Elegant and ACL based trajectory tuning for FAST facility

(ElegantACL)

PERSONNEL

Jinhao Ruan (co-PI) AD/Fnal, Scientist, ruanjh@fnal.gov

Chip Edstrom (co-PI) AD/Fnal, Engineer Physicist, edstrom@fnal.gov

A high energy, high intensity beam has been commissioned in the FAST electron injector LINAC. Trajectory tuning is routinely performed applying a response matrix method through 6DSim, a program developed by Dr. Alexsandr Romanov. This provides a high degree of stability and repeatability for day to day operations, including start-up, beamline studies, and IOTA injection. 6Dsim was developed as a stand-alone program, relying on an expert knowledge of the program, specific privileges, mapped drives, and distributed resources, and it would benefit from full integration into the accelerator control system.

• *Scientific or technical motivation and purpose:* We would like to integrate trajectory tuning into ACNET using a method similar to 6Dsim. This would include interaction with the sequencer, including it as an option in normal turn-on and mode-switching procedures followed by qualified accelerator operators at the FAST facility. This would simplify beam tuning activities and dramatically increase machine study efficiency by streamlining the setup process.

• *Experimental methods:*

1. To fulfill the requirement of a real-time, beam-based tuning package, we will first establish an online simulation with a shell able to read and set various ACNET devices (e.g. BPMs and beamline trims respectively). We have chosen to use ELEGANT as our simulation software as it's been installed in CLX cluster on CLX33, and so is available to native ACNET programs. This would allow us to make use of lattice files already developed as a part of the design process of the FAST facility accelerators. ELEGANT also uses the same mechanism as 6Dsim for trajectory tuning, allowing us to compare the two packages. Initially, device reads and sets will be performed through the Accelerator Control Language (ACL), which is already used extensively through the sequencer to turn on or off sections of the

accelerator, and to perform operational mode changes. This could be replaced with Python or a C++ program in the future if necessary.

2. In the last several months, considerable effort has been dedicated to benchmarking the ELEGANT simulation against 6Dsim, working to resolve any discrepancies between the two. Once decent agreement was reached between the two codes, we established the lattice and the corresponding BPM readings from a datalogger to simulate the real-time tuning. This response was encouraging as well.

3. A real-time, beam-based test will be our next step.

• *Expected results*: We expect to develop a sequencer-based trajectory tuning package that will allow operators to rapidly and easily tune the beam back to a golden trajectory. This is complicated by operational factors, such as energy drift, thermal effects, and magnetic errors (e.g. through hysteresis). Note that the trajectory tuning method proposed here is only able to compensate for the magnetic errors and further study will be needed to better distinguish and mitigate other sources of error in our beamline. Beyond, we hope that this effort will lay a good foundation for further exploration, allowing for development of tools that make use of more sophisticated codes (e.g. neural networks) to perform more rigorous error compensation and offer greater operational flexibility.

BEAM CONDITIONS

Beam species: electrons

Intensity: < 250 pC (minimum depends on BPM sensitivity).

Energy: 41 MeV at entrance to CM2 and ~100 MeV initially at exit of CM2

Number of bunches: 10 bunches at most

Micropulse repetition Freq.: 3 MHz

Trajectory requirement: Try to use the golden trajectory establish during run 2 as our reference

APPARATUS

BPM: Need to be functional along the line. We are hoping to use all of them but can live with if just 1 or 2 is malfunctioning.

RUN PLAN

• Requested running period and approximate duration: 3 shifts in run3. Preferred to take half shifts as one unit, i.e., every time we will take 4 hours, i.e 0.5 shifts. We will need at least 2 days between each study period to make sure data will be analyzed and new script can be written if necessary.

Shift 1 (2 study period, 8 hours total) Establish the reference trajectory. Establish the real time simulation with live parameter. Calibrate BPM response vs magnet strength.

Shift 2 (1 study period, 4 hours total) Finish the final setup of the program. Preliminary study of trajectory tuning with our ACL.

Shift 3 (2 study period, 8 hours total) integration of our ACL to sequence and real beam test.

The final 4 hours will be used as backup just in case any of the planned study take longer than expected.