MADX Tools for building and analysing ATS optics

R. De Maria (on behalf of S. Fartoukh)

CERN, ABP/LCU

February 1, 2012

Aim:

- hand-on tutorial for who wants to build ATS optics,
- detailed explanation for who wants to use and analyse ATS optics (aperture, DA, chromatic properties),
- expose any interested colleague on the tools and techniques used so far,
- get feedback from collaborators, users and motivate for new ideas

Tools for building ATS optics.

Reference source (with heritage from Phase I SLHCV1.0/2.0):

`/afs/cern.ch/eng/lhc/optics/SLHCV3.0 (S. Fartoukh)`

Content:

- Final products: thick/thin layout and optics (injection, pre-squeeze, squeeze)
- Directory structure: base files, toolkit, aperture, errors, beam-beam, squeeze
- Work-flow:
  - pre-squeeze: triplet/MS design, IR1/5 matching, phase adjustment, IR15 crossing knobs, chromatic correction, dispersion correction knobs, aperture estimates/evaluation, save optics
  - squeeze with IR2846, IR28 crossing knobs, chromatic correction, dispersion correction knobs, save optics
  - unsqueeze to injection optics, aperture evaluation, save optics
  - makethin with invariants and selected knobs, error assignment, beam beam kick/footprint

Variants (file with same names are logically identical):

`/afs/cern.ch/eng/lhc/optics/SLHCV3.0 (123T/m)`
`/afs/cern.ch/eng/lhc/optics/SLHCV3.01 (123T/m with sext in Q10)`
`/afs/cern.ch/eng/lhc/optics/SLHCV3.1a (100T/m, crossings closed at Q4)`
`/afs/cern.ch/eng/lhc/optics/SLHCV3.1b (150T/m, crossings closed at Q4)`
`/afs/cern.ch/eng/lhc/optics/ATS_V6.503 (200T/m tested in the LHC)`

Base files SLHCV3.0

Content of the `/afs/cern.ch/eng/lhc/optics/SLHCV3.0` directory

aperture	SLHC.inj.str	SLHC.inj.thin.str
beambeam	SLHC.col_1111.str	SLHC.col.thin_1111.str
errors	SLHC.col_2222.str	SLHC.col.thin_2222.str
toolkit	SLHC.col_2828.str	SLHC.col.thin_2828.str
IR_OPTICS	SLHC.col_2882.str	SLHC.col.thin_2882.str
removeinstall.madx	SLHC.col_4444.str	SLHC.col.thin_4444.str
job_formakethin.madx	SLHC.col_8228.str	SLHC.col.thin_8228.str
job_tracking.mask	SLHC.col_8282.str	SLHC.col.thin_8282.str
samplejob_general.madx		

What can be reused from V6.503 is taken, in particular `V6.5seq`. Each variant is independent from the other.

Base files SLHCV3.1a/b (work in progress)

Small modification for the new version SLHCV3.1

```
aperture                        opt_100_0500_0500_v1.00.madx
beambeam                        opt_100_0500_0500_v1.00thin.madx
errors                          opt_100_0190_0190_v1.00.madx
toolkit                         opt_100_0190_0190_v1.00thin.madx
slhc_sequence.madx              opt_100_0150_0150_v1.00.madx
slhc_removeinstall.madx         opt_100_0150_0150_v1.00thin.madx
crab_install.madx
job_sample.madx
```

Basic usage:

```
system,"ln -fns /afs/cern.ch/eng/lhc/optics/SLHCV3.1a slhc";
call,file="slhc/toolkit/macro.madx";
call,file="slhc/slhc_sequence.madx";
call,file="slhc/opt_100_0190_0190_v1.00.madx";
call,file="slhc/slhc_removeinstall.madx";
exec,mk_beam(7000);
exec,check_ip(b1); exec,check_ip(b2);
```

Build LHC upgrade sequence:

- `slhc_sequence.madx` Import V6.5.seq, define new elements and parametrized positions:

```
l.MQXL      :=          8.006 ; l.MQX       :=          6.958 ;
dq1q2a      :=          2.688 ; dq2aq2b     :=          1.640 ;
dq2bq3      :=          2.688 ;
deltaposD2  :=        -15.000 ; deltaposQ4  :=          0.000 ;
deltaposQ5  :=         11.000 ; deltaposQ6  :=          0.000 ;
```

- `slhc_removeinstall.madx` Modify sequence, to be used anytime a parameter change.
- `crab_install.madx` Install crab cavities and computes required voltages.

IR15 matching

Matching pre-squeeze (or nominal-like) IR15 optics.

- `toolkit/macro.madx`: contains all macro used
- `SELECT`: usuals select of periodic arc cells to `s.ds.l e.ds.r` markers. Works for all insertions: e.g. `exec,SELECT(7,67,78,b2)`).
- `SELECT15`: computes phase advances between specific sextupole families used for chromatic aberration control, e.g. `selectIR15(5,45,56,b1);`
- `toolkit/rematch_ir15_b12imb.madx` Example: match IR1/5 Beam1/2 simultaneously (needed for left/right phase matching), options for Q4-6 (strict antisymmetry, control of Beam1/2 imbalance). Other rematching scripts are available for different aim.
- `toolkit/mktriplet.madx` Independent sequence containing macros to generate optimized triplet layouts.

  ```
  mk_triplet(bxs,bys,grad,betxq4,betyq4): macro= ...
  exec,mk_triplet(0.60,0.60,150,betx_marke, bety_marke);
  ```

IR15 phase adjustment

Core script for the ATS optics: find the phase advance between the triplet and the two chosen (*strong*) sextupole families such that a Δk_2^f, Δk_2^d change the imaginary part of the chromating beating phasor [1] at the IP for x, y plane, orthogonally. It is based on matching using virtual elements. The procedure finds also a first guess for the strong sextupole families, suitable for the pre-squeeze β^*.

- `toolkit/macro.madx`: contains all macro used
- `toolkit/rematch_betabeat.madx`: contains the whole procedures which uses:
- `PHASESHIFT(LOC,dphix,dphiy,BIM)`: compute and install a parametric symplectic matrix element that has the only effect of shifting the phase advance (phase trombone) without inducing β-beating and dispersion beating.
- `toolkit/rephase_lhc.madx`: rephase the arcs to keep the same working point (involve rematching the entire rings).

[1] for both planes $W_0 \exp(2i\pi\phi_0) = B_0 + iA_0 = \Delta\beta_0/\beta_0 + i(\Delta\alpha_0 - \alpha\Delta\beta_0/\beta_0) = \frac{i}{2}\int_{s_a}^{s_0}(k_2(s)D_x(s) - k_1(s))\beta(s)\exp(2i\pi\Delta\mu(s))ds$

IR15 crossing knobs

Computes the crossing angle and parallel separation knobs using two strategies.

- `toolkit/xing.IP15.madx`: used in SLHCV3.0 and SLHCV3.01, uses orbit corrector in Q3, Q4, Q5, Q6
- `toolkit/rematch_xing_ir15.madx`: used SLHCV3.1, uses upgrade specific new orbit correctors in Q1, Q3, D2 to close the bump before crab cavities.

Bumps are parametrized by:

```
on_sep1:=1;on_x1:=1;phi_IR1 :=   90.000
on_sep5:=1;on_x5:=1;phi_IR5 :=    0.000
```

including `phi_IR1`, `phi_IR5` which allow to choose arbitrarily the crossing and separation plane for the two IPs (default VH crossing).

Tune and chroma adjustment

Once triplet induced off-momentum beta-beating is corrected by the *strong* arcs (81,12,45,56), the phase advances and sextupole strengths must be frozen in those regions, therefore tune and chroma adjustments must be performed using the *weak* arcs (23,34,67,78).

- `CORCHROMA_WEAK(2,B1)`: e.g. correct Q' to 2 units using only weak arcs for Beam2.

NB: this is for building optics, for *operational* adjustment proper knobs are setup in the optics files (see later `save_optics.madx`).

Dispersion correction knobs

Crossing angle and parallel separation bumps induce dispersion in the machine that can be extremely large for small beta*.
Strict $\pi/2$ arc cells, strong sextupoles, orbit excitation, ATS phases generates a dispersion kick in phase with the kick caused by the orbit in the triplets.

- `toolkit/match_disp.madx`: computes the orbit corrector strengths to build the knobs.

APERTURE ESTIMATES/EVALUATION

The repository includes an aperture model for the new elements. Based on the Phase I script prepared also by J. Miles, J-B. Jeanneret for several beam screen types: round, racetrack, octagonal.

- `aperture_upgrade_IT.madx` define aperture model for the common pipe region TAS - D1. Starts from the inner coil diameter and build a beam screen that fits in the cold bore using mechanical input and tolerances. Default for octagagonal beam screen shape (`aperture_upgrade_octagon.madx`).
- `aperture_upgrade_MS.madx` define aperture model for region from TAN to Q5
- `offset.ip[15].b[12].tfs` used for the cold bore separation of 2-in-1 elements TAN-D2.
- `PLOT_AP1(NIR,BIM,NRJ,FULL)` compute aperture using nominal settings.
- `check_aperture.py` tool for checking aperture injection constraints.

In SLHCV3.0 the IT aperture is scaled to 150mm. In SLHCV3.1 the IT aperture is parametrized by `ap_mqx`.
NB. aperture model for the HL-LHC is evolving quickly.

Save optics

Optics are saved in pretty-print format containing any information to generated an optics starting from a parametrized sequence: layout parameters, strengths, knobs definitions, optics parameter informations.

- `toolkit/save_optics.madx` madx script (using `toolkit/save_optics.f`)
- `ARC_SQUEEZE` controls the knobs generation depending on the type of optics: 0: injection, 0.5: pre-squeeze, 1: squeeze.

Once an SLHCV3.1 optics is saved, it can be used with optics:

```
system,"ln -fns /afs/cern.ch/eng/lhc/optics/SLHCV3.1a slhc";
call,file="slhc/toolkit/macro.madx";
call,file="slhc/slhc_sequence.madx";
call,file="slhc/opt_100_0190_0190_v1.00.madx";
call,file="slhc/slhc_removeinstall.madx";
exec,mk_beam(7000);
exec,check_ip(b1); exec,check_ip(b2);
```

Squeeze IR2846

Find the new optics for IR2/8/4/6 that squeeze IR1/5.

- `SELECTIRAUX(7,8,1,2,3,b1,scxir1,scyir1,betx0,bety0)` find the boundary condition that further squeeze IR1 starting from a presqueeze beta* (betx0, bety0) and reducing by two factors (scxir1, scyir1).
- `toolkit/rematch_ir2/4/6/8` perform the rematch
- `toolkit/xing.IP28.madx` recompute the crossing knobs

Chromatic correction

Re-optimize the sextupole families to reduce any leakage of chromatic-beating.

- `global_rematchw(Ix1,Iy1,Ix5,Iy5,1)`: macro that optimize sextupoles strengths to create a (balanced l/r) kick that compensates as locally as possible the off-momentum beta-beating absorbing any leakage from the whole ring.
- `intkbeta(5,L,B1)` macro that computes $\int k\beta ds$

Usage:

```
exec,intkbeta(5,L,B1); exec,intkbeta(5,R,B1);
exec,intkbeta(5,L,B2); exec,intkbeta(5,R,B2);
DeltaIx=(ix5.l_b1-ix5.r_b1)/2;
DeltaIy=(iy5.l_b1-iy5.r_b1)/2; DeltaIy=-DeltaIy;

exec,CORCHROMA_WEAK(2,b1); exec,CORCHROMA_WEAK(2,b2);
exec,global_rematchw(DeltaIx,DeltaIy,DeltaIx,DeltaIy,1);
exec,global_rematchw(DeltaIx,DeltaIy,DeltaIx,DeltaIy,2);
exec,CORCHROMA_WEAK(2,b1); exec,CORCHROMA_WEAK(2,b2);
```

Injection optics

- rematch with aperture constraints (guess estimates to be always checked with the aperture model)
- working point shift from 0.31 to 0.28 and 0.32 to 0.31 by reducing the phase of IR1/5.
- for IR1/5 left/right and triplet optimization constraints need to be dropped.

For optics transitions there are tools (not included in the repository) for

- create optics tables
- interpolating two different optics creating linear interpolating knobs from TFS tables
- generate function with high order polynomial to provide smooth transition for strengths or matching targets

Makethin

Target: generate sequence and optics using only thin elements (for symplectic tracking) as close as possible to the thick optics.

`job_formakethin.madx`
`toolkit/rematch_thin.madx`

Invariants:

- phase advances of arcs and IR
- twiss parameters at all IPs (including IR3 and IR7)
- selected quadrupole strengths for squeeze: IR1/5(Q4-6,QT13.R1B1, QT13B2), IR2(QX), IR8(QX, QT13L8B1), IR6(Q4.L6B1,Q4.R6B2), IR3/7(Q4-Q5),
- selected quadrupole strengths for injection: IR1/5(QX,QT13.R1B1,QT13.L1B2,MCBX), IR2(QX,Q4-5B1,QTL11.R2B1,QT12L2B1), IR8(QX,Q4-5B2,QT13L8B1), IR6(Q4.L6B1,Q4.R6B2), IR3/7(Q4-Q5),

Knobs:

- insertions quads
- KQF/D
- KSF,KSD
- orbit corrector (but not MCBX)

Imperfection assignment and correction

So called mask file (`job_tracking.mask`) used also to generate sixtrack input.

- **as-built** machine: error assignment routines (T. Risseleda) for each class with WISE tables (P. Hagen)
- triplet/D1/D2/Q4/Q4: specification for new elements (to be studied by the collaboration)
- triplet D1 correction (resonance suppression and minimization)
- arc-by-arc correction of the MB imperfection
- global correction (tune, chroma, coupling, orbit)
- beam beam kick installation
- sixtrack generation and/or beam beam

Beam beam kick installation and footprint

Install beam beam elements from optics input.
Added features from H. Grote, 2002:

- automatic calculation of the long range interactions till D1 for any layout
- includes long range interaction in D1 using survey data (`n_insideD1=...`)
- supports Beam 2 as well
- compute separation bump automatically using actual beta to adjust the halo collision

In the mask file:

- `ON_COLLISION`: whether beam in collision with some halo defined by: halo1, halo2, halo5, halo8 using `CALCULATE_XSCHEME($halo1,$halo2,$halo5,$halo8)`
- `ON_BB_SWITCH`: install bb lenses using `INSTALL_BB_MARK(b1)`, `CALCULATE_BB_LENS`
- `ON_BB_CHARGE`: scale beam beam charge
- `b_t_dist`: bunch separation in ns (typically 25 or 50)
- `n_insideD1`: number of additional parasitic encounters inside D1