

BEAM TUNING AUTOMATION ACTIVITIES AT TRIUMF*

S. Kiy, J. Adegun, F. Ames, A. Andres, R. Baartman, H. Bagri, K. Ezawa,
W. Fedorko, P. Jung, O. Kester, K. Lucow, J. Nasser, T. Planche, S.D. Rädcl,
B. Schultz, O. Shelbaya, B. Stringer, D. Thomson, D. Wang, K. Wu
TRIUMF, Vancouver, BC, Canada

Abstract

The particle accelerator complex at TRIUMF provides beams for secondary particle production including rare isotopes. The post acceleration of rare isotope ions demands frequent changes of beam properties like energy and changes of the ion species in terms of isotope and charge state. To facilitate these changes to beam properties and species, a High Level Applications (HLA) framework has been developed that provides the essential elements necessary for app development: access to sophisticated envelope simulations and any necessary beamline data, integration with the control system, version control, deployment and issue tracking, and training materials. With this framework, one can automate collection of beam data and subsequently pull that data into a model which then outputs the necessary adjustments to beam optics. Tuning based on this method is model coupled accelerator tuning (MCAT) and includes pursuits like the training of machine learning (ML) agents to optimize corrections benders. A summary of the framework will be provided followed by a description of the different applications of the MCAT method — both those currently being pursued, and those envisioned for the future.

MOTIVATION

Programs that aid accelerator operation are often developed in an ad-hoc manner by physicists — *particularly for themselves as end-users* — as the facility expands.

In the case of TRIUMF, a lab more than 50 years old, there are various tools and programs that have been developed by experts for different operating systems using different programming languages for a variety of different accelerators across the campus. However, this approach does not scale well as the size and complexity of the lab increase — too many tools can end up being duplicated and not all are well maintained.

The newly installed e-linac and upcoming new proton beamline from the main cyclotron will drive rare isotope beam (RIB) production via new target stations and associated beamlines in ARIEL. This is expected to triple the number of deliverable RIB hours to experiments.

To allow for reliable operation in this new era, a High Level Applications (HLA) task force was established at TRIUMF in 2017 [1] with the mission to reduce facility overhead and improve beam reliability [2]. To accomplish this, the taskforce was to create a standard software framework for

creating applications that can be easily used at any of our many accelerator facilities, as depicted below in Fig. 1.



Figure 1: Layout of TRIUMF beamlines based on data from acc xml files. E-linac devices are shown in orange, primary proton beamlines in red, the new ARIEL RIB transport beamlines in green, and ISAC RIB transport beamlines in blue.

HLA FRAMEWORK

The software development framework itself is critical to the development of robust apps as well as enticing staff members to contribute to existing code repositories rather than working alone.

Jaya – Integration with the Control System

Jaya is a web application acting as the middle layer that allows communication with the underlying low-level control system. It currently maintains channel access (CA) monitors on over 5000 process variables (PVs) distributed over five independent EPICS subnets at TRIUMF.

PV values are stored in an in-memory database (Redis). This way, any HLAs that request to read PV values only poll the HLA server database and don't produce any additional load on the EPICS networks.

A second noteworthy component of jaya is the additional backend processes that allow for long running measurements. Jaya uses a Python task queue package called Celery, which allows passing of jobs between the jaya web application and the Celery backend process via Redis. Using this tool, jaya allows users to create custom scans or measurements that allow for setting, waiting, and saving of data as required by the users. An example of this functionality is shown in Fig. 2. In the measurement shown [3], the current of a magnetic quadrupole was changed in various step sizes and directions, while saving magnetic field data to an HLA database used for storing beam measurements (beamDB).

* Funded under a contribution agreement with NRC (National Research Council Canada).

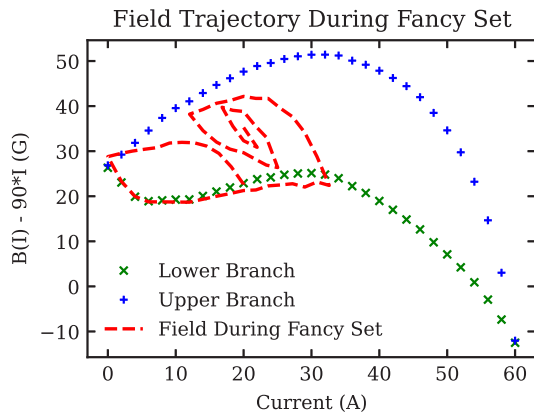


Figure 2: Field during algorithm for setting of magnetic quadrupoles.

Acc – xml Description of Beamline Information

Acc is a collection of xml files describing TRIUMF's beamlines that are intended to contain all necessary information needed to locate devices, generate envelope simulations, load and scale optics settings, and more. It contains the necessary references to all relevant design drawings from which beamline positions were attained.

The acc files are stored in a version controlled (git) repository and available via the DevOps platform Gitlab. Over 20 staff members have contributed to over 1000 commits in this repository, demonstrating the interest and utility in an easily accessible open-source tool such as this.

beamDB – Database of Beam Measurements

Coordination and consistency in saving and accessing beam related measurements can be difficult. Historically users have saved data from diagnostics to a shared file server on a case by case basis. Filenames are chosen by users, other relevant data is difficult to link, and the end result is a large flat directory of data which can be hard to navigate. This could pose a significant issue as we look forward to applications of machine learning on the operation of beamlines and accelerators.

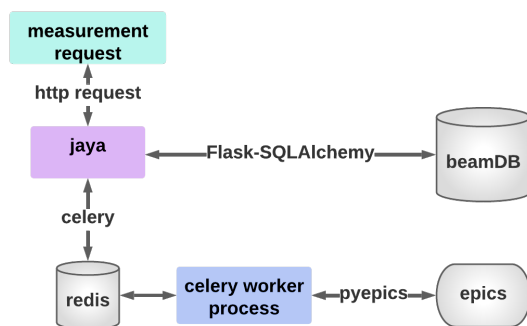


Figure 3: Schematic of the fundamental elements for automation - jaya, the beamDB, backend processes, and connection to EPICS.

To improve the way in which beamline/tuning related data is saved and later searched/utilized, we have implemented a database solely to store measurements of various particle beams [4]. We are using a PostgreSQL database accessed via jaya using Flask-SQLAlchemy, as shown in Fig. 3.

For the RIB facilities of TRIUMF, the beamDB stores beam properties, optics settings, and a user-defined collection of diagnostics measurements all linked together in various tables. Paired with the automation capabilities in the backend of jaya, this has enabled us to begin building a large training dataset that can in the future be used to train various ML agents.

Profiles – Package for Processing Data

Measuring the profiles of various particle beams is an essential part of understanding both the machine and beam properties. One of the most common instruments for measuring beam profiles at TRIUMF is a wire scanner type device, called a rotary profile monitor (RPM). The raw data output from an RPM is two arrays: one a voltage (representing position) and the other a current.

These devices (and others) have various different features that can complicate processing of the profile data. This can include electrical noise, mechanical issues, potentiometer issues, and more. To provide a consistent approach that all HLA users and apps can utilize, a co-op project was completed [5] which developed a Python package to process profile data. This package utilizes a random forest classifier (RFC) system to select the most appropriate of four deterministic algorithms for processing the data. The RFC was trained by having staff members with experience of the various wire scanned failure modes classify a large set of RPM data into a number of different categories. Quality data processing of a beam profile (Fig. 4) accepts all data points that are really representative of the underlying beam distribution, while rejecting other data. Poor profile processing often incorrectly includes noise or background in the calculation of the beam size (Fig. 5).

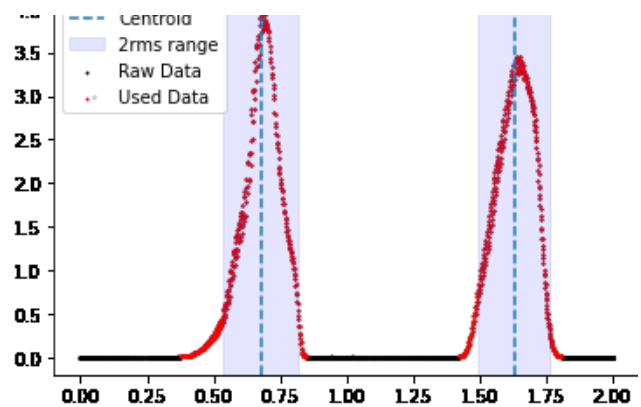


Figure 4: Example of good data processing of an RPM scan.

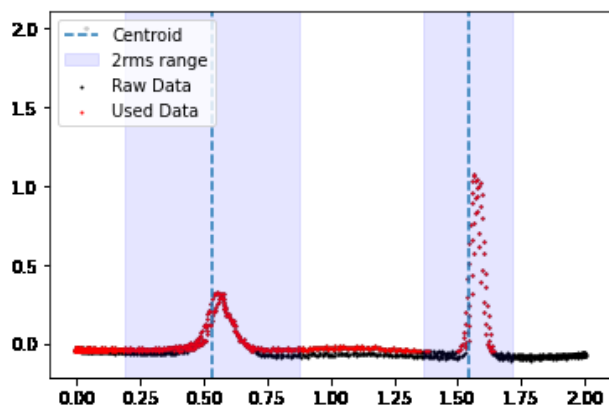


Figure 5: Example of poor data processing of an RPM scan.

accpy and fancy_set – Interfacing with acc and jaya

accpy is a Python package built with a number of different modules. One module provides a collection of functions to connect to the jaya backend running on the server to get and set data to the control system. A second module allows users to easily query and pull beamline information from the acc xml file structure.

A recent addition to accpy was from a co-op project [3]. This project added an algorithm to improve the way in which magnetic optics are set such that effects of hysteresis are minimized. This new ramping technique, fancy_set, is particularly of use for quadrupoles with unipolar power supplies. The routine cycles the quadrupole setpoint around the desired final current at successively smaller amplitudes as shown in Fig. 6 and the resulting path the magnetic field takes is shown earlier in Fig. 2.

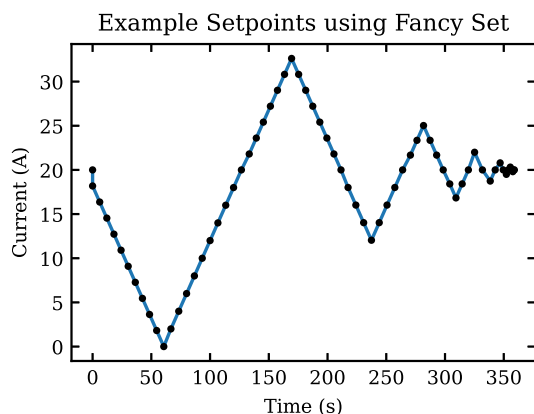


Figure 6: Quadrupole current vs time using the fancy_set routine.

This new routine is expected to help establish more accurate and stable tunes, thus reducing tuning time for operators overall. Initial tests have also been carried out to use this routine for ramping of the main cyclotron magnet.

Gitlab – DevOps Platform

To enable the development of HLAs at TRIUMF, a Gitlab server was established. Gitlab is a DevOps¹ platform that has many benefits. This include the Continuous Integration/Continuous Deployment (CI/CD) feature, which allows custom pipelines of jobs to be executed when new versions of code are committed. With this, we are able to have applications automatically redeploy when new code is committed to gitlab, which can be to either a development or production machine. Gitlab also allows us to serve up repositories of packaged Python code which can easily be installed by users or applications. Last, Gitlab includes an issue tracking feature through which requests, bugs, and improvements are tracked, allowing for easy collaboration.

Docker and Kubernetes

Applications developed are now being built and deployed primarily as Docker images within a Kubernetes cluster. This has a number of significant benefits including: basically fully eliminating dependency issues both when developing and deploying; improving scalability - Kubernetes can start extra processes of an app when demand is high; further simplifying and expediting the deployment process.

APPLICATIONS

Automation

The long-running backend measurements enabled by jaya give students and physicists the ability to code custom measurements. This can be done both on a local Python script which posts the measurement request to be approved by a control room console, or within apps themselves. One recent example of this was to enable multi-parameter scans of a FEBIAD (Forced Electron Beam Induced Arc Discharge) ion source, shown in Fig. 7. This ability assists both with the analysis of FEBIAD models and the identification of optimal operating settings.

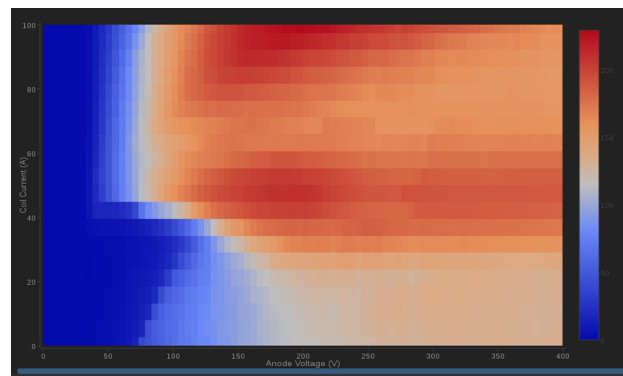


Figure 7: Heatmap showing ion beam intensity for varying FEBIAD source settings (Coil current and Anode voltage).

¹ Software development and IT operations. DevOps is an approach toward expediting the cycle between developing and deploying new code.

Snapshot – Data Collection

In order to populate the beamDB quickly with well categorized information, there was a need to improve the way in which beam current, profiles, and optics settings were collected. Historically data had been collected via buttons on EPICS Extensible Display Manager (EDM) screens and a Perl/TK tool. Information was then stored as html tables and screenshots in an electronic logbook.

A project led by an operator leveraged the existing HLA framework to create a web application that automates these processes and stores the data into the beamDB. In its first year of use this app has already noticeably reduced the work load on operators and has contributed to the over 1000 data sets saved to the beamDB. Figure 8 shows the UI view of two beam profiles saved with snapshot.

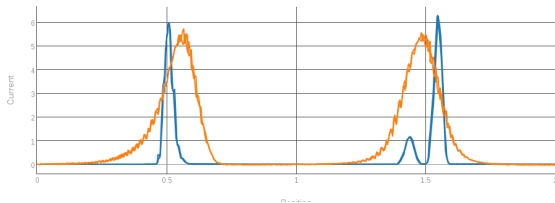


Figure 8: Webpage UI showing profiles stored in the beamDB that were automatically collected using the snapshot app.

Model Based Phasing of ISAC-II Linac

The ISAC-II superconducting heavy ion linac at TRIUMF accelerates both stable and radioactive ions up to as high as 16.5 MeV/u through 40 quarter wave resonators distributed over eight cryomodules. The cavities are individually controlled and their configuration regularly changes depending on cavity performance and requested beam properties from the experiments it serves.

Also utilizing the HLA framework, a project for modelling and automating the phasing of the cavities is nearing completion. The phase offsets and amplitudes of the cavities have been investigated and calibrated allowing for initial tests of an automated phasing routine [6] which has been demonstrated to reduce overhead in switching linac configuration from over eight hours to less than one.

Model-Coupled Accelerator Tuning (MCAT)

The envelope code TRANSOPTR [7], in use at TRIUMF since 1984 [8], has recently been expanded to include RFQs [9] and KONUS DTLs [10]. Its native FORTRAN is capable of executing start to end accelerator simulations in under one second, suitable for real-time use. Using the code's internal optimizers (simulated annealing or downhill simplex), it is possible to rapidly re-compute entire machine tunes, using on-line beam diagnostic data as inputs [11]. This technique of model coupled accelerator tuning (MCAT) is being developed at the ISAC-I linac, aiming to produce a closed loop, diagnostic driven feedback control system for the accelerator.

Machine Learning Based Beam Steering

Tuning the steerers of a particle accelerator is a complex process, often with a high dimensional parameter space and limited observables. These characteristics make it potentially suited for reinforcement algorithms. The ultimate goal is to reduce the tuning time for such facilities. The training process is an iterative process in steps in which the agent takes the state s_t of a beam line as an input and predicts an action a_t sampled from a policy $\pi(a_t|s_t)$. The policy is updated using Recurrent Deterministic Policy Gradient (RDPG) algorithms based on the so-called reward (calculated from beam transmission) [12, 13]. The optimization process for different steps on a simulated beam line using TRANSOPTR is shown in Fig. 9. Initial investigations and tests of this approach [14] have been carried out over the past year using the offline ion source (OLIS) at TRIUMF.

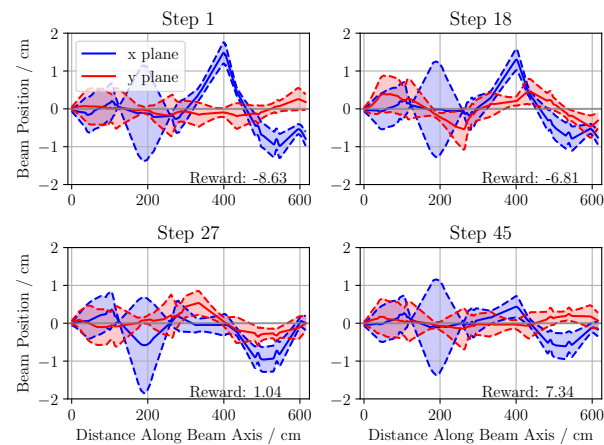


Figure 9: A validation of a trained agent on the simulated OLIS beam line. The panels show the evolution of the steerer predictions of a trained agent for different steps. The reward (calculated from the beam transmission) is increasing during the optimization process.

CONCLUSION

Automation of beam tuning activities is a major area of interest in labs around the world. TRIUMF has now implemented a robust framework using modern tools that has allowed development of apps for automation, high-level physics modelling and feedback, and user-defined measurements that can be saved to a properly indexed database. All software is available openly within the TRIUMF network for other staff members to utilize and contribute to. Further automation is expected in the near future as we strive to accumulate large data sets for future applications of machine learning.

REFERENCES

- [1] C.B. Barquest *et al.*, “Web-Based Control Room Applications at TRIUMF”, in *Proc. IPAC'18*, Vancouver, Canada, Apr.-May 2018, pp. 4832–4835. doi: 10.18429/JACoW-IPAC2018-THPML078

- [2] C. Barquest, “A brief history of high level applications at TRIUMF,” TRIUMF, Tech. Rep. TRI-BN-19-01, 2019.
- [3] J. Nasser, R. Baartman, O. Kester, S. Kiy, T. Planche, S. Rädels, and O. Shelbaya, “Algorithm to mitigate hysteresis in magnets with unipolar power supplies,” in *Proc. IPAC’22*, Bangkok, Thailand, Jun. 2022, to be published. doi:10.18429/JACoW-IPAC2022-MOPOST039
- [4] O. Shelbaya, “HLA-beamData: An Elementwise Database for Passive Beam Data Accumulation,” TRIUMF, Tech. Rep. TRI-BN-19-16, 2019.
- [5] K. Wu, “Profile Monitor Classification using Random Forest Classifier,” TRIUMF, Tech. Rep. TRI-BN-21-11, 2021.
- [6] S. Kiy, R. Baartman, O. Kester, and O. Shelbaya, “First Tests of Model-Based Linac Phasing in ISAC-II,” presented at the 15th Heavy Ion Accelerator Technology Conf. (HIAT’22), Darmstadt, Germany, June 2022, paper TUP19, this conference.
- [7] R. Baartman, “TRANSOPTR Reference Manual,” TRIUMF, Tech. Rep. TRI-BN-22-08, 2022.
- [8] R. Baartman, “TRANSOPTR: Changes since 1984,” TRIUMF, Tech. Rep. TRI-BN-16-06, 2016.
- [9] O. Shelbaya, R. Baartman, and O. Kester, “Fast radio frequency quadrupole envelope computation for model based beam tuning,” *Phys. Rev. Accel. Beams*, vol. 22, no. 11, p. 114602, 2019. doi:10.1103/PhysRevAccelBeams.22.114602
- [10] O. Shelbaya, T. Angus, R. Baartman, P.M. Jung, O. Kester, S. Kiy, T. Planche, and S. D. Rädels, “Autofocusing drift tube linac envelopes,” *Phys. Rev. Accel. Beams*, vol. 24, no. 12, p. 124602, 2021. doi:10.1103/PhysRevAccelBeams.24.124602
- [11] O. Shelbaya, R. Baartman, P. Jung, O. Kester, S. Kiy, T. Planche, Y.-N. Rao, and S. Rädels, “On-Line Retuning of ISAC Linac Beam with Quadrupole Scan Tomography,” in *Proc. IPAC’21*, Campinas, SP, Brazil, May 2021, pp. 4187–4189. doi:10.18429/JACoW-IPAC2021-THPAB205
- [12] T.P. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” 2019. doi:10.48550/arXiv.1509.02971
- [13] N. Heess, J.J. Hunt, T.P. Lillicrap, and D. Silver, “Memory-based control with recurrent neural networks,” 2015. doi:10.48550/arXiv.1512.04455
- [14] D. Wang, R. Baartman, H. Bagri, W. Fedorko, P. Jung, O. Kester, S. Kiy, C. Macdonald, T. Planche, and O. Shelbaya, “Accelerator tuning with deep reinforcement learning,” in *Proceedings of the Fourth Workshop on Machine Learning and the Physical Sciences (NeurIPS)*, 2021. https://ml4physicalsciences.github.io/2021/files/NeurIPS_ML4PS_2021_125.pdf