latter may also include systematic sextupole components of bending magnets.

However, changes to the chromaticities will always be established by adjusting the strengths $k_2\ell_c$ of s families of sextupoles for chromaticity correction. The relation between chromaticities and strengths is typically linear. Therefore, we define the *chromaticity response matrix* C^{λ} as follows:

$$\delta Q'_k \equiv \sum_{c=1}^{s} C^{\lambda}_{kc} k_2\ell_c \quad k = h, v \tag{13}$$

Note that this matrix depends on the optic λ, as indicated.

The calculation of strength changes from chromaticity changes requires the inversion of this relation. Since in general there may be more than two families of sextupoles for chromaticity correction, we describe this inversion by a pseudo-inverse $\overline{C}^{\lambda}$ of the chromaticity response. Given this pseudo-inverse, the sextupole strength changes can be calculated as follows:

$$\delta k_2 \ell_c = \sum_{k=h,v} \left(\overline{C}^{\lambda}\right)_{ck} \delta Q'_k \tag{14}$$

Like for the tunes, it is, of course, possible to calculate the chromaticity response or its pseudo-inverse using an ion optical simulation tool. However, the textbook formulae are usually sufficient:

$$C^{\lambda}_{hc} = \frac{1}{4\pi} \sum_{n=1}^{s_c} k_2 \ell_n \, \beta^{\lambda}_{x,n} D^{\lambda}_{x,n} \tag{15}$$

$$C^{\lambda}_{vc} = -\frac{1}{4\pi} \sum_{n=1}^{s_c} k_2 \ell_n \, \beta^{\lambda}_{y,n} D^{\lambda}_{x,n} \tag{16}$$

Note that the sums run over the s_c sextupoles pertaining to the family with index c. The beta functions β_x, β_y, and the horizontal dispersion D_x are to be evaluated at the center of the sextupole with index n. The superscript λ indicates that the values are taken from the Twiss parameters calculated with the theory strengths of the optic λ.

In the simplest case, the number of independent sextupole families is just $s = 2$. Then, the pseudo-inverse is equal to the true inverse of the chromaticity response matrix, i.e. $\overline{C}^{\lambda} = (C^{\lambda})^{-1}$. When $s > 2$, additional constraints may be used to define $\overline{C}^{\lambda}$ uniquely. For instance, in the SIS100 proton optics with shifted γ_t, the strengths of the sextupoles are calculated by minimizing the term $\sum_c (\beta_{x,c} k_2 \ell_c)^2$ in order to reduce the effect of the non-linearities introduced by the sextupoles. The chromaticities are specified as constraints for the minimization. Of course, it is also possible to define the pseudo-inverse by using a singular value decomposition, with or without additional constraints.

The optics model employed in the settings management represents the chromaticity changes by two parameters of type DCHROMATICITY, connected to the beam device <BEAM> of the particular ring particle transfer and called <BEAM>/DCH and <BEAM>/DCV. The changes in the sextupole strengths are represented by s parameters of type DKL, labeled <CS#c>/DKL, where <CS#c> denotes the device representing the power converter of the sextupole family with component index c.

The values of the pseudo-inverse response matrix $\overline{C}^{\lambda}_{ck}$ are stored in the table KNOBS, with the chromaticity change parameters used as knobs and the strength change parameters used as components. Table 4 displays the connection of the symbols used in the mathematical description with the corresponding entries in the table KNOBS.

1.4. Momentum compaction factor

The momentum compaction factor α_c is a global property of the optic (see 1). In the machine models for FAIR, this quantity is needed in particular to calculate the phase

Table 4: Mandatory knob values required for the calculation of sextupole strength changes from chromaticity changes, stored in the table KNOBS.

Knob	Knob symbol	Component	Comp. symbol	Value
<BEAM>/DCH	$\delta Q'_h$	<CS#1>/DKL	$\delta k_2 \ell_1$	$\overline{C}^{\lambda}_{1h}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
<BEAM>/DCH	$\delta Q'_h$	<CS#s>/DKL	$\delta k_2 \ell_s$	$\overline{C}^{\lambda}_{sh}$
<BEAM>/DCV	$\delta Q'_v$	<CS#1>/DKL	$\delta k_2 \ell_1$	$\overline{C}^{\lambda}_{1v}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
<BEAM>/DCV	$\delta Q'_v$	<CS#s>/DKL	$\delta k_2 \ell_s$	$\overline{C}^{\lambda}_{sv}$

slip factor η to be used extensively in calculations for the RF systems. Other applications are the calculation of orbit length changes caused by changes in the momentum of the beam or the field in the bending magnets.

In the machine model, we use only the linear term of the general expansion in powers of the momentum deviation δ, i.e. we define α_c by:

$$\frac{\Delta L}{L} \equiv \alpha_c \cdot \delta + \mathrm{O}\left(\delta^2\right) \tag{17}$$

Hence, it can be calculated according to the usual textbook formula:

$$\alpha_c^{\lambda} = \oint \mathrm{d}s \frac{D^{\lambda}(s)}{\rho(s)} \tag{18}$$

In practice, the momentum compaction factor α_c^{λ} for a particular optic λ will in most cases be obtained from an ion optics simulation.

1.5. Radial displacements

In circular accelerators the beam is frequently displaced deliberately from the design orbit by adjusting the magnetic field of the main dipoles and the RF frequency in such a way as to keep the energy of the beam constant. In the machine models of GSI for SIS18 and ESR, such changes have been described by a change in average radius $\Delta R/R = \Delta L/L$. Consequently, the calculations of field and frequency offsets involved the momentum compaction factor α_c.

This strategy has two important drawbacks: Firstly, the algorithm requires a division by α_c and thus fails if α_c approaches zero. Secondly, the value ΔR used to parametrize the change has no direct relation to the position of the beam at the beam posistion monitors.

To overcome these limitations for FAIR, we define a new quantity $\alpha_{\delta R}$ as follows:

$$\sqrt{\langle x^2 \rangle}\Big|_{\mathrm{BPM}} \equiv \left(\frac{1}{N_{\mathrm{BPM}}} \sum_{n=1}^{N_{\mathrm{BPM}}} x_n^2 \right)^{\frac{1}{2}} \equiv \alpha_{\delta R} \cdot \delta \tag{19}$$

Thus, $\alpha_{\delta R}$ relates momentum deviation δ to the RMS beam position at the N_{BPM} beam position monitors. In the simplest case of a fully symmetric lattice with identical dispersion at each BPM, the RMS position then equals, up to a sign, the position observed at each BPM. In the more general case, the RMS position is still a standard observable easily accessible to the operator of the machine.

From the definition 19 it is clear that, for a particular optic λ, $\alpha_{\delta R}^{\lambda}$ is given by:

$$\alpha_{\delta R}^{\lambda} = \sqrt{\langle D^2 \rangle}\Big|_{\mathrm{BPM}}^{\lambda} = \left(\frac{1}{N_{\mathrm{BPM}}} \sum_{n=1}^{N_{\mathrm{BPM}}} D_n^{\lambda\,2} \right)^{\frac{1}{2}} \tag{20}$$

Table 5: Tables for the storage of optics dependent data in the LSA database.

TABLE	CONTENT
OPTICS	definition of optics including name and description
OPTIC_STRENGTHS	optic dependent values for LSA devices
OPTIC_PARAMETERS	optic dependent values for LSA parameters
KNOBS	optic dependent values for pairs of LSA parameters
TWISS_OUTPUTS	optic dependent Twiss parameters for LSA elements

Table 6: Tables providing keys for the storage of optics dependent data in the LSA database.

TABLE	CONTENT
DEVICES	representation of magnet power converters
ELEMENTS	representation of components in the beamline
ELEMENTS_LOGICAL_HARDWARE	$n \to 1$ mapping between elements and devices
PARAMETERS	representation of properties of devices
PARTICLE_TRANSFERS	representation of beamline sections
OPTICS_PART_TRANS	association of optics with particle transfers

2. Optics Data in the LSA Database

The information on optics is distributed over several tables in the LSA database. For the purpose of this document, these tables can be grouped into two categories:

1. Database tables providing storage space for optic dependent values. These are listed in table 5. They always use the OPTIC_ID from the table OPTICS to identify the particular optic as a primary key. Other keys are employed to associate optics values with elements, devices, or parameters.

2. Database tables providing the primary keys for identifying elements, devices, parameters, optics, and others. These are listed in table 6.

The table OPTICS_PART_TRANS from the second group is a little special in that it does not actually specify foreign keys, but rather provides the association of optics with particle transfers. This is very important for the definition of optics for transfer lines, because in the modeling scheme adopted for FAIR those transfer lines consist in general of several particle transfers. Since optic names must be unique in the table OPTICS, this table cannot be used to establish this association.

However, the mapping from particle transfers to transfer lines is available from the LSA framework. Therefore, the table OPTICS_PART_TRANS will be filled automatically by the import scripts based on the entry in the column TRANSFER_LINE of the corresponding import file. (See section 3.1.1.)

The following subsections describe the structure of the database tables used for the storage of optics dependent data in detail.

Note, however, that we do not list database columns in the order in which they are defined in the LSA database. Instead, we sort the columns according to an logical order adapted to the usage for FAIR. In particular, columns which need not be provided by the machine modeler or which are not actually used for FAIR will be mentioned last.

Note also that we do not display the actual Oracle data type to describe the value type of a certain column, using generic types instead. The Oracle data type can be obtained from the LSA database using standard database tools, if necessary.

2.1. Optic definitions

The LSA database table OPTICS defines the optics available from LSA. It provides a link between the key identifying the optic and the database id used to reference the optic in the tables storing the actual optic dependent data.

In practice, optics will typically be defined for a small part of the whole FAIR facility only, for instance a particular circular accelerator or a certain transfer line. On the other hand, the constraints on the table OPTICS require the key identifying the optic, i.e. the entry in column OPTIC_NAME, to be unique over the whole FAIR facility.

Therefore, a naming convention is required in order to facilitate the book keeping on optics. Each optic name shall be composed of two identifiers:

- a string <TRANSFER_LINE> identifying the part of the accelerator the optic is defined for;
- a string <LABEL> identifying the type of optic.

The full optic name is then defined as <TRANSFER_LINE>_<LABEL>.

The beamline identifier shall follow the following conventions:

- For circular accelerators, <TRANSFER_LINE> must be set to the name of the accelerator, e.g. SIS18, CRYRING, CR.
- For transfer lines, <TRANSFER_LINE> must be set to the name of the transfer line, e.g. SIS18_TS_ESR, ESR_CRYRING, SIS100_CBM.

The label should be a short hand for the purpose of the particular optic. If an optic is supposed to be the default for the corresponding beamline, the label STANDARD must be used.

When creating an entry for a particular optic in the table OPTICS, the optic name must be written to the column OPTIC_NAME, while the label must be written to the column TITLE.

Table 7 lists the columns and data types of this database table together with a description of their meaning.

2.2. Strength values

The LSA database table OPTIC_STRENGTHS associates optic dependent strength values with devices. In LSA, devices are abstractions of the power converters driving a single magnet or a string of magnets. Thus, the table OPTIC_STRENGTHS allows to associate strengths with magnets or strings of magnets.

The machine model used in the settings management to calculate settings requires a theory strength value of each magnet or string of magnets for every optic defined. These theory strength values will be stored in the table OPTIC_STRENGTHS, associated with the corresponding device.

Table 8 lists the columns and data types of this database table together with a description of their meaning.

Table 7: Columns of the LSA database table `OPTICS` used to define the available optics.

Column	Type	Description
`OPTIC_NAME`	string	unique identifier for the optic: `<TRANSFER_LINE>_<LABEL>`
`TITLE`	string	label for purpose of the optic: `<LABEL>`
`USED_FOR`	string	textual description of the purpose of the optic
`OPTIC_ID`	integer	*database id of the optic, automatically created by the database*
`GEOMETRY_FILE`	string	*not used for FAIR, can be ignored and need not be filled*
`BASE_STRENGTH_FILE`	string	*not used for FAIR, can be ignored and need not be filled*
`OVERRIDE_STRENGTH_FILE`	string	*not used for FAIR, can be ignored and need not be filled*
`PROGRAM_VERSION`	string	*not used for FAIR, default value* `None` *provided by the import script*
`PARTICLE_TRANSFER`	integer	*obsolete, has been superseded by table* `OPTICS_PART_TRANS`

Table 8: Columns of the LSA database table `OPTIC_STRENGTHS` used to store theory strengths for devices.

Column	Type	Description
`OPTIC_ID`	integer	database id of the optic, foreign key into the database table `OPTICS`
`LOGICAL_HARDWARE`	integer	database id of the device, foreign key into the database table `DEVICES`
`STRENGTH_L`	float	normalized integral strength value of the device for the specified optic
`BEAM`	string	*fixed entry* `B1` *for FAIR, provided by the import script*
`STRENGTH`	float	*not used for FAIR, can be ignored and need not be filled*

Table 9: Columns of the LSA database table OPTIC_PARAMETERS used to store theory values for parameters.

COLUMN	TYPE	DESCRIPTION
OPTIC_ID	integer	database id of the optic, foreign key into the database table OPTICS
PARAMETER_ID	integer	database id of the parameter, foreign key into the database table PARAMETERS
PARAMETER_VALUE	float	value of the parameter for the specified optic

2.3. Optic parameters

The LSA database table OPTIC_PARAMETERS associates optic dependent values with parameters. In LSA, parameters are used to represent physical or hardware properties used in the machine model. The table OPTIC_PARAMETERS allows to store values for such parameters in any optic.

The generic ring model used in the settings management to calculate settings requires a number of theory values of global properties for every optic defined (see table 2). These theory values will be stored in the table OPTIC_PARAMETERS, associated with the corresponding parameter.

Of course, the usage of the table OPTIC_PARAMETERS is not restricted to the global properties of the generic ring model. Machine modelers are free to use this table for storing other optic dependent values at their convenience. As an example, the machine model of SIS18 makes use of additional optic parameters to model the optic dependence of the bumper strengths for multi-turn injection.[2]

Table 9 lists the columns and data types of this database table together with a description of their meaning.

[2] In principle, the theory strengths of magnets could also be stored in the table OPTIC_PARAMETERS, rendering the table OPTIC_STRENGTHS obsolete. The strategy presently employed follows the usage of these tables at CERN.

Table 10: Columns of the LSA database table KNOBS used to store the coefficients of linear relations between two parameters.

Column	Type	Description
OPTIC_ID	integer	database id of the optic, foreign key into the database table OPTICS
KNOB	integer	database id of the knob parameter, foreign key into the database table PARAMETERS
COMPONENT	integer	database id of the component parameter, foreign key into the database table PARAMETERS
KNOB_VALUE	float	coefficient for the relation between knob and component in the specified optic

2.4. Knobs

The LSA database table KNOBS associates optic dependent values with pairs of parameters, of which one is referred to as *knob*, the other as *component*. Since LSA parameters represent properties of the machine model, this means that the table KNOBS allows to store values for combinations of such parameters in any optic.

The generic ring model used in the settings management to calculate settings makes frequent use of knob values for optic dependent linear approximations. For instance, knob values are used to calculate the magnet strength changes associated with changes in tunes and chromaticities (see tables 3 and 4). Another prominent example is provided by the modeling of local orbit bumps, where the optic dependent strengths of the correctors are stored as components together with a knob parameter representing the amplitude of the bump. Similarly, optic dependent strength values for fast extraction kickers and injection bumpers can be stored in the table KNOBS. Other use cases are easily conceivable. Machine modelers are fee to use this table for optic dependent linear relations at their convenience.

Table 10 lists the columns and data types of this database table together with a description of their meaning.

2.5. Twiss parameters

The LSA database table TWISS_OUTPUTS is designed to hold the Twiss parameters at each element in the beamline for a given optic as well as the element strengths of magnets.

Note in this context that LSA elements are separate entities not to be confused with LSA Devices. The LSA element represents an accelerator component in the beamline, essentially only for the purpose of storing Twiss parameters at beamline components. Considering magnets and their power converters as an example, the magnets are represented as elements, while the power converters are represented as devices.

The data stored in the table TWISS_OUTPUTS is not used directly in the machine model. However, it is frequently employed to calculate linear response in applications such as orbit and trajectory control, optic measurement and control. Even in the machine model, the orbit response matrix calculated from Twiss parameters is used to define the target value for the orbit at the beam position monitors in the presence of intentional orbit deformations like extraction bumps or radial displacements.

For these applications it is essential that the Twiss parameters stored in the table TWISS_OUTPUTS are defined at the *center* of each element.

Table 11 lists the columns and data types of this database table together with a description of their meaning.[3]

2.5.1. Elements and devices

Before going on to describe in more detail which of the columns in TWISS_OUTPUTS need to be filled for which elements, the reader is reminded of the distinction between elements and devices in LSA, which are distinct entities not to be confused with each other.

The LSA element represents an accelerator component in the beamline, essentially only for the purpose of storing Twiss parameters at beamline components. In particular, no setting can ever be associated with an element.

Devices, on the other hand, represent the hardware controlled by the control system, e.g. power converters or beam instrumentation readout systems. The properties of these devices can be represented by LSA parameters, which in turn are used to create settings and send set values to the devices or read measured values from them.

While elements and devices are distinct entities, the LSA data model provides a mapping between the two by means of the database table ELEMENTS_LOGICAL_HARDWARE. In the simplest case of a power converter driving a single magnet, this mapping is one-to-one. If, on the other hand, a power converter drives a string of magnets, each magnet will be represented by an individual element. In that case, the mapping from elements to device is many-to-one, mapping all elements in the string onto the same device representing the power converter.

It is important to note that the table TWISS_OUTPUTS deals exlusively with elements, which makes sense because the optical functions are defined at the beamline components. If one has the need to access such values for a particular device, one needs to use the mapping from elements to device to retrieve the possibly many elements connected to the device.

[3]Note that, even though the columns resemble the output created by the MADX TWISS command, there are actually slight differences in the conventions for the dispersion functions (see section 2.5.5).

Table 11: Columns of the LSA database table `TWISS` used to store Twiss parameters and strengths for elements.

COLUMN	TYPE	DESCRIPTION
`OPTIC_ID`	integer	database id of the optic, foreign key into `OPTICS`
`ELEMENT_ID`	integer	database id of the element, foreign key into `ELEMENTS`
`BETX`	float	Twiss parameter β_x
`BETY`	float	Twiss parameter β_y
`ALFX`	float	Twiss parameter $\alpha_x = -1/2\,(\partial\beta_x/\partial s)$
`ALFY`	float	Twiss parameter $\alpha_y = -1/2\,(\partial\beta_y/\partial s)$
`MUX`	float	phase advance in tune units $\mu_x = 1/(2\pi)\int ds/\beta_x$
`MUY`	float	phase advance in tune units $\mu_y = 1/(2\pi)\int ds/\beta_y$
`DX`	float	dispersion[3] $D_x = \partial x/\partial\delta$
`DY`	float	dispersion[3] $D_y = \partial y/\partial\delta$
`DPX`	float	dispersion[3] $D'_x = \partial x'/\partial\delta$
`DPY`	float	dispersion[3] $D'_y = \partial y'/\partial\delta$
`X`	float	closed orbit x
`Y`	float	closed orbit y
`PX`	float	closed orbit x'
`PY`	float	closed orbit y'
`K0L`	float	integral strength of normal dipole component
`K1L`	float	integral strength of normal quadrupole component
`K2L`	float	integral strength of normal sextupole component
`K3L`	float	integral strength of normal octupole component
`K4L`	float	integral strength of normal decapole component
`K5L`	float	integral strength of normal dodecapole component
`K1SL`	float	integral strength of skew quadrupole component
`K2SL`	float	integral strength of skew sextupole component
`K3SL`	float	integral strength of skew octupole component
`HKICK`	float	integral kick strength of horizontal corrector
`VKICK`	float	integral kick strength of vertical corrector
`BEAM`	string	*fixed entry* `B1` *for FAIR, provided by the import script*

Table 12: Columns of the LSA database table TWISS_OUPUTS with mandatory entries.

Columns	Description
BETX, BETY, ALFX, ALFY, MUX, MUY, DX, DY, DPX, DPY	linear optical functions
K0L, K1L	strengths of bending magnets and lattice quadrupoles

2.5.2. Mandatory entries

The table TWISS_OUTPUT has a large number of columns, not all of which are actually used by the machine model and standard applications. Apart OPTIC_ID and ELEMENT_ID, entries for the following quantities are, however, mandatory:

- all linear optical functions, i.e. β_z, α_z, μ_z, D_z, D'_z, where $z = x, y$;
- the strengths $k_0\ell$ and $k_1\ell$ of bending magnets and lattice quadrupoles.

All other columns except BEAM are optional and need not be filled. If no value is provided, a default value of zero will be supplied automatically at the database level. Note that this includes, in particular, the closed orbit functions, since the design orbit typically has zero offset.

The column BEAM requires an entry due to a database constraint. However, since for FAIR this entry is fixed to the string B1, this value need not be provided by machine modelers and will be automatically created by the import script.

2.5.3. Mandatory elements

The data for the import into the table TWISS_OUTPUTS is typically calculated using an ion optical simulation program. The input lattice used in the simulation is usually created and maintained by those who perform the simulations. For some of the FAIR machines, standard lattices may be available, but presently this cannot be granted. The sequence of elements in the lattice is therefore to some extent up to the choice of the simulators.

On the other hand, the usage of Twiss parameters in the machine model and the standard applications relies on the existence of entries for certain elements. Without those entries, such basic functionalities as settings generation and closed orbit correction will fail. **It is therefore absolutely vital that for the mandatory elements specified below, values of the linear optical functions will be stored in the LSA database.**

The values of the linear optical functions at the following elements are mandatory and must be imported into the table TWISS_OUTPUTS:

- all magnets, including all types of correctors, especially orbit correctors, chromaticity and resonance sextupoles, multipole correctors, fast extraction kickers, injection bumpers, and magnetic septa;
- all beam instrumentation devices measuring beam position;

Table 13: LSA element types[4] of mandatory elements for linear optical functions in the table `TWISS_OUTPUTS`.

LSA Element Type	Components
`SBEND`, `RBEND`	main bending magnets
`QUADRUPOLE`	lattice and correction quadrupoles
`HKICKER`, `VKICKER`	orbit correctors (exclusively)
`TKICKER`	injection bumpers fast extraction kickers, septa
`SEXTUPOLE`, `OCTUPOLE`, `MULTIPOLE`	lattice and correction multipoles
`SOLENOID`	solenoids
`HMONITOR`, `VMONITOR`, `MONITOR`	position measuring devices
`RCOLLIMATOR`	aperture restrictions

- aperture restrictions, especially collimators or beam catchers.

For those elements, it is essential that the linear optical functions are defined at the *center* of the element.

The only exception to this rule is the case of extended aperture restrictions, where the optical functions should be defined at the position of the smallest opening. If an aperture restriction like a collimator is very long, so that the beam size evolution along the collimator matters, it may be better to put two aperture restriction elements at the start and end of the collimator. As long as the aperture restriction under consideration is not inside a focusing element, this guarantees that acceptance can be estimated independent of the particular optic chosen.

Table 13 summarizes the list of mandatory LSA element types in the database table `TWISS_OUTPUTS` together with a description of the type of accelerator components to be represented by them.[4]

In addition to the elements listed above, additional *markers* are mandatory:

- Two markers at the start and end of the lattice. The names of these markers must follow the pattern `M.<RING>.START` and `M.<RING>.END`, where `<RING>` is the name of the circular accelerator[5], e.g. SIS18, ESR, CRYRING, etc.

- Markers at the start of each super-period of the lattice. The names of these markers must follow the pattern `M.<RING>.SPER.<#n>`, where `<#n>` is the a consecutive number identifying the nth super-period of the circular accelerator `<RING>`, starting at one. Also, the start of the first super-period must coincide with the start of the lattice, i.e. `M.<RING>.START`.

- Markers at the entrance to and exit from the effective field of each bending magnet. The naming convention must follow the pattern `M.<BEND>.START` and

[4] Note that, even though the LSA element types resemble the MADX element classes, consistency between them is actually not guaranteed (see section 2.5.5).

[5] This name must be identical to the string value of the Java enum GsiAccelerator in the LSA API.

Table 14: Mandatory markers for linear optical functions in the table TWISS_OUTPUTS.

Marker	Location
`M.<RING>.START`	Start of ring
`M.<RING>.END`	End of ring
`M.<RING>.SPER.<#n>.START`	Start of super-period n of ring
`M.<BEND#n>.START`	Start of bending magnet n
`M.<BEND#n>.END`	End of bending magnet n
`M.<RING>.INJECTION`	Injection point of ring
`M.<RING>.EXTRACTION`	Extraction point of ring (if present)

M.<BEND>.END, where <BEND> is the name of the element representing the bending magnet.

- One marker at the injection point of the lattice. The injection point is defined as the position where the injection line and the ring meet. Usually, this is defined by the exit of a magnetic or electro-static septum. The name of this marker must follow the pattern M.<RING>.INJECTION, where <RING> is the name of the accelerator as above.

- One marker at the extraction point of the lattice, *if the ring has an extraction*. The extraction point is defined as the position where the extraction line and the ring meet. Usually, this is defined by the entrance to a magnetic or electro-static septum. The name of this marker must follow the pattern M.<RING>.EXTRACTION, where <RING> is the name of the accelerator as above.

Table 14 summarizes the list of mandatory markers at which optical functions need to be available in the database table TWISS_OUTPUTS.

Of course, machine modelers are free to add entries for additional elements in the table TWISS_OUTPUT, if they need them to model additional aspects not covered by the standard tools. It should be noted, however, that those elements would first have to be defined in the database table ELEMENTS. This requires propercoordination with the controls group, especially the LSA team.

2.5.4. Nested elements

Due to space constraints accelerators often have nested components, especially in small machines. Some cases frequently encountered for FAIR machines are listed here as reference cases:

1. combined horizontal and vertical steering magnets (e.g. SIS100);
2. nested multipole correction magnets (e.g. SIS18, SIS100);
3. correction windings used as replacement for separate horizontal steering magnets, either on bending magnets (e.g. SIS18, ESR) or on sextupoles (e.g. CR);

4. beam position monitors integrated into magnets (e.g. CR).

If elements are nested, each element controlled independently by a separate device[6] must have a separate entry in the table `TWISS_OUTPUTS`.
For the reference cases above, this requirement has the following implications:

1. Combined horizontal and vertical steering magnets need separate entries for each plane, with one element of type `HKICKER` and one element of type `VKICKER` connected to the corresponding devices representing the individual power converters. Both elements have the same s coordinate, located at the center of the combined magnet.

   As an example, consider the combined steering magnet in cell `1S11` of SIS100. There are two power converters, one for each plane, represented by two devices `1S11KH1` and `1S11KV1`. The two steering components of the magnet are represented by two elements with the same names, mapped one-to-one in the table `ELEMENTS_LOGICAL_HARDWARE`. The table `TWISS_OUTPUTS` then must contain an entry for both elements, *even if the lattice used to calculate the Twiss parameters would represent this magnet as a combined element, e.g. a* `KICKER` *element in MADX.*

2. Nested multipole correction magnets need separate entries for each multipole component, represented by an element of appropriate type and connected to the corresponding devices representing the individual power converters. All elements have the same s coordinate, located at the center of the combined magnet.

   As an example, consider the combined multipole corrector magnets in cell `1S14` of SIS100. This magnet provides a normal quadrupole, a skew sextupole, and a normal octupole component, each powered by an independent power converter. The corresponding devices and elements are `1S14KM1Q`, `1S14KM1S`, `1S14KM1O`, respectively, with element types `QUADRUPOLE`, `SEXTUPOLE`, and `OCTUPOLE`.[7] All elements have the same s coordinate, located at the center of the combined magnet.

3. Correction windings used for horizontal steering need an independent element of type `HKICKER`, connected to the device representing the corresponding power converter. The element must be located at the longitudinal center of the steering field created by the winding. *This element is mandatory even if the location coincides with the center of the main magnet containing the correction winding.*

   As an example, consider the horizontal steering coil `GE01KX1` in the ESR. This coil covers one fourth of the yoke area at the upstream end of the sixty degree bending magnet `GE01MU1`. Thus, the center of the steering field is located at one eight of the effective length of the bending magnet. The element is connected to the device with the same name representing the corresponding power converter.

4. Beam position monitors integrated into magnets need a separate entry. Depending on the pickup layout, the monitor may be represented either by a single element of type `MONITOR`, or by two separate elements of type `HMONITOR` and `VMONITOR`. These elements should have an s coordinate located at the center of the corresponding pickups.

[6] See section 2.5.1 for an explanation of the relation between elements and devices.
[7] Note that the LSA element types do *not* distinguish between normal and skew multipoles.

As an example, consider the beam position monitor integrated into the vertical steering magnet `CR01KV2`. The monitor can be represented by a single element `CR01DX02` located at the center of the monitor, which in this case coincides with the center of the magnet.

2.5.5. Caveat for MADX users

The reader familiar with the ion optical simulation program MADX will notice that the format of the table `TWISS_OUTPUTS` (see table 11 on page 15) is reminiscent of the columns in the output of a MADX Twiss table as obtained by the `TWISS` command. Indeed, the Twiss parameters in the table are labeled by the same identifiers used by MADX for the linear optical functions. The additional columns (strengths, beam, and closed orbit) can be included in the MADX table by selecting them explicitely for inclusion in the Twiss table. However, the use of the MADX program for calculation of the Twiss parameters is by no means mandatory. Any ion optical simulation program can be used to prepare the data for import into the `TWISS_OUTPUTS` table.

Care must be taken when using MADX to calculate the dispersion functions $D_{x,y}$ and $D'_{x,y}$, because MADX conventions deviate from the literature standard: MADX defines these functions in terms of the canonical longitudinal momentum p_t instead of the relative momentum deviation $\delta \equiv \Delta p/p$. Since $p_t = \beta \cdot \delta$, the MADX functions are related to the standard definition as follows:

$$D_{x,y} = \beta \cdot (D_{x,y})_{\mathrm{MADX}} \tag{21}$$
$$D'_{x,y} = \beta \cdot (D_{x,y})_{\mathrm{MADX}} \tag{22}$$

Using standard conventions is absolutely essential, since otherwise the correct interpretation of the dispersion functions would require knowledge of the relativistic β of the beam used to simulate the Twiss parameters. **Machine modelers relying on MADX to calculate the Twiss parameters should keep this in mind and perform the necessary conversion when preparing the Twiss parameters for the import into the LSA database.**

Furthermore, the LSA element types listed as mandatory in table 13 on page 17 remind the MADX user of the MADX element classes. While these types haven been initially defined in LSA based on those MADX classes, there is no guarantee that possible changes in MADX element classes will be incorporated into LSA. For instance, the MADX element class corresponding to the LSA element type `RCOLLIMATOR` has been removed from MADX. Therefore, if one uses a MADX `COLLLIMATOR` class in simulations, the corresponding element in LSA would have the element type `RCOLLIMATOR`.

3. Database Import

3.1. File formats and constraints

```
<TABLE_NAME> ; <OPTIC_NAME>
<COLUMN#1_NAME>; ... ; <COLUMN#n_NAME>
<VALUE>        ; ... ; <VALUE>
.                 .    .
.                 .    .
.                 .    .
<VALUE>        ; ... ; <VALUE>
```

3.1.1. Optics definition

The format for the definition of the optics deviates slightly from the general format of the table data, since it defines the optic name which is used by the other tables to reference the optic. Therefore, the optic name is not placed after the table name on the first header line, but rather shifted into a separate column with header OPTIC_NAME.[8]

```
OPTICS
OPTIC_NAME   ; TITLE   ; USED_FOR ; TRANSFER_LINE
<OPTIC_NAME> ; <LABEL> ; <DESC>   ; <TRANSFER_LINE>
```

3.1.2. Strength values

```
OPTIC_STRENGTHS ; <OPTIC_NAME>
DEVICE_NAME; STRENGTH_L
<DEVICE#1> ; <KNL#1>
.            .
.            .
.            .
<DEVICE#n> ; <KNL#n>
```

3.1.3. Optic parameters

```
OPTIC_PARAMETERS ; <OPTIC_NAME>
PARAMETER_NAME ; PARAMETER_VALUE
<PARAMETER#1>  ; <VALUE#1>
.                .
.                .
.                .
<PARAMETER#n>  ; <VALUE#n>
```

3.1.4. Knobs

```
KNOBS ; <OPTIC_NAME>
KNOB_NAME ; COMPONENT_NAME ; KNOB_VALUE
<KNOB#1>  ; <COMPONENT#11> ; <VALUE#11>
```

[8] This format would in principle allow to accumulate all optic definitions for an accelerator in a *single* file instead of creating one file per optic.

```
.             .                 .
<KNOB#1>   ; <COMPONENT#1m> ; <VALUE#1m>
.             .                 .
.             .                 .
.             .                 .
<KNOB#n>   ; <COMPONENT#n1> ; <VALUE#n1>
.             .                 .
<KNOB#n>   ; <COMPONENT#nm> ; <VALUE#nm>
```

3.1.5. Twiss parameters

```
TWISS_OUTPUTS ; <OPTIC_NAME>
ELEMENT_NAME;BETX;BETY;ALFX;ALFY;MUX;MUY;DX;DY;DPX;DPY;K0L;K1L;<OPT_COLS>
<ELEMENT#1> ;<>  ;<>  ;<>  ;<>  ;<> ;<> ;<>;<>;<> ;<> ;<> ;<> ;<OPT_VALS#1>
.                                                               .
.                                                               .
.                                                               .
<ELEMENT#n> ;<>  ;<>  ;<>  ;<>  ;<> ;<> ;<>;<>;<> ;<> ;<> ;<> ;<OPT_VALS#n>
```

3.2. Directory structure

```
<ROOT>
 ├─ <ACCELERATOR#1>
 │   ├─ OPTICS
 │   ├─ OPTIC_STRENGTHS
 │   └─ TWISS_OUTPUTS
 ├─ ...
 ├─ ...
 ├─ ...
 └─ <ACCELERATOR#n>
     ├─ OPTICS
     ├─ OPTIC_STRENGTHS
     ├─ TWISS_OUTPUTS
     ├─ OPTIC_PARAMETERS
     └─ KNOBS
```

3.3. Import scripts

4. Usage of Optics Data in the Machine Model

5. Usage of Optics Data in Beam Based Applications

A. Version History

Versions with a zero major version (0.x) are draft versions and have no official status.

- 0.1 First draft shared with colleagues for discussions
- 0.2 Current working version