Environment

The online modeling main purpose is the establish the connection between:

**Accelerator Physics Simulation** JMad [4] the JAVA API to MAD-x – the de facto standard accelerator lattice design and simulation software at CERN – is used to perform simulations from nominal and current settings of the machine.

**Control Infrastructure** LHC Software Architecture (LSA) [5] provides access to optics data, machine settings, online measurement data and the framework for settings generation for all LHC devices.

**Operation** Creation and verification of new operational configuration, visualization of current beam configuration and machine settings as well as simulation of setting changes.

Knobs and Optics

Large number of Optics (twiss) functions are required for the LSA settings generation for the LHC (see Table 1). High-level parameters – Knobs – are needed for the adjustment of key machine parameters like tune, chromaticity, coupling and orbit deviation. These are managed, created and uploaded by the Knob and Optics Manager (Fig. 1). Optics models are defined in the framework of JMad model definitions.

Setting Verification

After optics are uploaded to LSA, they are used to define a beamprocess as a sequence of optics that should be reached while the generated setting functions are driven to the machine. The generated settings are then checked with the online model manager which allows, beside other functionality to:

- plot and compare optics functions (Fig. 2)
- check correct definition of beamprocesses (Fig. 3)
- extract settings to MAD-X input and run MAD-X scripts

The following setting checks can be performed:

- extraction of full power converter settings and calculate the optics functions in MAD-X and compare to the nominal
- trim of knob in LSA followed by setting extraction and MAD-X run to check the knob shape
- scan over the settings of a beamprocess: the evolution of the optics parameters based on the generated settings is checked over the time of the beamprocess.

Architecture

Suitable project layout (see Fig. 4) and software management has been put in place for the toolchain:

- 8 core projects/packages, dedicated for each of the systems the online modeling has to interact with
- currently three main application which use the available packages
- build project for automated release of the toolchain and the applications
- continuous integration and code testing with BAMBOO [6] automatic build plans which execute JUNIT [7] test cases for core functionality

References