

AUTOMATED RF CONDITIONING ON CLARA

Matthew King^{1,2}, Nasiq Ziyani^{1,2}, Anthony Gilfellow^{1,2}, Bradley Wilson^{1,2}

¹ASTeC, STFC, Daresbury Laboratory, Keckwick Lane, Warrington, UK

²Cockcroft Institute, Daresbury Laboratory, Keckwick Lane, Warrington, UK

Abstract

In this paper we present the design and deployment of the software tools used for the automated parallel conditioning of 7 RF cavities on CLARA (Compact Linear Accelerator for Research Applications). The software tools consist of 4 distributed tools with the following responsibilities: Control, Acquisition, Visualisation, and Communication. In combination, these tools enabled the round-the-clock conditioning of all RF cavities with minimal operator intervention, achieving operating parameters for beam commissioning activities.

INTRODUCTION

CLARA is a high-brightness electron beam facility at STFC Daresbury Laboratory, aiming to deliver ultra-short electron bunches to a wide range of user experiments [1]. CLARA has recently resumed operations after a major upgrade, and is currently undergoing beam commissioning [2] at its nominal energy (250 MeV) bunch charge (250 pC) and repetition rate (100 Hz). Prior to beam commissioning, a conditioning programme was conducted on CLARA for its S-Band Photoinjector [3], 4 Linacs, X-Band Lineariser Cavity [4], and Transverse Deflecting Cavity. After the successful proof of automated conditioning using No Operator Automated RF Conditioning (NO-ARC) [5] for the photoinjector [6] at a repetition rate of <50 Hz, it became time to redesign the suite of tools such that the unstaffed conditioning of 7 cavities could happen in parallel at the higher repetition rate of 100 Hz. This would ultimately allow 24/7 conditioning in a consistent and safe manner and reduce the overall impact of conditioning on the CLARA commissioning schedule.

The primary focus of the redesign was to modularise the codebase into specific services with the following responsibilities: Conditioning, Communication, Data Acquisition, and Visualisation. The previous iteration of NO-ARC had consolidated these responsibilities into a single application which meant that an error in the Operator Interface (Visualisation) could result in the conditioning algorithm being unable to run. These proceedings provide details on the constituent services of the NO-ARC suite and how they were combined to deliver operational cavities to CLARA commissioning activities and experimental users.

CONTROL AND COMMUNICATION

The NO-ARC suite comprises four cooperating applications (see Fig. 1): the conditioning FSM, a central REST API, a data logger, and an operator interface (MetaGUI). One FSM instance and one data logger (DL) instance are deployed per cavity, whilst the API and MetaGUI are shared

across all cavities. All inter-application communication is routed through the API, which serves as both a message broker and an in-memory data store. The API is built on FastAPI [7], with a MongoDB [8] backend providing persistent storage for breakdown and new power data.

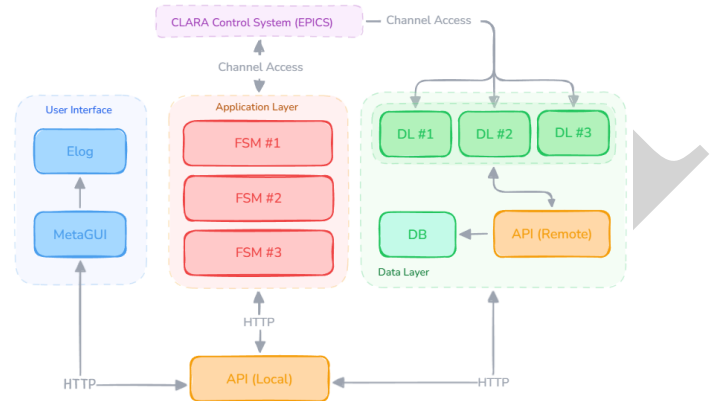


Figure 1: Schematic of NO-ARC suite

State Reporting

Upon initialisation, each FSM instance registers itself as active for the corresponding cavity via the API, and is cleared on shutdown. This mechanism prevents multiple conditioning instances from operating concurrently on the same cavity.

During the runtime of each FSM the current state, previous state, runtime, and conditioning parameters are reported to the API, with additional updates issued upon state transitions. This transient data is stored in memory on the API and shared with the MetaGUI for operator display and for composing automated electronic logbook (elog) entries on state transitions of interest.

Operator Interaction

On startup, any configuration changes are automatically detected and recorded in the elog, ensuring that updates to the conditioning strategy (for example longer soaking periods) are captured without requiring manual intervention.

The MetaGUI provides a graphical interface for monitoring and controlling all active FSM instances, retrieving the latest FSM states via the API to update the display. Operators may terminate conditioning for an individual cavity via the MetaGUI which is sent via the API to the FSM for that cavity. This will result in immediate removal of forward power and transition to a controlled shutdown sequence. Loss of MetaGUI connectivity is treated equivalently and triggers the same shutdown behaviour.

CONDITIONING ALGORITHM

The conditioning algorithm is implemented as a configurable finite state machine (FSM) responsible for incrementally raising each cavity's forward power to operational levels while detecting and responding to cavity breakdowns. One instance of the FSM is deployed per cavity, interfacing with the control system via EPICS Process Variables (PVs). The same algorithm is applied in each FSM instance, with parameters defined in per-cavity YAML configuration files such as power targets, ramp rates, recovery delays, vacuum thresholds, and breakdown tolerances as well as cavity-specific EPICS PVs for interfacing with the control system. Below we detail the states of the FSM, their actions, and the various transitions between states.

Ramping

Two distinct ramping regimes are employed: a linear ramp used for steady-state power progression, and a logarithmic ramp used during initialisation and breakdown recovery. The cavity forward power was controlled via the CLARA RF API [9] which provided simple EPICS PVs to control power setpoints in MW, and also PVs to track the status for when the power was achieved by the RF API. During steady-state operation, the linear ramp increases power in discrete steps towards the next target power (See Normal Ramp section of Fig. 2), with the option for this power-delta to take place over a series of pulses. Above a configurable low-power threshold, each step applies a fixed power delta. Below this threshold, the step size is instead set as a fraction of the current power, ensuring that increments remain small relative to the operating level — where a fixed delta would otherwise represent a disproportionately large step.

During initialisation and following a breakdown event, a logarithmic ramp (see Fig. 2) is used to gradually approach a target power level (see breakdown events in Fig. 2). At ramp step s , over a configurable total of N steps, the requested power is given by

$$P_{\text{next}} = P_{\text{target}} \cdot \log_N(s).$$

On initialisation, P_{target} is set to a configured initial value. As conditioning progresses and higher power levels are reached, this target is updated to reflect the new frontier power. Following a breakdown, the logarithmic ramp restores to this level and the FSM transitions to SOAK.

Soaking

Progression between ramp steps is governed by a SOAK state, during which the cavity is held at constant power setpoint to assess stability. Three conditions must be satisfied before further power increases are permitted. First, a minimum number of RF pulses must be delivered without a breakdown event. Second, the number of breakdowns observed within a configurable sliding pulse window must remain below a defined tolerance. If there are more breakdowns in the pulse window than allowed, a penalty number of pulses are added to the window and the FSM remains in

SOAK until the pulse window is clear of breakdowns, ensuring a sufficient period of stable operation is observed before power ramping resumes (see Fig. 2).

Lastly, the vacuum systems associated with a given cavity must be reading back pressures that are below the thresholds stated in the configuration file. If the vacuum levels are above the threshold, the FSM will wait until they are all below the stated threshold before entering the RAMP state.

Breakdown Detection and Recovery

Cavity breakdown events are inferred by the FSM through two mechanisms. The first is an RF trace violation, detected by a dedicated breakdown detector at machine repetition rate. The same breakdown detection method as used in the previous NO-ARC version [5] is implemented on a RedPitaya system connected directly to the RF system of the cavity. On detection, an attenuation is applied to the incoming RF, which in effect stops the power from reaching the cavity. The RedPitaya will stay in this state until reinitialised by the FSM's state transition to Breakdown Recovery. The second breakdown event is a vacuum spike, which occurs when any monitored vacuum PV exceeds its configured pressure limit. This is a slower (detectable at <5 Hz) and more secondary mechanism for breakdown detection.

Upon either event, the requested power is immediately set to zero. A hold period is then applied, during which diagnostic data is captured (see Data Acquisition and Visualisation), followed by a recovery procedure. For trace violations, the breakdown detector is reinitialised prior to recovery. The FSM then applies the logarithmic RAMP, restoring the cavity to frontier power, after which the SOAK/linear-RAMP cycle resumes. For vacuum spikes, recovery is deferred until the pressure returns below the configured threshold.

Fault Handling and Safety

The fault-handling path is distinct from breakdown recovery as it's responsible for responding to abnormal system conditions and subsystem failures. A range of signals and services are continuously monitored across all conditioning states, including heartbeats from hardware and NO-ARC systems, as well as operating conditions such as modulator states, LLRF system limits, and vacuum system interlocks.

If any monitored condition becomes invalid, cavity power is immediately cut and the FSM transitions to a FAULT state. A dedicated alarm PV, assigned per subsystem, is set to identify the cause and notify operators.

Recovery from faults requires restoration of all monitored subsystems and interlocks externally to NO-ARC. Upon recovery, the FSM returns to the initialisation (INIT) state, restarting the logarithmic ramp procedure. However, critical subsystems that fail to recover within a given time initiate a controlled shutdown, enabling safe unstaffed and overnight conditioning.

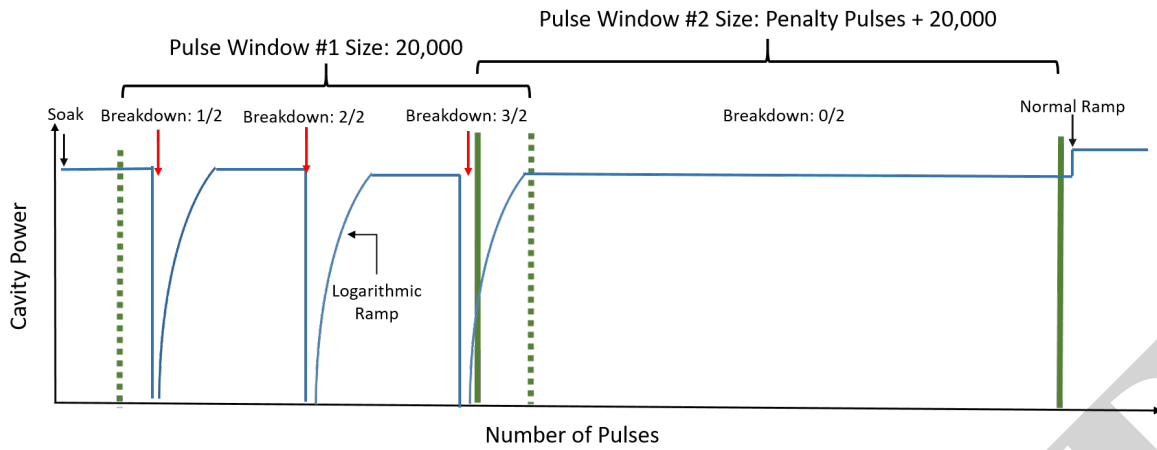


Figure 2: Diagram of exceeding breakdown (shown by drop in cavity power) tolerance of 2 breakdowns per 20,000 pulses within the initial pulse window (dashed green line) during a soak phase resulting in a number of penalty pulse being applied to the next pulse window (solid green line).

DATA ACQUISITION AND VISUALISATION

The dedicated data logger (DL) operates as an event-driven acquisition program, continuously monitoring EPICS PVs and acquiring data in response to conditioning events. On each breakdown event, the DL captures the instantaneous values of all relevant breakdown-related metrics, including RF trace waveforms from the breakdown detector, and forwards them to the API for persistent storage.

In addition, at each ramp in power, the DL records a range of conditioning parameters, including forward power, cavity electric field and scaled breakdown rate. This approach provides a compact representation of conditioning progress, enabling RF scientists to analyse long-term trends and refine conditioning strategies.

The API provides query endpoints over both collections, supporting date-range filtering on a per-cavity basis. For breakdown records, waveform traces from the most recent events can be retrieved directly. For frontier power records, retrieval of measured forward power as a function of total pulse count over a configurable pulse window is provided. To manage the volume of data returned over long conditioning runs, optional pulse-count binning is supported, reducing the dataset for users on request. The scaled breakdown rate as a function of cumulative pulse number, providing a normalised measure of breakdown activity [10] is also directly accessible from the API.

The structured nature of the stored data, combined with the query flexibility of the API, enables RF scientists to extract and analyse conditioning history directly as is shown in Fig. 3. This has supported iterative refinement of conditioning strategies at CLARA, by observing long-term trends in breakdown rate and frontier power progression informing parameter adjustments.

CONCLUSIONS

In these proceedings, we have discussed the architectural design and deployment of the NO-ARC suite of tools on

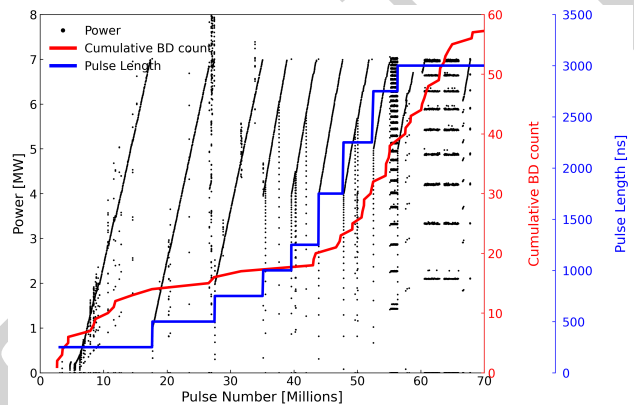


Figure 3: Example of progress plot for the Transverse Deflecting Cavity conditioning created from NO-ARC API data queries.

CLARA: Conditioning Algorithm, Communication/Control, and Data Acquisition/Visualisation, as well as the details of each service and how they were combined to enable the conditioning of several cavities at 100 Hz in parallel with minimal operator intervention. The separation of responsibility for each service reduced the impact of any one service failure on conditioning progress, which ultimately led to an increased robustness in the system overall. With the successful conditioning of CLARA cavities to operational powers, the majority of the suite is now under maintenance.

FUTURE PLANS

Our future work is dedicated to improving the accessibility of the data collected during the conditioning run. We have already started work on providing cloud-based mirrors to both the database and the API such that conditioning data for the CLARA cavities can be queried externally. Though the data are currently being used to assess the conditioning run, we believe that the methods for accessing and visualising the data could be greatly improved so that non-expert users can more easily probe this rich source of data.

REFERENCES

- [1] J.A. Clarke, D. Angal-Kalinin, N. Bliss, R. Buckley, *et al.*, “CLARA Conceptual Design Report”, *J. Instrum.*, vol. 9, p. T05001, 2014. doi:10.1088/1748-0221/9/05/T05001
- [2] D. Angal-Kalinin, M.A. Johnson, J.K. Jones, T.H. Pacey, *et al.*, “CLARA Commissioning and First Friendly User Experiments”, presented at IPAC'26, Deauville, France, May 2026, paper TUO2M05, this conference.
- [3] J.W. McKenzie, L.S. Cowie, P. Goudket, B.L. Militsyn, *et al.*, “High Repetition Rate S-band Photoinjector Design for the CLARA FEL”, in *Proc. FEL'14*, Basel, Switzerland, Aug. 2014, pp. 889-892.
- [4] L.S. Cowie, A.D. Brynes, J.K. Jones, *et al.*, “An X-Band Lineariser for the CLARA FEL”, in *Proc. IPAC'18*, Vancouver, Canada, Apr.–May 2018, pp. 3848-3851. doi:10.18429/JACoW-IPAC2018-THPAL084
- [5] L.S. Cowie and D.J. Scott, “RF Conditioning of the CLARA 400 Hz Photoinjector”, in *Proc. IPAC'19*, Melbourne, Australia, May 2019, pp. 2067–2070, doi:10.18429/JACoW-IPAC2019-TUPTS065
- [6] D. Angal-Kalinin, A. Bainbridge, A.D. Brynes, *et al.*, “Design, Specifications, and First Beam Measurements of the Compact Linear Accelerator for Research and Applications Front End”, *Phys. Rev. Accel. Beams*, vol. 23, p. 044801, 2020. doi:10.1103/PhysRevAccelBeams.23.044801
- [7] S. Ramírez, “FastAPI: High Performance, Easy to Learn, Fast to Code, Ready for Production”, software framework, version 0.115, 2018. <https://github.com/fastapi/fastapi>
- [8] MongoDB Inc., “MongoDB: The Developer Data Platform”, document-oriented database, 2009–present. <https://www.mongodb.com>
- [9] N. Joshi, A. Wheelhouse, J. Wilson, and W. Edward-Olaniyi, “RF-API: A Modular Control Framework in EPICS for RF Systems at CLARA”, presented at IPAC'26, Deauville, France, May 2026, paper WEP6104, this conference.
- [10] A. Gilfellow, N. Joshi, A. Wheelhouse, *et al.*, “High Power Conditioning of CLARA's RF Structures”, presented at IPAC'26, Deauville, France, May 2026, paper THP2144, this conference.