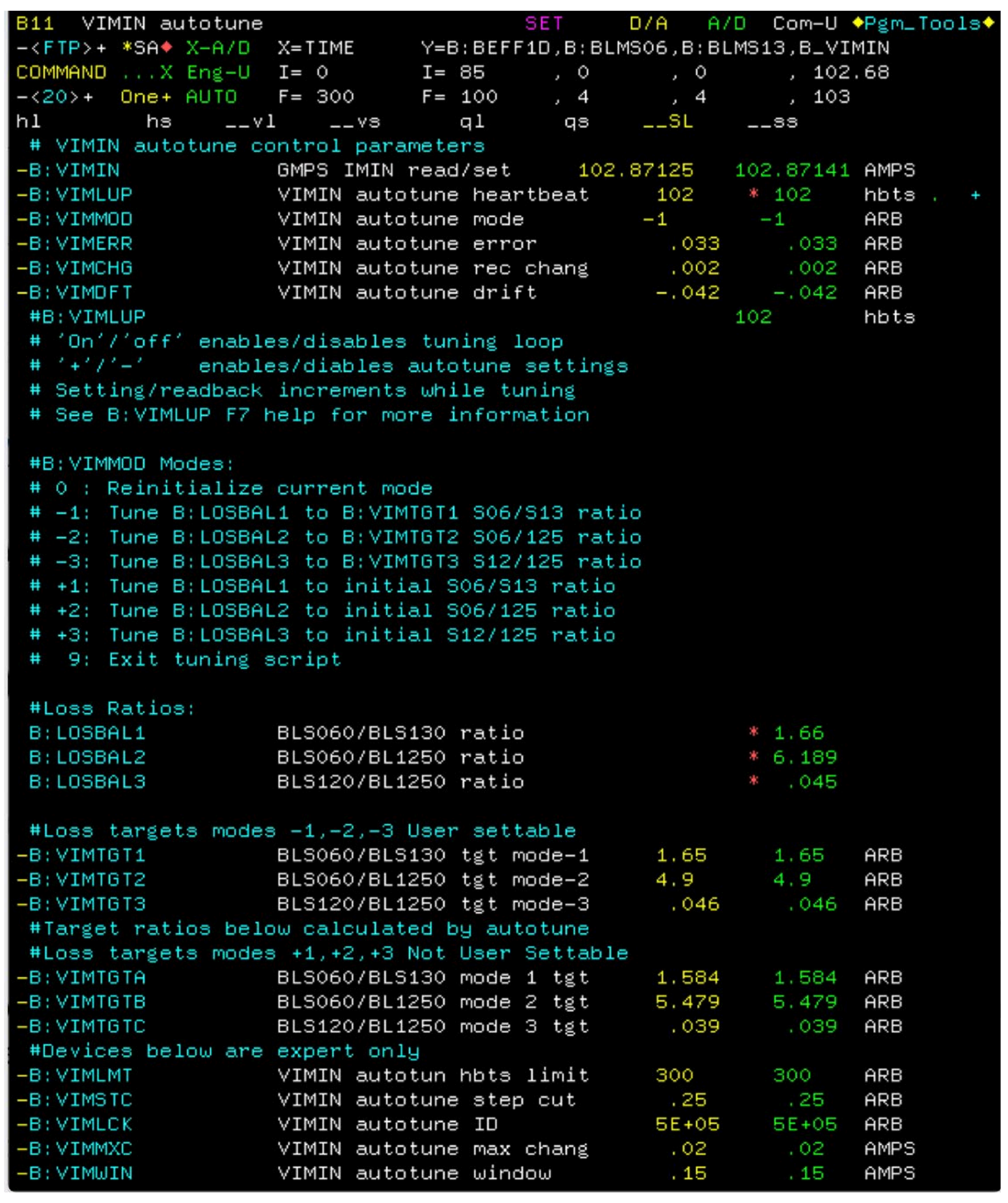
FERMILAB-POSTER-24-0123-AD

MLOps for Beam Controls

Gopika Bhardwaj, Fermilab

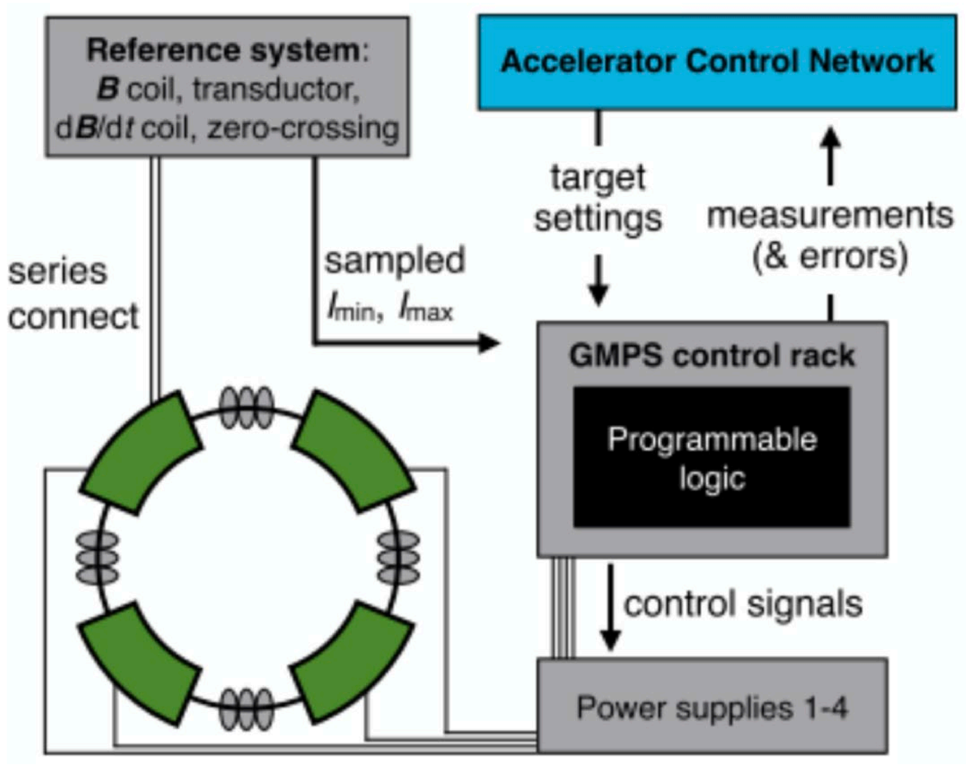
Problem Statement

Currently, applications for accelerator tuning are written in Fermilab's in-house accelerator scripting language (ACL). Code development and testing is done **solely by the subject matter expert**. Parameter tuning is **not explored systematically** and **not documented**. The code has **limited documentation** which makes it **challenging to replicate** in modern programming languages.



The Gradient Magnet Power Supply in Fermilab's Booster synchrotron

We consider a simple example in Fermilab's Booster Synchrotron that accelerates protons from the Linac from 400 MeV to 8 GeV. Undesired variations in the electromagnet current causes beam losses as protons are injected into Booster. VIMIN is the minimum current set point for the **G**radient **M**agnet **P**ower **S**upply (GMPS) 15 Hz sinusoidal curve.



Currently, there is a persistent ACL script that uses the integrated beam loss monitors every super cycle to correct for VIMIN drift.

It takes beam data once per supercycle and makes a change to regulate the selected loss ratio to the target value.

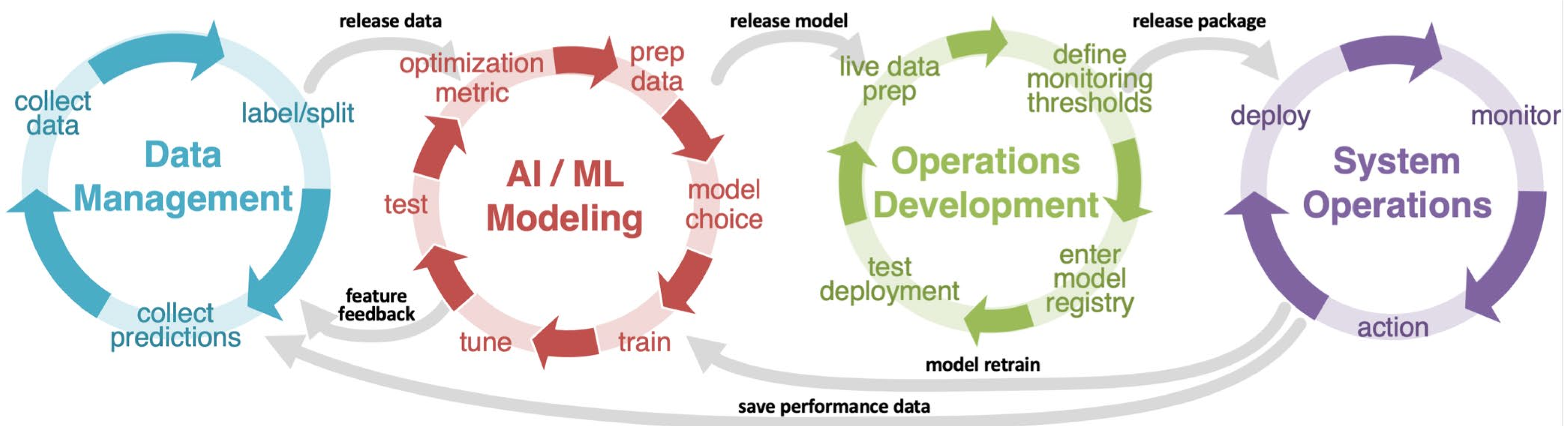
While the script only considers loss monitor ratios to make its changes, subject matter experts recognize that Linac beam energy variations, gallery/rack temperatures, beam intensity/rep rate changes, orbit tuning and any other beam condition changes also impact VIMIN tuning. To prepare for a more sophisticated model that considers other indicator variables, we need to deploy a principled MLOps workflow.

Machine Learning Operations (MLOps)

MLOps is the standardization and streamlining of the ML development lifecycle to address the challenges associated with large-scale machine learning projects such as changing data dependencies, varying business needs, reproducibility, and diverse teams working with differing tools and skills.

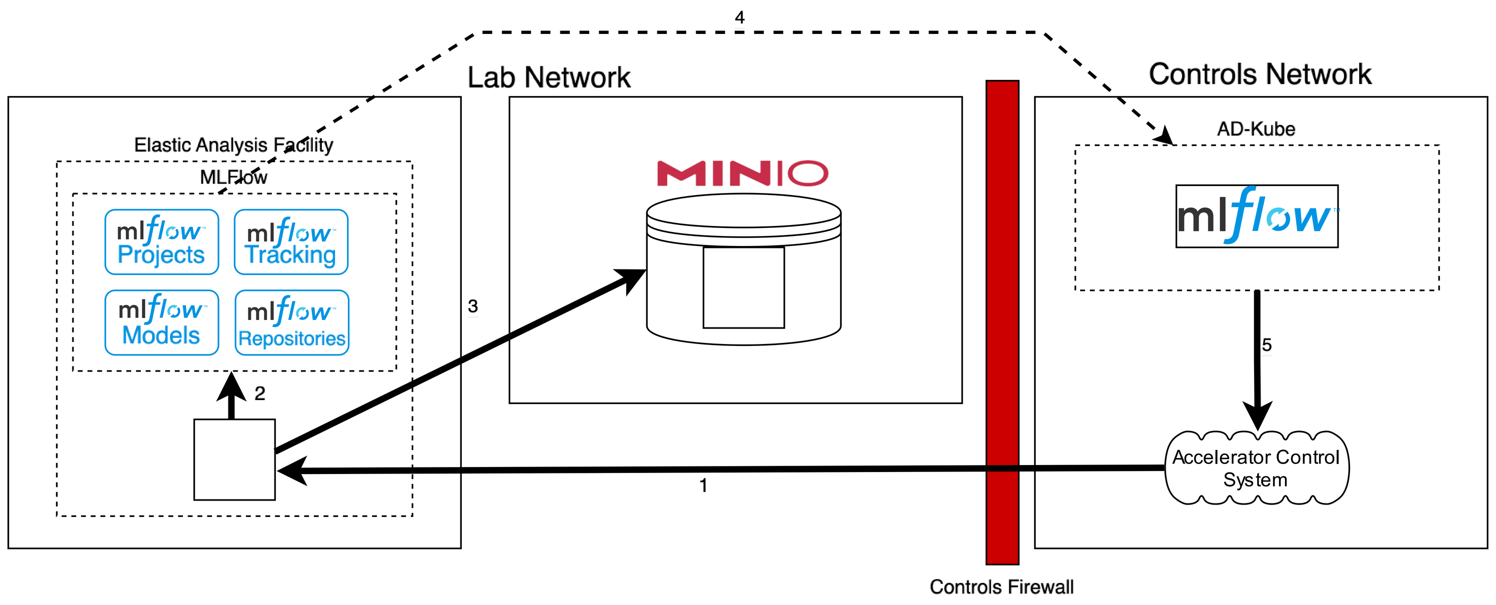
Phase I: MLOps Rollout

For Phase 1, we re-wrote the ACL script in Python and successfully replicated the VIMIN values computed. This script is for demonstrating the rollout of our MLOps pipeline. It gets historical values of loss monitors and computes the value for VIMIN. It was tested alongside the current ACL script, and both scripts computed the same VIMIN setting.



1. **Data Management**: Data is collected from the Accelerator Control System using acsys-python. The data is split into train/validation/test set and versioned in DataHub and archived in MinIO S3 compatible store
2. **AI/ML Modeling**: We choose metrics to optimize and conduct model trials with training and tuning. MLFlow Tracking allows us to keep track of the code, data, configuration and results for each experiment.
3. **Operations Development**: Models are stored in a central repository in MLFlow Registry which provides model lineage versioning, aliasing, tagging, and annotations.
4. **System Operations**: MLFlow Models is used for serving models and models can be reproduced with MLFlow Projects in a platform-agnostic format.

We will test this prototype pipeline by deploying the VIMIN Python code.



Phase 2: ML Booster Optimization

For Phase 2 we will explore optimization-based ML techniques and expand the range of input features to include beam conditions, gallery and ambient conditions, Booster tunnel parameters, and other utility parameters to improve Booster performance and minimize average beam loss over time.

Acknowledgments

This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.