FRIB LATTICE-MODEL SERVICE FOR COMMISSIONING AND OPERATION*

D. Maxwell†, Z. He, G. Shen, Facility for Rare Isotope Beams, East Lansing, USA

Abstract

Accelerator beam simulation is crucial for the successful commissioning and operation of the FRIB linear accelerator. A primary requirement of the FRIB linear accelerator is to support a broad range of particle species and change states. Beam simulations must be performed for these various accelerator configurations and it is important the results be managed to ensure consistency and reproducibility. The FRIB Lattice-Model Service has been developed to manage simulation data using a convenient web-based interface, as well as, a RESTful API to allow integration with other services. This service provides a central location to store and organize simulation data. Additional features include search, comparison and visualization. The system architecture, data model and key features are discussed.

INTRODUCTION

The Facility for Rare Isotope Beams (FRIB) uses a superconducting linear accelerator designed to accelerate all stable ions to energies greater than 200 MeV/u with beam power on target up to 400 kW [1]. This driver accelerator must be highly configurable in order to accommodate a broad range of ion species and charge states. In order meet these requirements for energy and power on target, accelerator beam simulation software will be important to facilitate accelerator configuration and optimization. Management of beam simulation data will be critical for efficient commissioning and reproducible operation of the FRIB driver accelerator.

The FRIB Lattice-Model Service (LMS) has been designed to allow for easy organization and retrieval of beam simulation data. The primary software used for the FRIB driver accelerator design is IMPACT [2] and therefore the LMS initially supports IMPACT simulation data. However, its flexible data model allows it to be easily extended to support additional types of simulation data. Accelerator simulation data can be divided into two parts: the Lattice and the Model.

Lattice

The Lattice consists of the position and orientation of the accelerator elements along with any information needed to fully define the particles trajectory through the accelerator. This includes the element types and any type specific configuration parameters. As well as, the initial beam conditions including the particle species and initial change state.

Model

The Model is the complete result of executing the beam simulation software with a Lattice as input. The exact content of the Model is specific to each type of beam simulation software. In general the Model includes the following beam parameters at multiple locations along the accelerator: size, position, divergence and energy.

ARCHITECTURE

The LMS must provide users with convenient access to beam simulation data, while being efficient and flexible to meet the demands of commissioning and operation. The overall architecture is adopted from a similar Lattice-Model Service in use at NSCL II [3]. The most significant change from original architecture is the switch from a traditional relational database to a document-oriented database.

The document-oriented database provides a flexible system for storing information and is well suited for storage of beam simulation data. The original relational database scheme was designed to store objects with related arbitrary properties. For example, the Lattice contains a Quadruple element with its magnetic gradient being a property. Likewise, an RF Cavity element with properties amplitude and phase. Through the use of a document-oriented database, more information is now being stored, while the implementation has been greatly simplified. Furthermore, this data model can be more easily extended to support future requirements.

To facilitate convenient access to the simulation data a web interface is provided to users for both submitting new data and for retrieval of existing data. A RESTful API is also provided for programmatic access to simulation data and includes both data submission and retrieval.

IMPLEMENTATION

The LMS is implemented in Python [4] using the Tornado [5] web framework. The use of Python allows for the reuse of existing libraries for reading IMPACT input and result data files. In addition, Python has excellent support for numeric processing, text processing and web applications. Tornado includes a high performance web server which supports modern generator-based asynchronous programming. However, the standard Tornado text template engine was found to be insufficient and instead the more powerful Jinja2 [6] text template engine was used.

The MongoDB [7] document-oriented database is used as it provides excellent performance and integration with Tornado through the use of the Motor [8] client. Motor provides a high performance asynchronous library for interacting with MongoDB.

* This material is based upon work supported by the U.S. Department of Energy Office of Science under Cooperative Agreement DESC0000661, the State of Michigan and Michigan State University.

† maxwelld@frib.msu.edu

FEATURES

When simulation data is submitted to the LMS the Lattice or Model type, particle type and name must be specified. The raw data is then processed and properties are automatically extracted. Figure 1 shows typical properties which have been extracted from IMPACT data files. In addition, this figure shows a list of the raw data files that are available to the user for download.

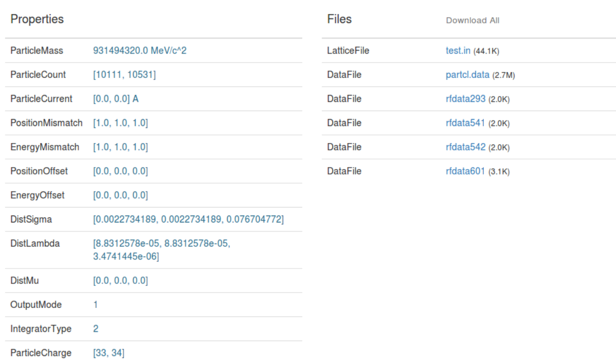

Figure 1: Screen capture showing the Lattice properties and the raw data files that are available for download.

Search

Simulation data can easily be found using the LMS search feature. For Lattice data, searchable fields include the Lattice type, particle type, name, branch name, version number and properties. For Model data, searchable fields include the Model type, particle type, name and properties. Lattice and Model data is completely cross referenced, so the Model data associated with a specific Lattice can easily be found, as well as, the Lattice data associated with a specific Model.

Visualization

Model data can be easily viewed and multiple beam properties can be plotted for convenience. Figure 2 shows a plot of the beam position and energy.

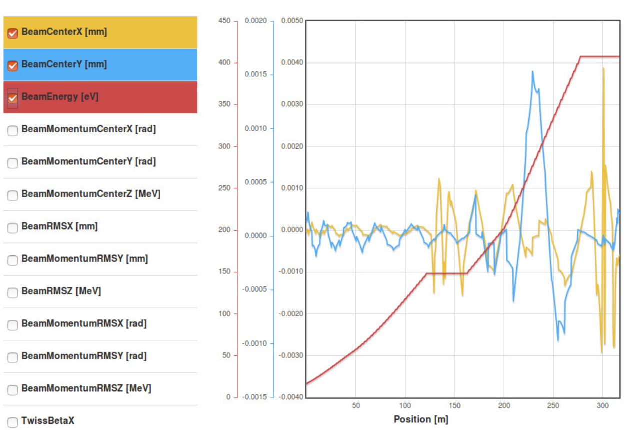

Figure 2: Screen capture showing Model data multi-plot with beam position X (yellow), position Y (blue) and energy (red). All data is plotted versus the accelerator position in meters.

Comparison

Another important feature of the LMS is the ability to compare Lattice and Model data in order to determine the differences between the two data sets. Figure 3 shows the comparison of a short section of Lattice data. The different element property values are clearly indicated to the user by marking them in red.

Figure 3: Screen capture showing a comparison of two Lattices with the differences indicated in red.

The difference between two Models can be seen easily by plotting the data on the same plot. Figure 4 shows comparison of the beam position in the X-direction.

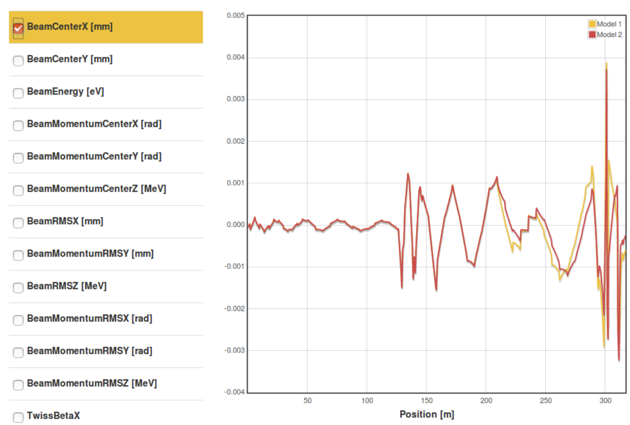

Figure 4: Screen capture showing the comparison of beam position X for Model 1 (yellow) and Model 2 (red).

FUTURE DEVELOPMENT

The Fast Linear Accelerator Modelling Engine (FLAME) [9] will be the primary simulation software used for commissioning and operation of the FRIB driver accelerator. FLAME has been designed to balance the need for both fast and accurate on-line accelerator beam simulation with particular attention to the specific requirements of FRIB. The LMS will be extended to support FLAME Lattice and Model simulation data.

The web interface will be enhanced to provide a more streamlined approach for data submission as well as allowing the user to specify additional properties when submitting data. Furthermore, an improved security model may be required for more control over user authorization.

CONCLUSION

The FRIB LMS will be an important tool for commissioning and operation of the FRIB driver linear accelerator. Beam simulation will be critical for meeting the FRIB the performance requirements and the LMS provides a centralized repository for management of beam simulation data.

ACKNOWLEDGEMENT

This work is supported by the U.S. Department of Energy Office of Science under Cooperative Agreement DESC0000661, the State of Michigan and Michigan State University.

REFERENCES

[1] Jie Wei, D. Arenius, N. Bultman, F. Casagrande, C. Compton, K. Davidson, et al., “Progress toward the Facility for Rare Isotope Beams,” *Proc. NA-PAC’13*, Pasadena, CA, USA, Sept . 2013, paper FRYBA3.

[2] Ji Qiang, Robert D. Ryne, Salman Habib, Viktor Decyk, “An Object-Oriented Parallel Particle-in-Cell Code for Beam Dynamics Simulation in Linear Accelerators,” *Proc. of the ACM/IEEE conference on Supercomputing*, Portland, OR, USA, Nov. 1999, pp. 55.

[3] G. Shen et al., “NSLS II Middlelayer Services”, *Proc. of ICALEPCS’13*, San Francisco, CA, USA, Oct. 2013, paper MOPPC155.

[4] Python, `https://www.python.org`

[5] Tornado, `http://www.tornadoweb.org`

[6] Jinja2, `http://jinja.pocoo.org`

[7] MongoDB, `http://mongodb.com`

[8] Motor, `http://github.com/mongodb/motor`

[9] FLAME, `http://github.com/frib-high-level-controls/FLAME`