

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.

```

B11 VIMIN autotune      SET      D/R   A/D  Com-U  Pgm_Tools
--<FTP>+ *$#* X-A/D  X=TIME   Y=B:BEFF1D,B:BLMS05,B:BLMS13,B:VIMIN
COMMAND  X Eng-U  I= 0      I= 85   , 0      , 0      , 102.68
-->O+ One+ AUTO   F= 300   F= 100   , 4      , 4      , 103
h1   hs   --v1   vs   q1   qs   _SL   _LSS
# VIMIN autotune control parameters
-B:VIMIN   GMPS IMIN read/set 102.87125 102.87141 AMPS
-B:VIMLUP   VIMIN autotune heartbeat 102 * 102 hbt5 . +
-B:VIMMOD   VIMIN autotune mode 102 * 102 ARB
-B:VIMERR   VIMIN autotune error .033 .033 ARB
-B:VIMCHG   VIMIN autotune rec chang .002 .002 ARB
-B:VIMDFT   VIMIN autotune drift -.042 -.042 ARB
#B:VIMLUP 102 hbt5
# 'on/off' enables/disables tuning loop
# '+' '-' enables/disables autotune settings
# Setting/readback increments while tuning
# See B:VIMLUP F7 help for more information

#B:VIMMOD Modes:
# 0 : Reinitalize current mode
# -1: Tune B:LOSBAL1 to B:VIMGT1 S06/S13 ratio
# -2: Tune B:LOSBAL2 to B:VIMGT2 S06/125 ratio
# -3: Tune B:LOSBAL3 to B:VIMGT3 S12/125 ratio
# +1: Tune B:LOSBAL1 to initial S06/S13 ratio
# +2: Tune B:LOSBAL2 to initial S06/125 ratio
# +3: Tune B:LOSBAL3 to initial S12/125 ratio
# 9: Exit tuning script

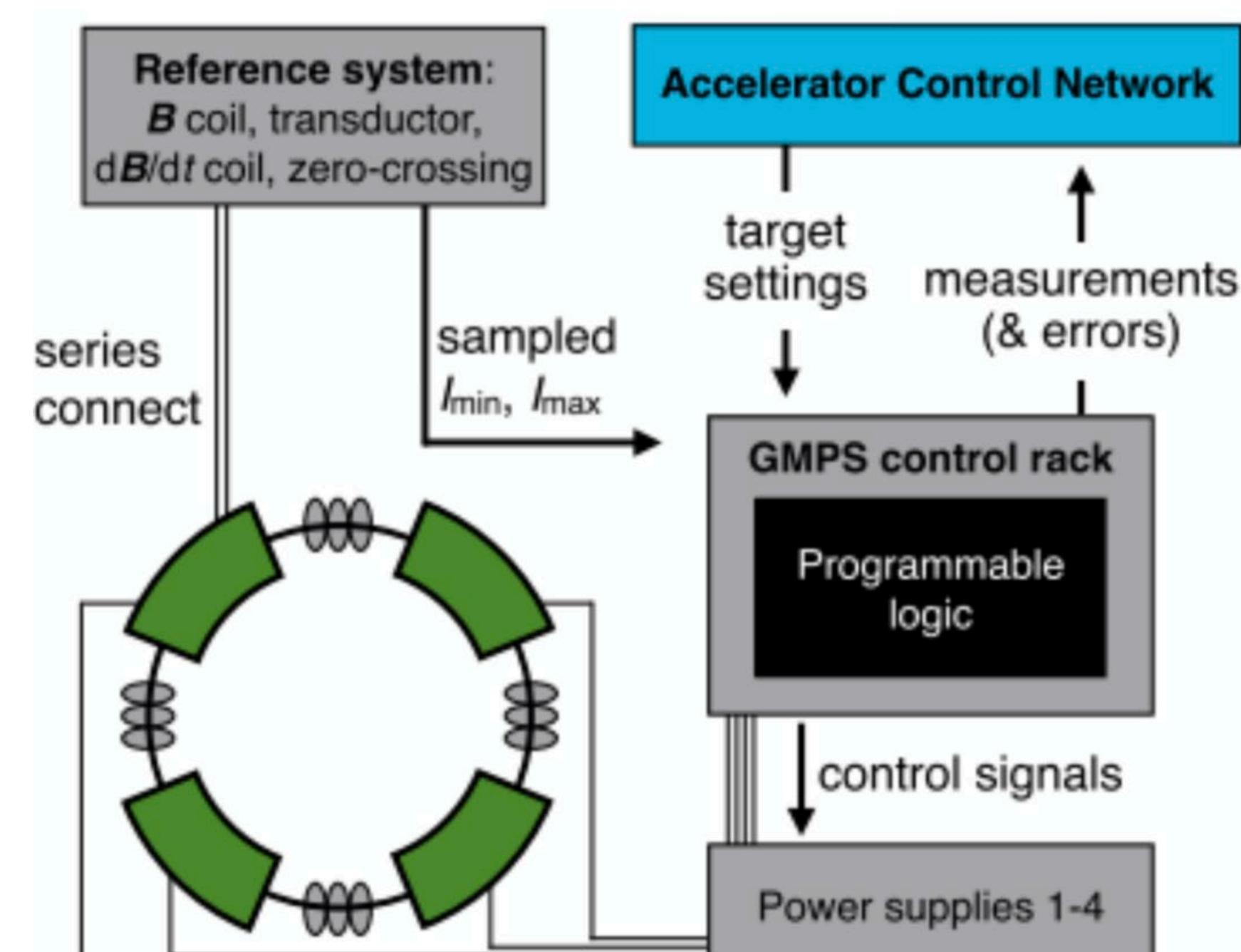
#Loss Ratios:
B:LOSBAL1   BLS060/BLS130 ratio * 1.66
B:LOSBAL2   BLS060/BLS125 ratio * 1.619
B:LOSBAL3   BLS120/BLS125 ratio * 0.45

#Loss targets modes -1,-2,-3 User settable
-B:VIMGT1   BLS060/BLS130 tgt mode-1 1.65 1.65 ARB
-B:VIMGT2   BLS060/BLS125 tgt mode-2 4.9 4.9 ARB
-B:VIMGT3   BLS120/BLS125 tgt mode-3 .046 .046 ARB
#Target ratios below calculated by autotune
#Loss targets modes +1,+2,+3 Not User Settable
-B:VIMGT4   BLS060/BLS130 mode 1 tgt 1.584 1.584 ARB
-B:VIMGT5   BLS060/BLS125 mode 2 tgt 5.479 5.479 ARB
-B:VIMGT6   BLS120/BLS125 mode 3 tgt .039 .039 ARB
#Devices below are expert only
-B:VIMLMT   VIMIN autotune hbt5 limit 300 300 ARB
-B:VIMSTC   VIMIN autotune step cut .25 .25 ARB
-B:VIMLCK   VIMIN autotune ID 5E+05 5E+05 ARB
-B:VIMMXC   VIMIN autotune max chang .02 .02 AMPS
-B:VIMWIN   VIMIN autotune window .15 .15 AMPS

```

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 **Gradient Magnet Power Supply (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.

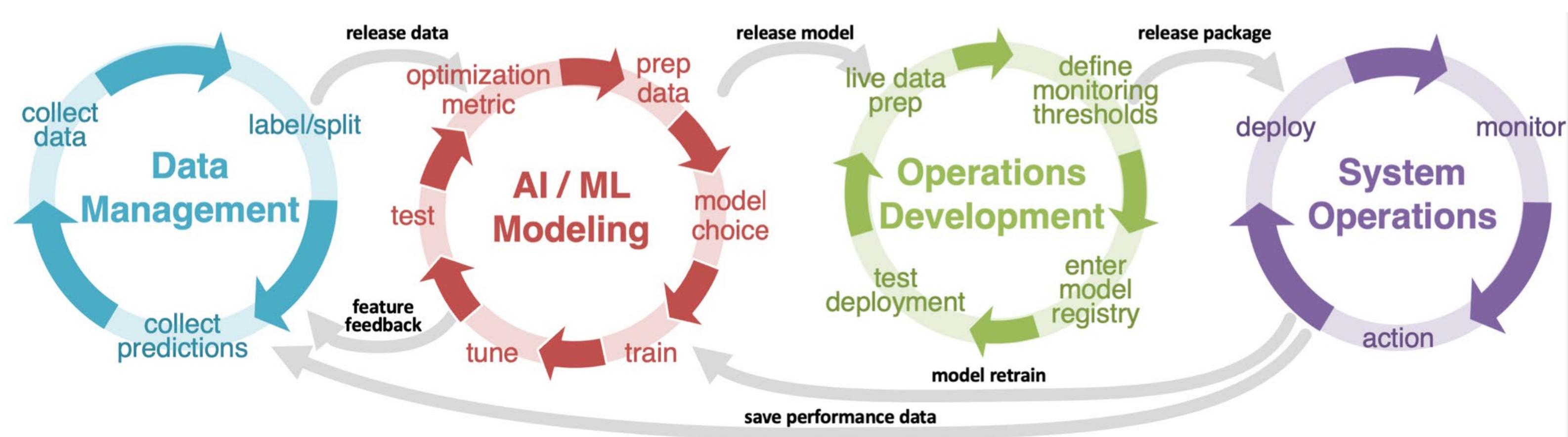
While the script only considers loss monitor ratios to make its changes, subject matter experts recognize that Linac beam energy variations, gallery/rack temperatures, beam intensity/rep rate changes, orbit tuning and any other beam condition changes also impact VIMIN tuning. To prepare for a more sophisticated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



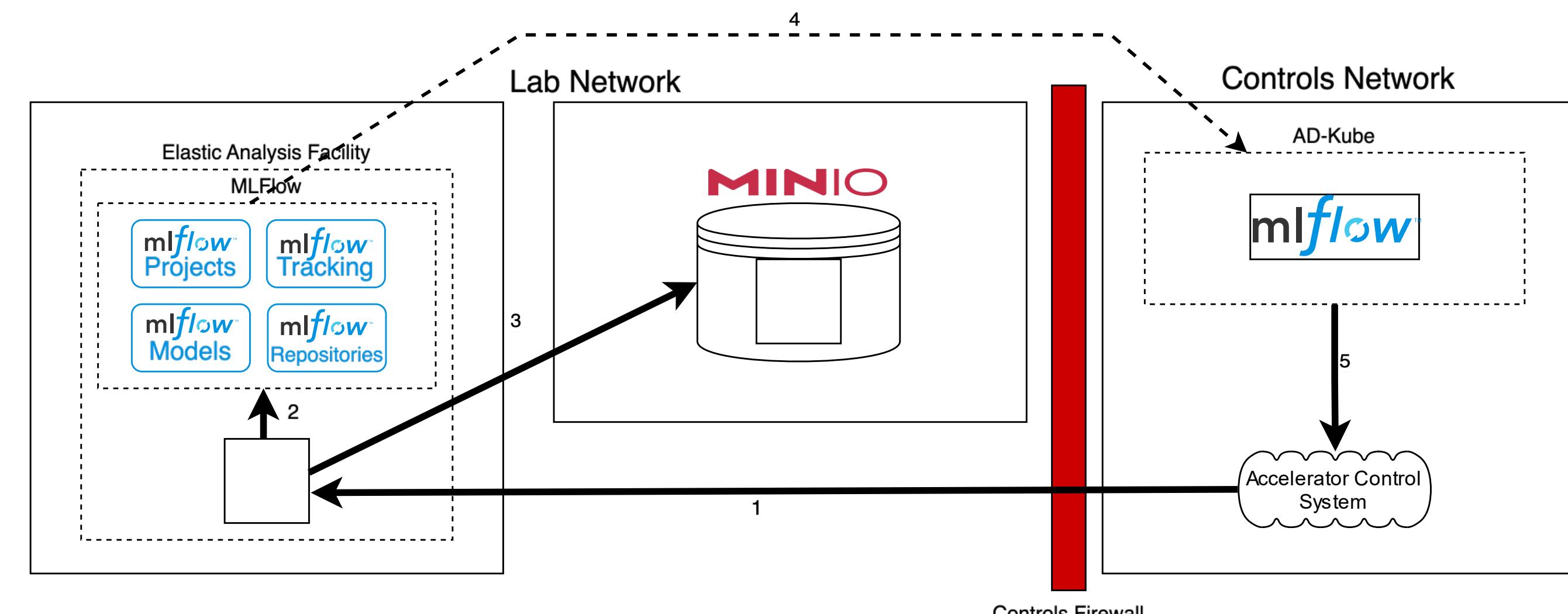
1. **Data Management**: Data is collected from the Accelerator Control System using acsys-python. The data is split into train/validation/test set and versioned in DataHub and archived in MinIO S3 compatible store

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.