

Tools within the LHC Online Model

- OM modules
- Example of using OM
- Overview of use-cases and tools in view of LHC commissioning stages (A.1 – A.4)

OM Modules (as of now; scripts mostly set up for SPS)

Module	Functionality
'OM server'	madx gui Creating madx input files from the LSA database Processing requests for computations from client software (Java and python API available) Updating settings (magnet strengths) via LSA BPM data acquisition – multi-turn and orbit (via file exchange) BLM data acquisition via LSA (<i>not finished</i>)
Script repository	Python/madx scripts to be run on the OM server: Orbit bumps; arbitrary knobs; aperture checks; orbit correction; energy matching; tracking (orbit in control system format); lattice matching to measured tune/chrom.
Trim 'dummy'	Application similar to LSA-trim but sending trims to OM server (with Jutta Netzel of AB/CO). Will be testing this week.
Analysis toolkit	Tune/optics measurements from few turns of bpm data (<i>not finished</i>) Beam matrix measurement from beam profile scans (including coupling) Driving terms (in collaboration with J. Rowland of RAL)

In control system

Not in control system

An example of OM usage – checking out MADX input files from LSA DB

The screenshot displays the OM server v0.0.3 interface. The main window shows a repository structure on the left and a list of MADX elements in the center. An 'Import Wizard' dialog is open on the right, showing a list of elements to be imported.

Repository Structure:

- Decks
 - SFT-LHCMD_L14400_PDOT_V1 (multiple instances)
 - SPS.TH.elements.elx
 - SPS.TH.sequence.seq
 - SPS.TH.strengths.str
 - SPS.elements.elx
 - SPS.sequence.seq
 - SPS.strengths.str
- Scripts
 - aperture_db
 - bootstrap.madx
 - knob_example.py
 - knob_mad
 - correct-orbit.madx
 - correct-orbit.py
 - convert-orbit.py
 - test.madx
 - tracking
 - tracking_py
 - twiss
 - twiss_aper

MADX Elements List:

```
! Element definition, generated automatically by machine
! @date = Tue Sep 18 12:42:55 CEST 2007
! @accelerator_zone = SPS
QF2.F : QUADRUPOLE, l = 3.085, k1 := kQF2 ;
MBA.F : RBEND, l = 6.260018603, angle := kMBA ;
MBB.F : RBEND, l = 6.260018603, angle := kMBB ;
VVSA : INSTRUMENT, l = 0.175 ;
LSDA.F : SEXTUPOLE, l = 0.42, k2 := kLSDA ;
MDV10107.F : VKICKER, l = 0.25, kick := kMDV10107 ;
BPV : VMONITOR, l = 0.275 ;
QD.F : QUADRUPOLE, l = 3.085, k1 := kQD ;
LSFB.F : SEXTUPOLE, l = 0.423, k2 := kLSFB ;
MDH10207.F : HKICKER, l = 0.25, kick := kMDH10207 ;
BPH : HMONITOR, l = 0.275 ;
LODOUT : OCTUPOLE, l = 0.677, k3 := kLODOUT ;
LSDB.F : SEXTUPOLE, l = 0.42, k2 := kLSDB ;
MDV10307.F : VKICKER, l = 0.25, kick := kMDV10307 ;
LOE10402.F : OCTUPOLE, l = 0.74, k3 := kLOE10402 ;
MDH10407.F : HKICKER, l = 0.25, kick := kMDH10407 ;
MDV10507.F : VKICKER, l = 0.25, kick := kMDV10507 ;
LSE10602.F : SEXTUPOLE, l = 0.74, k2 := kLSE10602 ;
MDH10607.F : HKICKER, l = 0.25, kick := kMDH10607 ;
LOD.F : OCTUPOLE, l = 0.677, k3 := kLOD ;
MDV10707.F : VKICKER, l = 0.25, kick := kMDV10707 ;
LOF.F : OCTUPOLE, l = 0.705, k3 := kLOF ;
LSFA.F : SEXTUPOLE, l = 0.423, k2 := kLSFA ;
MDH10807.F : HKICKER, l = 0.25, kick := kMDH10807 ;
MDV10907.F : VKICKER, l = 0.25, kick := kMDV10907 ;
MDH11007.F : HKICKER, l = 0.25, kick := kMDH11007 ;
MDV11107.F : VKICKER, l = 0.25, kick := kMDV11107 ;
MDH11207.F : HKICKER, l = 0.25, kick := kMDH11207 ;
MDV11307.F : VKICKER, l = 0.25, kick := kMDV11307 ;
QE11402.F : QUADRUPOLE, l = 0.698, k1 := kQE11402 ;
MDH11407.F : HKICKER, l = 0.25, kick := kMDH11407 ;
TIDP : RCOLLIMATOR, l = 4.3 ;
MDV11507.F : VKICKER, l = 0.25, kick := kMDV11507 ;
BCT : INSTRUMENT, l = 0.694 ;
MDH11605.F : HKICKER, l = 0.25, kick := kMDH11605 ;
QF2A.F : QUADRUPOLE, l = 3.791, k1 := kQF2A ;
QMS.F : QUADRUPOLE, l = 0.705, k1 := kQMS ;
VVSB : INSTRUMENT, l = 0.175 ;
MKQH11653.F : HKICKER, l = 0.96, kick := kMKQH11653 ;
VDBC : INSTRUMENT, l = 0.5 ;
BTV : INSTRUMENT, l = 0.45 ;
VZBA : INSTRUMENT, l = 1.875 ;
TBSM : RCOLLIMATOR, l = 2.0 ;
MKQV11679.F : VKICKER, l = 1.416, kick := kMKQV11679 ;
MDV11705.F : VKICKER, l = 0.25, kick := kMDV11705 ;
```

Import Wizard Dialog:

Format: Layout

- LHC_FESA
- LHC_SM18
- LHC_SM18_2
- LINAC3
- NORTH_EXTRACTION
- PS
- PS_FESA
- SPS

Buttons: Import, Quit

Output Server:

```
log4j: WARN No appenders could be found for logger (org.springframework.core.CollectionFactory).
log4j: WARN Please initialize the log4j system property.
Caching is ENABLED
Initializing desc
```

An example of OM usage – checking out magnet settings from LSA

The screenshot displays the OM server v0.0.3 interface. The main window shows a file tree on the left with 'Decks' and 'Scripts' folders. The 'Data' tab is active, showing a list of magnet settings for 'ec.strengths.str'. An 'Import Wizard' dialog is open, showing a list of magnets and a 'time interval' field set to 14400.

Repository Data

- sps
 - Decks
 - SFT-LHCMD_L14400_PDC
 - SPS.TH.elements.elx
 - SPS.TH.sequence.seq
 - SPS.TH.strengths.str
 - SPS.elements.elx
 - SPS.sequence.seq
 - SPS.strengths.str
 - TT2.elements.elx
 - TT2.sequence.seq
 - TT2.strengths.str
 - Scripts
 - aperture_db
 - bootstrap.madx
 - knob_example.py
 - knob_mad
 - correct-orbit.madx
 - correct-orbit.py
 - convert-orbit.py
 - test.madx
 - tracking
 - tracking_py
 - twiss
 - twiss_aper
 - twiss-ideal
 - orbit-bump.madx

ec.strengths.str Plot 0

```
KMDH21207 := 0.0 ;
KMDH53407 := 0.0 ;
KMDH53207 := 0.0 ;
KMDH11407 := 0.0 ;
KMDH42607 := 0.0 ;
KMDH10207 := 0.0 ;
KLSFC := 0.03758934918 ;
KLSDB := -0.148004081 ;
KLSDA := -0.16359453969999999 ;
KLSFA := 0.03758934918 ;
KLSFB := 0.024089784200000006 ;
KQF1 := 0.014494553849022075 ;
KQD := -0.01458976586051024 ;
KQE60302 := 0.0 ;
KQE60502 := 0.0 ;
KQE11402 := 0.0 ;
KLQSA := 0.0 ;
KQF2 := 0.014733763849022075 ;
KQSE51897 := 0.0 ;
KBBLR5177M := 0.0 ;
KBBLR5176M := 0.0 ;
KMDV41307 := 0.0 ;
KMDV31507 := 0.0 ;
KMDV42107 := 0.0 ;
KMDVA41932 := 0.0 ;
KMDV60507 := 0.0 ;
KMDV62907 := 0.0 ;
KMDV31107 := 0.0 ;
KMDV63507 := 0.0 ;
KMDV13307 := 0.0 ;
KMDV60907 := 0.0 ;
KMDV30907 := 0.0 ;
KMDV51907 := 0.0 ;
KMDV50307 := 0.0 ;
KMDV62107 := 0.0 ;
KMDV21307 := 0.0 ;
KMDV52107 := 0.0 ;
KMDV22907 := 0.0 ;
KMDV31907 := 0.0 ;
KMDV41107 := 0.0 ;
KMDV53307 := 0.0 ;
KMDV32507 := 0.0 ;
KMDV40907 := 0.0 ;
KMDV51307 := 0.0 ;
KMDVB51777 := 0.0 ;
KMDV23107 := 0.0 ;
KMDV10107 := 0.0 ;
KMDV62307 := 0.0 ;
KMDV20707 := 0.0 ;
KMDV50107 := 0.0 ;
KMDV23307 := 0.0 ;
```

Import Wizard

Format: Settings

LEIR	CNGS12sV1
LEIR_FESA	LHC_INDIV_BI
LHC	CNGS2_SC_V1
LHC_FESA	FT-MDLHC_L14400V1
LHC_SM18	CNGS12sV1
LHC_SM18_2	FT-MDLHC_L14400V1
LINAC3	SFTPRO-CNGSV1
NORTH_EXTRACTION	SFTPRO-CNGSV1
PS	SPS.USER.LHC25NS_SC_V1
PS_FESA	SPS.USER.MD1_SC_V1
SPS	SPS.USER.MD2_SC_V1
SPS_FESA	NON-MULTIPLEXED-SPS
TI2	SPS.USER.SFTPRO1_SC_V1
TI8	SPS.USER.SFTPRO2_SC_V1
TT10	
TT2	
TT21	
TT22	
TT23	

time interval : 0 14400

Import Quit

Output Server

Caching is ENABLED
Initializing desc

An example of OM usage – creating an orbit bump

The screenshot displays the OM server v0.0.3 web interface. The browser address bar shows the URL: `OM server v0.0.3 - /afs/cern.ch/eng/sl/online/om/repository/sps/repo.xml`. The interface is divided into several sections:

- Repository Browser:** A tree view on the left shows a hierarchy of files. The 'Scripts' folder is expanded, listing various files such as `aperture_db`, `bootstrap.madx`, `knob_example.py`, `knob_mad`, `correct-orbit.madx`, `correct-orbit.py`, `convert-orbit.py`, `test.madx`, `tracking`, `tracking.py`, `twiss`, `twiss_aper`, `twiss-ideal`, `orbit-bump.madx`, `orbit-bump.py` (highlighted), `tune-knob.madx`, `tune-knob.py`, `bootstrap.py`, and `test.py`.
- Code Editor:** The main area displays the content of `ts/orbit-bump.py`. The code includes comments and Python code for generating an orbit bump. Key parts of the code include:

```
#!/usr/bin/python
#
# script to create orbit bumps
#
# place matching parameters here
variables = ['kMPLH41672', 'kMPLH41994', 'kMPSH42198']
# format - sequence, place, parameter, value
targets = [['sps', 'BPCE.41801', 'x', '35.e-3'], ['sps', 'QF.42210', 'x', '0'], ['sps', 'QF.42210', 'px', '0']]

outf = "test_bump.knob"

# code
import os

base = '/afs/cern.ch/eng/sl/online/om/repository/sps/scripts/'
madx = '/afs/cern.ch/eng/sl/online/om/madx/linux/madx'

# if the configuration 'message' file present read it in
#
# produce madx file
#
tmp_madx_file = base + '_tmp_bump_matching.madx'
tmp_out_file = base + '_tmp_bump_matching.out'
tfs_file = base + "mytab.tfs"

# add defaults
temp = open(base + "template.madx", "r").read()
#temp = temp.replace('__OPTICS_PATH__', './decks')

f = open(tmp_madx_file, "w")

f.write('! bump matching job')
f.write('! generated by the online model')

f.write(temp)

# match target values
```
- Output/Server:** A panel at the bottom shows the status of the server, with the text "Initializing desc".

An example of OM usage – creating an orbit bump

The screenshot displays the OM server v0.0.3 interface. The main window shows a plot of X versus S. The x-axis (S) ranges from 0 to 6000, and the y-axis (X) ranges from 0 to 0.1. The plot shows a series of vertical lines representing the aperture, with a sharp bump at S=4000. The bump is characterized by a sharp increase in X, reaching approximately 0.1, followed by a sharp decrease. The plot is titled "Plot 0" and the file is "LSS4-extrbmp.tfs".

The interface includes a menu bar (File, Edit, Tools, Repository (SVN), Help) and a toolbar with "run" and "Update" buttons. A left sidebar shows a tree view of the repository structure, including folders like "MADX tfs", "Knobs", and "BPM data", and files like "tune.tfs", "orbit_x.tfs", "orbit_y.tfs", etc. The bottom panel shows the "Output" window with the following text:

```
wrong tfs format for apertures  
ERROR: number of columns 10 does not match the number of keys 9  
loaded table tmp/aperture.tfs  
wrong tfs format for apertures  
adding aperture length 984
```

The bottom status bar shows the system tray with various icons and the taskbar with several open applications, including "Writable", "Smart Insert", "1722 : 1", "Java - om/src/jav", and "OM server v0.0.3".

An example of OM usage – creating an orbit bump

The screenshot displays the OM server v0.0.3 interface. The main window shows a repository tree on the left with 'test_bump.knob' selected. The main area displays the knob's data: kmp1h41672 0.000516969495, kmp1h41994 0.0003786611558, and kmpsh42198 0.0001549141292. A 'Knob Creator' dialog is open, showing the 'Create Knob' form. The 'SYSTEMS' field is set to 'CORRECTORS H', 'PARAMETER TYPE' is 'KNOB', and 'DEVICE NAME' is 'SPSBEAM'. The 'OPTICS' list includes 'EastExtraction-FT-2006v1', 'EastExtraction-FT-2007v1', 'EastExtraction-LHC-2006v1', 'EastExtraction-LHC-2007v1', 'LHC B1 Transfer-2007v1', 'LHC B2 Transfer-13/02/06', 'LHC B2 Transfer-18/03/06', 'NORMAL04', 'NorthExtraction-FT-2006v1', 'NorthExtraction-FT-2007v1', and 'SECTORTEST'. The 'KNOB NAME' table shows three entries: logical.MPLH4167... with value 5.16969495E-4, logical.MPLH4199... with value 3.786611558E-4, and logical.MPSH4219... with value 1.549141292E-4. A 'Save knob in DB' button is visible at the bottom of the dialog. The 'Output' pane at the bottom shows 'loaded knob tmp/test_bump.knob'.

OM server v0.0.3 - /afs/cern.ch/eng/sl/online/om/repository/sps/repo.xml

File Edit Tools Repository (SVN) Help

run Update

Repository Data

Root

- MADX tfs
- Knobs
 - SPSBEAM-QPH.knob
 - EAST_EXTRBEAM-LSS4_VTRI
 - test_bump.knob
 - NORTH_EXTRBEAM-LSS2_EX
 - test_tune_knob.knob
 - EAST_EXTRBEAM-LSS4_EXTR
 - NORTH_EXTRBEAM-LSS2_HT
 - EAST_EXTRBEAM-LSS4_HTRI
- BPM data

test_bump.knob Plot 0

```
kmp1h41672 0.000516969495
kmp1h41994 0.0003786611558
kmpsh42198 0.0001549141292
```

Knob Creator

SPS

Create Knob

SYSTEMS	
CORRECTORS H	

PARAMETER TYPE: KNOB

DEVICE NAME: SPSBEAM

OPTICS

- EastExtraction-FT-2006v1
- EastExtraction-FT-2007v1
- EastExtraction-LHC-2006v1
- EastExtraction-LHC-2007v1
- LHC B1 Transfer-2007v1
- LHC B2 Transfer-13/02/06
- LHC B2 Transfer-18/03/06
- NORMAL04
- NorthExtraction-FT-2006v1
- NorthExtraction-FT-2007v1
- SECTORTEST

KNOB NAME	
test-knob	
Parameter name	Value
logical.MPLH4167...	5.16969495E-4
logical.MPLH4199...	3.786611558E-4
logical.MPSH4219...	1.549141292E-4

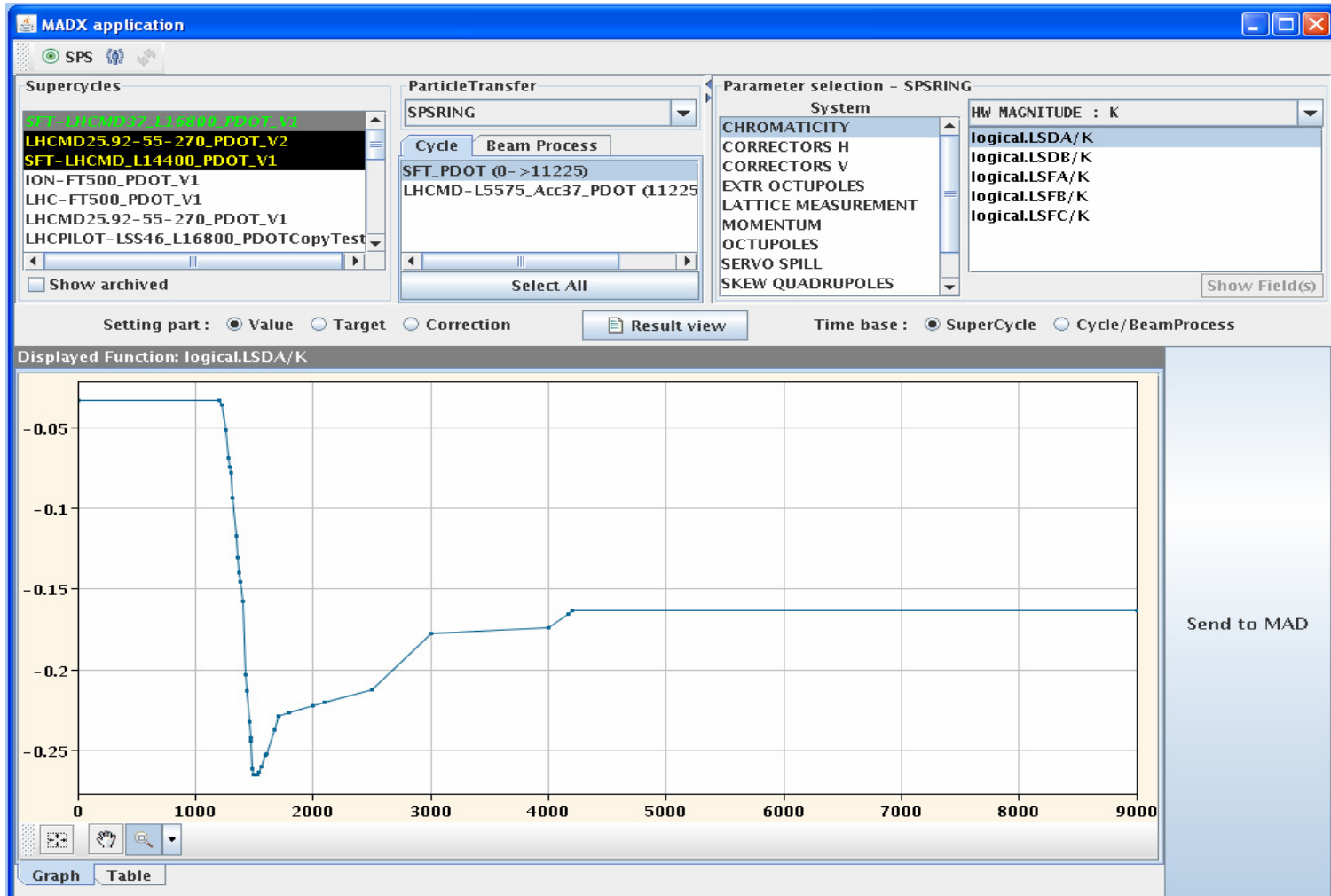
Save knob in DB

Output Server

loaded knob tmp/test_bump.knob

Taskbar: [Inbox for Ilya...], [iagapov@abpc], [iagapov@abpc], [Аквариум - 2], [Informations s], [Minutes of the], iagapov@abpc, Java - om/src/j, OM server v0.0.3, Knob Creator

An example of OM usage – ‘trim dummy’ prototype



Prototype application; tests in progress

LHC Stage A: Commissioning phases

Phases for full commissioning Stage A (pilot physics run)

Phase	Description
A.1	Injection and first turn: injection commissioning; threading, commissioning beam instrumentation.
A.2	Circulating pilot: establish circulating beam, closed orbit, tunes, RF capture
A.3	450 GeV initial commissioning: initial commissioning of beam instrumentation, beam dump
A.4	450 GeV optics: beta beating, dispersion, coupling, non-linear field quality, aperture
A.5	Increasing intensity: prepare the LHC for unsafe beam
A.6	Two beam operation - colliding beams at 450 GeV
A.7	Snap-back and ramp: single beam
A.8	Bringing beams into collision: adjustment and luminosity measurement
A.9	7 TeV optics: beta beating, dispersion, coupling, non-linear field quality, aperture
A.10	Squeeze: commissioning the betatron squeeze in all IP's
A.11	Physics runs: physics with partially squeezed beams, no crossing in IP1 and IP5

From LHCCWG web pages

Stage A.1: procedures and tools availability

Procedure	Activity	Candidate OM tools	Dupl.	Need
A.1.1 Commission injection region (steering onto TDI) beam 2	Check transfer line stability	Beam matrix measurement	y?	?
	Optics matching	?	?	?
A.1.2 Threading around the ring – beam 2	Trajectory correction (cross-check)	Optics model Orbit correction (YASP x-check)	n y	? ?
	Corrector/BPM polarity checks	Optics model		
	Linear optics checks (kick/response)	Optics model		
	Aperture checks with free oscillations and sliding bumps	Tracking; checking aperture model + optics model + BLM data for consistency orbit bump script	?	y?
	Energy matching (optional)	Energy matching	y?	?
A.1.3 - same as A.1.1 for beam 1				
A.1.4 - same as A.1.2 for beam 1				
A.1.5 interleaved threading	Similar to A.1.1 – A.1.4			

Stage A.2: procedures and tools availability

Procedure	Activity	Candidate OM tools	Dupl.	Need
A.2.1 Instrumentation and corrector checks				
A.2.2 Establish closed orbit – both beams	Close the trajectory			
	Closed orbit correction to get few turns	Orbit correction	y	?
	Aperture scans	See A.1.2		
A.2.3 Measurements with a few turns	Integer and fractional tune measurements	Tune measurement	y	?
	Adjustment of chromaticity	Input to the optics model; knobs		
	Adjustment of coupling	Input to the optics model; knobs		
A.2.4 Offsets between different sectors	Orbit correction	Orbit correction	y	?
	Correction of Bdl			
	Correction of MQ-MQ offsets			
A.2.5 RF observation equipment				
A.2.6 SPS-LHC energy matching	Match Bdl(SPS) Bdl(LHC1) Bdl(LHC2) f(RF)	Energy matching	?	y
A.2.7 Synchro loop commissioning				
A.2.8 Beam capture				
A.2.9 Measurements with captured beam		Similar to A.2.3		

Stage A.3: procedures and tools availability

Procedure	Activity	Candidate OM tools	Du pl.	Need
A.3.1 Final RF commissioning with pilot				
A.3.2 BPM checks with pilot	Calibrate BPMs and correct orbit	Orbit correction	y	?
	Acquire trajectory data	Tune/optics measurement		
A.3.3 First commissioning of beam dumping (pilot)	Set TCDQ to 10σ	optics model Optics measurement	?	?
	Aperture measurements	Optics model		
A.3.4 Commission systems with higher intensity	TDI setting up	Optics model Optics measurement		
	Tune meter and chromaticity measurement	Tune/optics measurement optics model; tracking		
	Initial BLM commissioning	Aperture model; bumps		
A.3.5 Establish cycling machine	Verify reproducibility			
A.3.6 Lifetime optimisation – get to 1h	Adjust chromaticity, tune, orbit, coupling	Bumps and knobs	n	y

Stage A.3: procedures and tools availability (continued)

Procedure	Activity	Candidate OM tools	Dupl.	Need
A.3.7 Further commissioning of beam instrumentation with >1h	Systematic BPM and corrector calibration	Use response data for optics fits		?
		Optics model -> LOCO	?	?
		Orbit correction (YASP x-check)	y	?
	Commission PLL tune and coupling measurement	Tune/optics measurement optics model; tracking		
	Commission wire scanners	Beam matrix; optics model	?	?
	SR and rest gas monitors			
A.3.8 Basic optics checks in addition to LOCO	Harmonic analysis of multi-turn data	Tune/optics	?	
		Driving terms	?	y
		Measure emittance	?	?
	RF tuning			
A.3.9 Further commissioning of beam dumping system	Aperture measurements	Optics model		
A.3.10 Commission feedback systems	Chromaticity measurement; orbit feedback; tune and coupling feedback; transverse feedback	Optics model		
A.3.11 Rough setting up of TDI		See A.3.4		

Stage A.4: procedures and tools availability

Procedure	Activity	Candidate OM tools	Dupl.	Need
A.4.1 Measurement and correction of the closed orbit	Orbit correction	Orbit correction	y	y
A.4.2 Measurement and correction of linear optics	Detailed tune measurement	Optics measurement (x-check) Chromaticity tuning		
	coupling	Tune knobs Coupling correction		
	Beta-beat Dispersion	Optics model		
	Refined optics model -> YASP etc.	Response matrix		
	Beta measurements with k-modulations	Input to optics model		
A.4.3 Measurement and correction of aperture	Aperture measurement with kick+scrape	Tracking; checking aperture model + optics model + BLM data for consistency		
	Orbit centring into aperture	bumps		
A.4.4 Detailed RF measurements				
A.4.5 Measurement of the momentum aperture		Input to optics & aperture model		
A.4.6 Collimators and protection devices	Measurement of beta-functions and beam sizes	Beam matrix; optics model	?	?
	MORE INPUT NEEDED	Tracking? Loss maps? MORE INPUT NEEDED		

Stage A.4: procedures and tools availability (continued)

Procedure	Activity	Candidate OM tools	Dupl.	Need
A.4.7 Measurement of the global nonlinear optics	Tune versus dp/p , q' , q'' etc.	Input to optics model	y	y
A.4.8 Commissioning of non-linear correctors	Polarity checks	Driving term application	?	?
A.4.9 Measurement and correction of aperture	Separation bumps	?		
	Spectrometer compensation IR2, IR8	?		
A.4.10 Detailed injection matching	Not yet defined	Optics matching?		
A.4.11 Detailed beam loss studies	Beam loss maps	Beam loss maps?		

OM applications availability summary (A.1 – A.4)

Phase	Tools
A.1 (Injection and first turn)	bumps and knobs; orbit correction; tracking; energy matching Beam matrix (TI8)
A.2 (Circulating pilot)	bumps and knobs; orbit correction; tracking; energy matching Tune/optics measurement
A.3 (450 GeV initial commissioning)	bumps and knobs; orbit correction; tracking; energy matching Beam matrix; driving terms
A.4 (450 GeV optics)	bumps and knobs; orbit correction; tracking; Beam matrix; driving terms Coupling correction beam loss maps

Color code: **developed** (!= successfully tested) **under development** **foreseen** proposed

COMMENTS :

In the first commissioning stages we mainly should:

- Provide a relevant up-to-date optics model. Requires up-to-date LSA DB. Optics model tuning for LHC more complicated than SPS (further studies required)
- Test tools
- Understand bottlenecks of linear optics model

Outstanding technical issues:

- Do we need to define procedures to transfer response matrix from OM to YASP, LOCO etc. or will it be done manually by Jorg?
- Do we need to transfer optics tables produced by OM into LSA? LSA has only 'nominal' optics tables. (Need to discuss with Mike & Co.)

Proposed focus:

- Beam matrix and optics fits with little data available (early commissioning)
- Strategy for fitting the madx optics model for LHC