THE ONLINE MODEL FOR THE LARGE HADRON COLLIDER

C. Alabau Pons, X. Buffat, M. Giovannozzi, G. Müller, S. Redaelli, K. Fuchsberger,
M. Lamont, F. Schmidt, CERN, Geneva, Switzerland

Abstract

The control of the high intensity beams of the CERN Large Hadron Collider is particular challenging and requires a good modeling of the machine. In recent years efforts were devoted to the design of a software infrastructure aimed at mimicking the behavior of the LHC. An online model of the machine, based on the accelerator design tool MAD-X, has been developed to support the commissioning and the operation of the LHC. This model is integrated into the Java-based LHC development framework and provides the full computing power of MAD-X, including the best knowledge of the machine aperture and magnetic models. In this paper, we present the status of the MAD-X online application and illustrate how it has been used during the LHC commissioning. Possible future implementations are also discussed.

INTRODUCTION

The aim of the online model (OM) for the LHC is to bring into the control room the full functionality of MAD-X [1] for online accelerator physics computation and the most updated model of the machine, including mechanical aperture, optics imperfections and magnetic models. While the detailed description of the accelerator model used is beyond the scope of this paper (see [2–4]), we focus here on the implementation of the OM manager application.

The first implementation of the OM manager [5] was based on interaction with MAD-X through single script execution enhanced by script generation, sequential script calling and modeling in Python. This approach provided a fast and flexible implementation that made the MAD-X computation power available in the control room since the beginning of the LHC commissioning. It was used extensively during transfer line commissioning, LHC sector tests and first circulating beam commissioning.

On the other hand, the LHC Software Architecture (LSA) [6] used for settings management, equipment control and beam measurements is almost entirely implemented in Java. The script based implementation, while providing the full computation power of MAD-X, was essentially decoupled from the core LSA implementation (settings, databases, trims, ...). It became therefore apparent that to improve and integrate the application more closely within the LSA environment and to open the ground for further development, a refined interface to LSA was needed. In addition, the requirement to handle some hundreds of different LHC optics and more and more demanding OM calculations imposed a revision of the MAD-X run calls.

MAD-X and its scripting language is a well known and established working environment in the field of accelerator design and has to be kept as part of the features of the OM to allow users to create and/or import their own MAD-X scripts. On the other hand, the OM needs to enhance its functionality towards data exchange at the application level and interaction with MAD-X in an object-oriented fashion. To this end the OM takes advantage of the new JMAD (Java API for MAD-X) implementation [7].

OM IMPLEMENTATION

User Repositories

The script repository management was reviewed to allow the user to either assign a local working directory or request a workspace on a dedicated AFS space which allows multiple users to share scripts among each other. A repository itself is nothing else than a directory structure saved in xml, which contains for each file:

- the position in the repository structure
- the origin (private file or imported from another user)
- status of the file (edited or original)
- the current location in the file system as well as
- the location of the original to revert local changes

A core repository with scripts for standard and basic MAD-X tasks is provided as a default and can be easily extended by the users to assemble and save their own repositories.

Improved interface to LSA

LSA, written in Java and based on Spring [6], provides a framework to:

- manage optics and machine layout definitions;
- generate, alter and drive settings for the required beam configurations;
- provide abstracted access to the underlying hardware levels;
- access beam-based measurements through Java API for Parameter Control (JAPC).

Key concepts of LSA are: *parameters*, *settings* and *contexts*. Parameters are organized in hierarchical dependencies that link top-level beam or accelerator quantities (momentum, tune, chromaticity, ...) to settings for the underlying equipment, e.g. currents in the magnet power converters (PCs) as well as collimator jaw positions. Top-level parameters in this hierarchy are so-called *knobs*, for example a tune knob has dependencies down to the different tune corrector circuits in the various powering sectors.

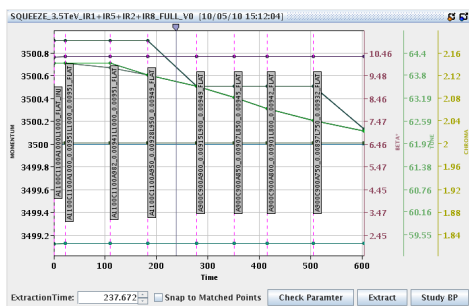01 Circular Colliders

A01 Hadron Colliders

Figure 1: Example of beam process information display for a betatron squeeze in all the interaction points.
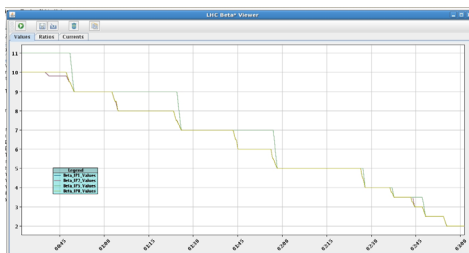

Figure 2: OM display of beta functions in the various interaction regions, based on the ratio of magnet currents during the squeeze [8].

LHC parameter settings are generated and collected in contexts for different machine conditions, such as injection, ramp, squeeze, etc. For the LHC a context is called *beam process* (BP) and can be functional or discrete (*actual*). For example, the energy ramp is driven by functions with well defined time length, whereas the machine has to sit at injection level for an undefined amount of time while the beams are injected, which is achieved with discrete settings.

It is important to note that setting values within a given machine context depend on the machine optics calculated by MAD-X taking as an input magnet strength files. The information about the optics are stored in appropriate database tables which are then used in the setting generation process. For example, the energy ramp is done with optics at constant energy, hence the generated magnet current functions follow a simple scaling with energy contrary to the betatron squeeze, for which the optics are changing as a function of time and the generated functions are therefore more complex.

The MAD-X OM manager features now a number of interfaces, based on the core LSA packages, to retrieve all relevant information for the accelerator operation, such as optics information (twiss functions, tune, chromaticity, ...), available parameter definitions and their settings for each BP. This is done using different sources: nominal database tables used for setting generation as well as real settings, possibly with corrections, to drive the machine. Both can be used as an input for OM MAD-X calculations and have been extensively used prior to the beam commissioning to validate the settings generated within LSA against MAD-X simulations.
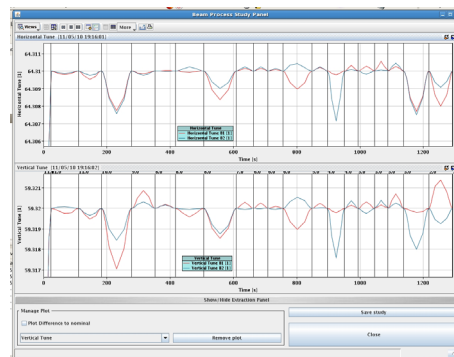

Figure 3: Evolution of horizontal and vertical tunes for both beams during a squeeze BP. The calculation is performed with MAD-X taking as an input the time-dependent LSA settings for the power converters.

Figure 1 shows the BP information panel in the OM manager with the optics (gray boxes) that are used within a beam process. This allows to visualize the time dependent variation of key parameters such as energy, β functions at the IPs, tunes and chromaticities. This information can be directly compared with the OM display of the beta functions in the IPs, which is provided within the manager (Fig. 2) as well. The interactive panel allows to extract settings at any time in the machine cycle in order to load machine settings, compare them with their nominal values or to convert them into MAD-X format for OM calculations. A special panel is also available to perform simulations using setting extractions as a function of time and to assess the evolution of key parameters. This is illustrated in Fig. 3, where the tune evolution of both beams as simulated by MAD-X is given over the time of a squeeze BP. These data are presently bench-marked against the results from the first squeeze commissioning experience.

Optics and knobs

The MAD-X OM manager is now used to import into LSA all the optics required for the beam operation as well as the configuration for key beam knobs (tune, chromaticity, coupling, separation, crossing, ...). In order to handle the many LHC optics (more than 150 optics are available to cover the squeeze in all IPs, each of which requires some 30 knob definitions), a model definition editor gives the possibility to create and alter xml-based JMAD model definitions to specify machine and optics. These definitions are then used to initialize JMAD before computing the twiss functions or key beam knobs for all defined optics.

During optics generation tunes and chromaticities are automatically checked against references and re-matched if necessary. Knobs are created by matching jobs and each of them is normalized and trimmed in simulation before importing them to LSA. An example of the separation knob schemes is given in Fig. 4. The knob applications have been used extensively during beam commissioning and validations performed prior to the beam tests were certainly an important ingredient of the smooth LHC commissioning.
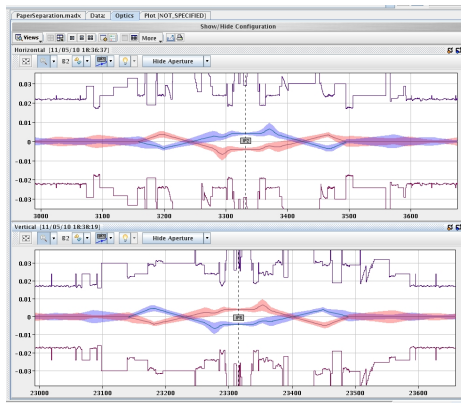
Figure 4: Separated beam orbits and 3 σ envelope in IP2 (ALICE - top) and IP8 (LHCb - bottom).
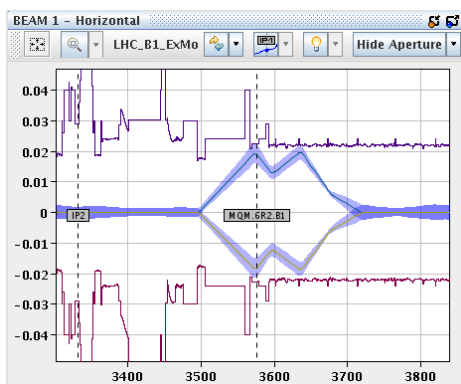

Figure 5: Simulated maximum orbit excursions during aperture measurements with local bumps.

Machine OM

The OM manager allows to extract LSA settings and to use them as JMAD input for OM calculations based on the actual machine settings. One can therefore test how the LHC performance varies with changes in the machine settings. To this end settings are varied and the calculated difference with respect to reference settings is added to the ideal LHC optics.

Dedicated software has been developed to handle the different data extraction sets: *containers* and *filters* allow to structure the data sets, export them into MAD-X format and re-play the JMAD simulations for a given set. Fig. 5 shows an example, where a local bump used for aperture measurements [2] is compared with the detailed aperture model. The beam envelope is computed by taking into account measured emittance and β function [3].

The approach of using the difference settings with respect to a reference is not the most accurate one but is probably the best that can be done until a complete non-linear model of the LHC is established. The present OM implementation proved to be useful in the on-going LHC commissioning.

CONCLUSION AND OUTLOOK

The LHC OM manager is now fully integrated in the standard LHC Software environment and has been extensively used during the LHC commissioning, without beam (settings verification, optics import, knob generation and validation) as well as with beam (dedicated measurements, display of real beam envelope, OM optics computation).

The present implementation features interfaces to the key LSA components and this opens a great potential for development of applications to continue to support the LHC commissioning and operation. In particular, we are presently i) extending further the OM verification of settings (simulate trims in the model and adding sequencer tasks to verify fill-to-fill setting changes as well as incorporated settings); ii) developing an "aperture meter" to monitor continuously the available aperture during the operation of unsafe beams; iii) automatizing OM analysis for a number of studies, e.g. to combine orbit, beam loss and wire scanner data during the aperture measurements; iv) extending the interface to beam-based measurements (multi-turn orbit acquisition, β functions from β-beat measurements, *post-mortem* buffers, power converter currents); v) improving the aperture model with measured collimator jaw positions; vi) providing the above functionality on a server-based implementation for other application clients.

It goes without saying that the OM will continue to follow the evolution of the optics model of the LHC to have the best model of the machine during all LHC commissioning phases.

ACKNOWLEDGEMENTS

REFERENCES

[1] MAD-X Webpage, www.cern.ch/mad

[2] S. Redaelli *et al*., "LHC Aperture Measurements", these proceedings.

[3] R. Tomás *et al*., "LHC Optics Model Measurements and Corrections", these proceedings.

[4] E. Todesco *et al*., "The Magnetic Model of the LHC in the Early Phase of Beam Commissioning", these proceedings.

[5] I. Agapov *et al*., "LHC On-Line Model", PAC'07, Albuquerque, New Mexico, USA, June 2007, THPAN071

[6] G. Kruk *et al*., "LHC Software Architecture [LSA] Evolution toward LHC Beam Commissioning", ICALEPCS'07, Knoxville, USA, October 2007, WOPA03, p. 307

[7] K. Fuchsberger *et al*., "JMad - Integration of Madx into the Java World", these proceedings.

[8] S. Redaelli *et al*., "Betatron squeeze," LHC Evian workshop (2010).