





J. Rye, C. Elliott, J. Mammosser, A. Zhukov - Spallation Neutron Source Oak Ridge, TN U.S.A.

Abstract

At the Spallation Neutron Source, operators have developed software for assessing tuner motor functionality on the superconducting linac (SCL). Additionally, a graphical user interface (GUI)-based application has been created to automate the optimization of beam losses and reduction of residual activation levels within the accelerator tunnels. Python and Qt made this possible. Modern high-level programming languages like Python and GUI frameworks such as Qt and Qt Designer enable individuals to quickly advance from beginners to creating sophisticated interactive applications. These tools have significantly enhanced our capacity to develop user-friendly solutions to accelerate the completion of otherwise lengthy and complex processes.

Testing SCL Tuner Motor Functionality

chief-oper@ics-opi-ccr10:~	<u>^ _ D</u>	×
Is the CHL at 2k? y/n: y Cavities currently detuned? (Answering 'y' will move tuners +100) To continue hit <enter> Ramping RF Finding max fields To continue hit <enter> Moving tuners 30 steps. Waiting</enter></enter>	y/n: n	

SCL tuners need to be tested before and after every run cycle. This process used to take on average 2 hours with the 81 cavities originally installed and could have taken 3 hours or more by the end of the Proton Power Upgrade (32 additional cavities). With the implementation of this software, we have gotten that down to just a few minutes regardless of the number of tuners to be tested.

General algorithm:

- Ramp cavities to a low gradient
- Move klystrons in 0.5 kHz increments to roughly find resonance
- Move klystrons in 0.1 kHz increments to find max fields • Move tuners off resonance, verify field drop

Parameters measured:

- Cavity field
- Klystron frequency
- Motor position

settling... Checking for field drop... To continue hit <ENTER> Moving tuners -30 steps. Waiting... Settling... 23d controls failure ************ Tuner 23d appears to have failed. *********** Checking for field return... To continue hit <ENTER> Writing file... Killing RF... [chief-oper@ics-opi-ccr10 ~]\$

• Move tuners back on resonance, verify field return

Considerations:

- Temperature of the cavity
- On resonance vs detuned at the start
- Where the tuners should end up, detuned vs on resonance
- As always, how to store the data for easy reference later

Future plans:

- Re-write in PySide6 utilizing an OOP approach
- Launching in separate threads
- Graphical User Interface

- Motor movement command
- Motor hard stop limit switches

Checks performed:

- Field drop on detune
- Field return on retune
- Motor position within an acceptable range of the goal after every move
- No hard limit reached
- Move and stop commands given / rescinded appropriately

Automated Beam Loss Tuning (ABLT) Interface

The Automated Beam Loss Tuning project is a collaboration between Operations and Accelerator Physics groups with the ultimate goal of utilizing machine learning (ML) in tuning beam losses and lowering residual activation levels in the accelerator tunnels.

Operators perform the majority of the work, writing all code related to the scan, the GUI, and postrun correlation data analysis / visualization. We also perform the scans and take all the data. A member of the Accelerator Physics group, Alexander Zhukov, is our point of contact / mentor for all Python-related questions and has been invaluable in this process. Physics group will also be handling the ML side of things once we have optimized our software and taken enough data.

With development of the ABLT routine underway, authored by Carrie Elliott, we needed an interface to establish the parameters of the scan. Initially, however, the various configurations in which operators may want to perform the scan were not yet clearly defined. As a result, there was a need for a GUI that could adapt flexibly and be readily modified as our understanding of the necessary conditions evolved. To address these considerations, the choice fell upon PyQt5 as the preferred framework. Its modular

nature and ease of editing perfectly aligned with our evolving needs, allowing us to not only incorporate the required elements as they became apparent but also refine the presentation to best suit our workflow and preferences.

Automated Beam Loss Tuning Setup Run Setup Filtering	Setup	CCL_PhaseScan_RemoveNewRTBT_20230814_144107	Run
Standard Deviation 1 Magnet Window (Amps) 10.0 Nagnet Window (Amps) 10.0 Nagnet Steps (Amps) 2.0 Phase Window (Degrees) 2.0 2.0 Phase Steps (Degrees) 0.1 Current Filter Current Filter Current Filter Selected Rows Selected Rows BLMs Magnets Phases	 Parameters: Window size / area to be scanned 	Activation Sum	• Live plot of the calculated activation sum shows entire history of scan vs just a

ect devices:	Ditt		
All BLMs	BLMs	Weights	mit
60 Pulse	SCL_Diag:BLM00:Slow60PulsesTotalLoss	26360.338442947865	85
 Linac BLMs DTL BLMs 	SCL_Diag:BLM01b:Slow60PulsesTotalLoss	55657.66609590866	85
CCL BLMs MB SCL BLMs	SCL_Diag:BLM01c:Slow60PulsesTotalLoss	45302.050310572864	85
HB SCL BLMs	SCL_Diag:BLM02b:Slow60PulsesTotalLoss	6682.828584216023	85
BLMS Before Arc BLMs After Arc Ring BLMs Injection Collimation	SCL_Diag:BLM02c:Slow60PulsesTotalLoss	6182.883176049464	85
	SCL_Diag:BLM03b:Slow60PulsesTotalLoss	6316.1304253940825	85
Extraction RTBT BLMs	SCL_Diag:BLM03c:Slow60PulsesTotalLoss	2711.998661473523	85
All Magnets Linac Magnets 	SCL_Diag:BLM04b:Slow60PulsesTotalLoss	3587.6048808256796	85
MEBT Magnets CCL Magnets	SCL_Diag:BLM04c:Slow60PulsesTotalLoss	1849.80210639297	85
MB SCL Magnets HB SCL Magnets	SCL_Diag:BLM05b:Slow60PulsesTotalLoss	2397.5532206781913	85
HEBT Magnets Ring Magnets	SCL_Diag:BLM05c:Slow60PulsesTotalLoss	1676.5978748829245	85
All Phases	SCL_Diag:BLM06b:Slow60PulsesTotalLoss	1322.810210615216	85
DTL Phases	SCL_Diag:BLM06c:Slow60PulsesTotalLoss	240.22997909718475	85
MB SCL Phases HB SCL Phases	SCL_Diag:BLM07b:Slow60PulsesTotalLoss	852.6976829410489	85
	SCL_Diag:BLM07c:Slow60PulsesTotalLoss	596.8249405722204	85
	SCL_Diag:BLM08b:Slow60PulsesTotalLoss	413.54686633388826	85
	SCL_Diag:BLM08c:Slow60PulsesTotalLoss	1726.5343495359239	85
	SCL_Diag:BLM09b:Slow60PulsesTotalLoss	2229.0232083652136	85
	SCL_Diag:BLM09c:Slow60PulsesTotalLoss	949.4324045407637	85
	SCL_Diag:BLM10b:Slow60PulsesTotalLoss	1485.708075209216	85
	SCL_Diag:BLM10c:Slow60PulsesTotalLoss	1390.3985507246377	85

Filtering:

- Selecting a group from the tree will auto filter by that group
- Text input field:
 - Applied on top of tree filter, if selected
 - \circ Single space = logical 'and'
 - \circ Pipe character = logical 'or'
 - Custom filtering supported via override of the 'filterAcceptsRow' method of the
 - 'QSortFilterProxyModel' class

or_expressions = filter_expression.split('

or_expression in or_expressions:
<pre>or_expression = or_expression.strip()</pre>
if not or_expression:
continue
and an and a stand of the second s

- Standard deviation
- Soft BLM limit, hard limit determined dynamically
- Loss tuning vs Data only

Selectable devices:

- 60 pulse BLM signals
- Single pulse BLM signals
- Quadrupole magnets
- MEBT SCL phases
- Clicking 'Initialize' loads the scan parameters as well as auto switches to the Run tab.

Safety features:

- Minimum / maximum window and step sizes hard coded, GUI will not accept values outside of these ranges
- Setup combo boxes will only change their respective values if the appropriate tab is selected and device rows are highlighted
- Locked to a single instance on a specific operator interface (OPI) to prevent multiple scans running simultaneously

Convenience features:

- Save / Load configurations, date and time added to filename automatically
- Curated groups of devices selectable from the tree, does not auto check however, must hit 'Select current filter' button



Future plans:

• Pause button

- Rewrite in PySide6, we have collectively decided to start developing in PySide vs PyQt
- Move application to a softIOC, this should make it more reliable overall as well as prevent the need to lock it to a single instance via software
- Real time data visualization, currently all done post scan
- General aesthetic improvements, suggestions welcome!
- Gain permission to tune losses during production
- Ultimately, incorporate machine learning

******* For detailed information on the scan routine, please consider attending Carrie Elliott's presentation "Development of an Automated Beam Loss Tuning Application in a High-Power Accelerator" on

Lessons learned:

- People WILL hit the 'run' button more than once, especially if a scan is in progress. Disable it until the scan is complete.
- Just because you CAN build it, does not necessarily mean you SHOULD build it, sometimes less is more
- Python wrappers for Qt are awesome

Thanks for reading!

- Informational statements, warnings, and other information printed to the text browser
- Multiple iterations of the same scan can be performed if desired. Once the previous thread closes, another instance will be created with the same scan configuration.
- Click 'Run' once ready. Stop button not shown here but featured in a later version.

if all(self.matches_any_keyword(source_row, source_parent, and_exp) for and_exp in and_expressions)

ef matches_any_keyword(self, source_row, source_parent, keyword) regex = QRegExp(keyword, Qt.CaseInsensitive, QRegExp.RegExp) index = self.sourceModel().index(source_row, 0, source_parent) data = self.sourceModel().data(index, Qt.DisplayRole) return regex.indexIn(data) != -1



device groups from the tree

• Next scan can be configured without





*This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-000R22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access- plan).