EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH
European Laboratory for Particle Physics

Large Hadron Collider Project

LHC Project Report 1017

LHC ON-LINE MODEL

I. Agapov, W. Herrr, G. Kruk, M. Lamont, F. Schmidt
CERN, Geneva, Switzerland

Abstract

The LHC machine will be a very demanding accelerator from a beam control perspective. There are tight constraints on the key beam parameters in the presence of large non-linearities and dynamic persistent current effects. Particle loss in the LHC must be actively minimized to avoid damage to the machine. Therefore any adjustment to the machine parameters would ideally be checked beforehand with a proper modeling tool.

The LHC On-Line Model is an attempt to provide such an analysis tool based mainly on the MAD-X code. The goal is not to provide a real-time interactive system to control the LHC, but rather a way to speed up interaction with the power of MAD-X and to facilitate off-line analysis to give results within appropriate time constraints. There will be a rich spectrum of potential applications such as

closed orbit correction, beta-beating analysis, optimization of non-linear correction and knob settings.

We report the status of the on-line model software which is at present being developed for the beginning of the LHC commissioning.

Presented at
PAC07, 22nd PAC Conference, June 25-29, 2007,
Albuquerque, USA

CERN,
CH-1211 Geneva 23
Switzerland

LHC ON-LINE MODEL

I. Agapov, W. Herr, G .Kruk, M. Lamont, F. Schmidt, CERN, Geneva, Switzerland

Abstract

The LHC machine will be a very demanding accelerator from a beam control perspective. There are tight constraints on the key beam parameters in the presence of large non-linearities and dynamic persistent current effects. Particle loss in the LHC must be actively minimized to avoid damage to the machine. Therefore any adjustment to the machine parameters would ideally be checked beforehand with a proper modeling tool.

The LHC On-Line Model is an attempt to provide such an analysis tool based mainly on the MAD-X code. The goal is not to provide a real-time interactive system to control the LHC, but rather a way to speed up interaction with the power of MAD-X and to facilitate off-line analysis to give results within appropriate time constraints. There will be a rich spectrum of potential applications such as closed orbit correction, beta-beating analysis, optimization of non-linear correction and knob settings.

We report the status of the on-line model software which is at present being developed for the beginning of the LHC commissioning.

INTRODUCTION

The destructive power of the LHC beam operating in a mainly superconducting environment places stringent demands on beam parameter control at the LHC. This is in the presence of serious persistent current effects and strong non-linearities. It is clear that an accurate optics model of the machine will be essential and instrumental in uncovering sources of mis-matches, beta-beating, transfer function errors etc. Proper control of these effects is vital in maintaining the aperture of the machine and thus effective collimation, essential for safe operation.

An on-line model will be useful for a number of reasons: adjustments to the machine must be checked beforehand with a proper modeling tool; trims made to the machine to correct measured errors can be exported to the On-line Model to attempt to maintain an up-to-date model; beam based measurements should be easily exportable for analysis (orbit correction for example).

The main aim is to harness the power of MAD-X [1] easily from the control system. Currently, the LHC On-line Model (OM) is being developed to provide such functionality.

The on-line model is aimed mainly at evaluating potential settings and providing simulations in the control room environment rather that monitoring the current machine status. It should be mentioned that certain variants of on-line modeling software have been also implemented in other accelerator laboratories [3, 4].

SOFTWARE REQUIREMENTS AND ARCHITECTURE

General requirements

As discussed in the introduction, there are many potential application of a on-line model and one might expect the software to meet the following general requirements:

1. Provide an accurate and up-to-date snapshot of the machine to be made available as input into either off-line or on-line evaluation.
2. Allows one to quickly perform standard optics calculations and to visualize the results.
3. Evaluate the effect of parameter adjustments before they are applied to the machine.
4. Consolidate changes made to the machine with the off-line version of the machine optics. There are potentially many different optics.
5. Provide the ability to introduce trims calculated by MAD-X into the control system for application to the machine.
6. Interface with the magnetic model of the machine.
7. Provide an aperture model for the whole machine.

In order to expose the required functionality, a number of Use Cases reflecting the key requirements enumerated above were analyzed.

Software architecture

One of the essential problems is communicating between the control system software and the on-line model. We need to be able to push proposed trims to the model for verification, push measurement data (e.g. closed orbit) to the model and pull back any proposed correction. We need to be able to pull from the model proposed corrections and apply them to the machine (e.g. quadrupoles trims, say, for the correction of beta-beating).

The LHC control software [2] is a Java based system providing a sophisticated settings management system. This includes a settings generation package which imports Twiss parameters and normalized magnets strengths from MAD-X, and as such makes available the ideal optics model. This model is used by the high level control system for basic parameter correction such as closed orbit control, tune and chromaticity adjustments.

For security reasons the on-line modeling software should be separated from the control applications. However, data exchange between the two should be quick and easy.

Loosely speaking the on-line modeling software will consist of the following subsystems:

- The model engine will run as a server providing simulation results to other software components.
- Accelerator model repository (mainly MAD-X input files).
- Simulation manager: A Java interface for configuration and execution of simulation programs (mainly MAD-X) and retrieving data.
- Repository management including version control.
- Exchange server: request exchange with the control system.
- A standard format and repository for beam based measurements which are written by the control system and read by the simulation manager or specialized off-line analysis programs.

A diagram of the conceptual software architecture design is shown in Fig. 1. More detailed description of subsystems is given below.

Most operations will be performed through a Java-based GUI (see Fig. 2). This is the main application which allows one to run scripts, browse and plot the resulting data, browse the control databases etc.

Repository management The machine model consists of layout, magnet settings, field errors, aperture settings. The on-line model repository accommodates machine models for accelerators of interest (LHC, SPS, transfer lines) and a set of scripts for typical jobs. One can switch between different repositories, e.g. LHC at collision, LHC at injection, TI10 transfer line etc. The version control system assures that the latest version of optics is available, changes can be reverted, machine snapshots in the past can be recovered etc. One should be able to import and export BPM readings to and from the model for comparison with the machine in various formats (SDDS [5] being the nominal). The LHC optics files are currently stored under CVS [6] version control. The on-line model is expected to undergo frequent changes so it requires a different version control system. At the moment a SVN-based [7] version control is implemented. An option to commit the current optics into CAMIF is foreseen. The repository managements system is a set of Java classes with underlying python scripts. A repository is a XML file providing references to accelerator layout, aperture and so on.

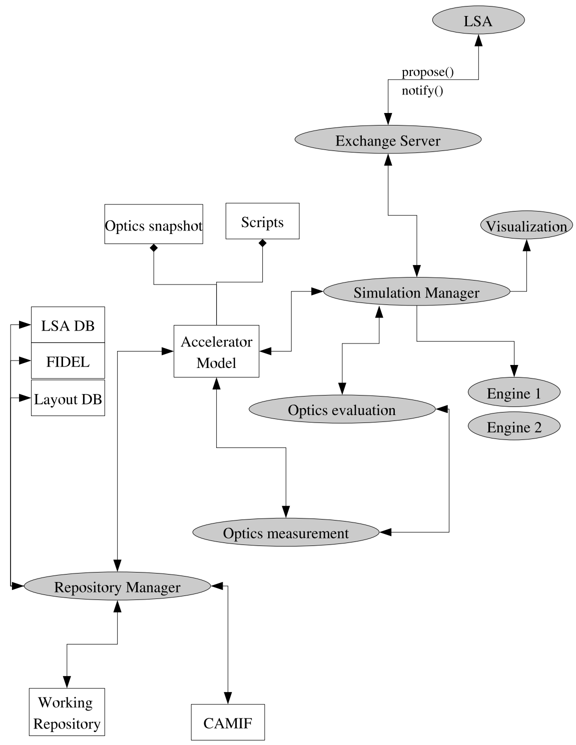

Figure 1: *Software Architecture Design*

Simulation Manager The simulation manager is a set of Java classes allowing to perform simulations and to retrieve results in different formats. Execution is performed via a call to a python wrapper of the simulation engine. Multiple simulation engines can be used. Any type of simulation program (e.g. a mixture of python scripts, MAD-X calls etc.) can be used as soon as it conforms to a certain output format which at the moment is TFS-table based. MAD-X will be used as the main simulation engine.

Optics evaluation It is crucial to have a set of tests for evaluating of how good the machine model represents the reality. These should be: simple tests to check if anything has changed in the machine from the last update and appropriate optics measurement and evaluation (see section below).

Cross system interfaces There are a number of systems intimately involved in the setting and configuration of the machine:

- LSA and its associated database for operational settings management.
- FiDel [8] and WISE [9] - for magnetic field model data.
- The LHC Layout database.

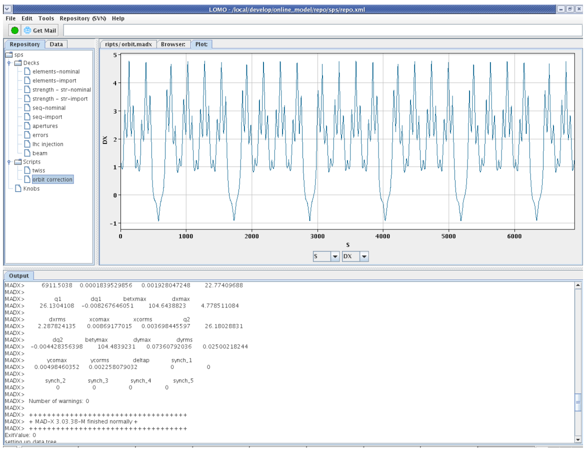

Figure 2: *Screen-Shot of Control Center*

Most of the LHC operation will be performed through the system based on the LSA architecture. From the control system front-end software it should be possible to send to the on-line model exchange server requests to check a certain optics, change or create a knob etc. The on-line model simulation manager receives the request and performs a sequence of actions as appropriate to meet the requirements outline above. The format of requests is based on XML.

OPTICS EVALUATION AND MODEL CORRECTION

In the beginning of the LHC commissioning the linear properties will be the main concern. Besides closed orbit issues, tune and phase adjustments and linear coupling corrections the measurement and minimization of beta-beating [10] will need an extensive study. In fact, a clear interface between on-line modeling and the beta-beating application is in the process of being defined.

On the other hand the LHC will be, from the start, a very non-linear machine which will require a full understanding and correction of nonlinear effects, in particular to maximize the dynamic aperture.

Once the linear imperfections and their adjustments are understood one has to deal with these non-linearities. To this end driving term measurements are envisaged: the measurements at the machines will be compared with LHC model simulations with MAD-X. In this way the non-linear LHC model that includes all measured multipole components can be compared and corrected to be in agreement with the true non-linear LHC. This corrected MAD-X model shall be used to calculate the corrector settings.

The SPS driving term experiments [11] were performed in view of demonstrating such a comparison. Fig. 3 (upper part) shows an almost perfect agreement between model and measurements of the third order resonance along the SPS. However, one failing sextupoles (lower part of Fig. 3) leads to substantial changes in the longitudinal driving term pattern. The interpretation of this resonance pattern in the case of the LHC will allow to locate longitudinal differences between model and the real LHC.

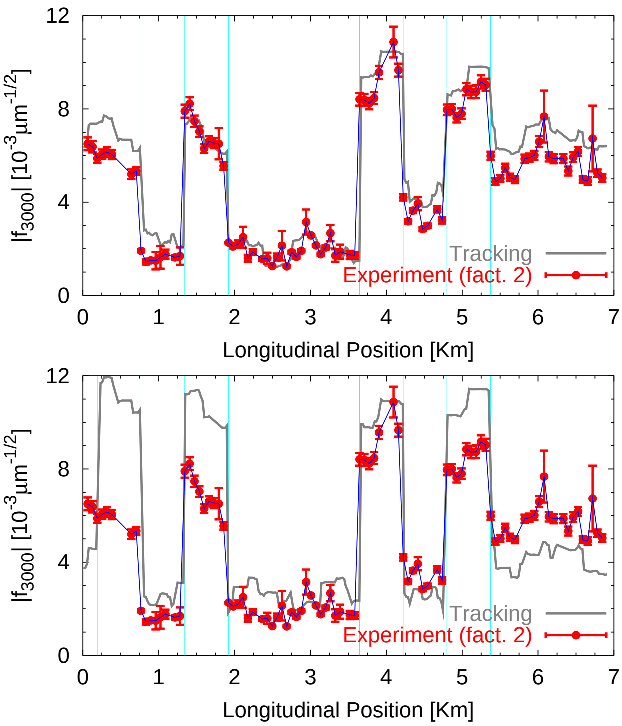

Figure 3: *Longitudinal Driving Term Pattern due to one failing Sextupole.*

PLANS

The beta version of the on-line modeling software should be ready for prototype tests at the SPS at the end of August 2007. We expect the on-line model software to be fully operational by November 2007 in time for the LHC commissioning. Possible further developments are collective effect evaluation, interaction region geometry and elements of machine-learning techniques.

REFERENCES

[1] http://cern.ch/mad/.

[2] http://ab-project-lsa.web.cern.ch/ab-project-lsa/.

[3] Y. Roblin et al., JLAB-ACE-01-10, 2001.

[4] T. Satogata et al., Proc. PAC 1999.

[5] http://www.aps.anl.gov/asd/oag/software.shtml.

[6] http://isscvs.cern.ch/cgi-bin/cvsweb.cgi/, CAMIF.

[7] http://subversion.tigris.org.

[8] http://fidel.web.cern.ch/fidel/.

[9] http://wise.web.cern.ch/WISE/.

[10] R. Calaga, et al., these proceedings.

[11] M. Benedikt et al., PRSTAB **10**, (2007).