

## Implementation of PASTA for RF Phase Scan Optimization at KOMAC

Myeong-Jin Jung\*, Seunghyun Lee, Jae-Ha Kim, Sung-Yun Cho, Young-Gi Song

Korea Multi-purpose Accelerator Complex, Korea Atomic Energy Research Institute, Gyeongju 38180

\*Corresponding author: mmj@kaeri.re.kr

### 1. Introduction

Beam simulation is essential for designing, operating, and optimizing particle accelerators. In particular, setting the correct phase and amplitude of RF cavities is a key factor for acceleration efficiency and beam quality. At KOMAC, RF phase scans are performed by adjusting parameters step-by-step and observing the beam energy response. This approach can be time-consuming and requires repetitive tasks, which reduces the efficiency of beam services provided to users. It also makes it difficult to respond quickly to various operational changes that occur during machine operation. To solve these problems, we adopted PASTA (Phase and Amplitude Scan and Tuning Application), an OpenXAL-based application, for the KOMAC system.

### 2. PASTA Customization for KOMAC

#### 2.1 PASTA Build

The PASTA application was implemented using the open-source code from the European Spallation Source(ESS) GitLab. Since PASTA inherits its core architecture from the OpenXAL framework, the OpenXAL source code was first compiled to establish the necessary hierarchical dependencies. To resolve build failures, we reconfigured the dependency versions in the pom.xml files. This adjustment was necessary because certain external libraries were no longer accessible or were incompatible with the current repository settings. The JDK version was adjusted to satisfy the specific compatibility requirements of both the OpenXAL and PASTA source code.

#### 2.2 Modeling of the KOMAC Accelerator Structure

To implement PASTA within the environment of KOMAC, it is essential to provide dedicated configuration files that describe the physical structure of the accelerator. Accordingly, the accelerator structure was modeled using the OpenXAL lattice format. This architecture consists of Sequences, which represent logically connected sections of the beamline, and Nodes, which denote individual hardware elements within those sections. For instance, an accelerating cavity such as DTL101 is defined as an independent sequence, containing specific nodes like RF Gaps and Quadrupole Magnets to represent its internal structure. Beam Position Monitors (BPMs) were intentionally separated into a

distinct sequence. Specific physical attributes were assigned to each node to enable precise beam dynamics calculations. For the RF Gaps, the actual electric field distribution was reflected by determining the gap offset and length based on PARMILA experimental data. For the quadrupole magnet nodes, the design field strength and field direction were defined. Figure 1 shows the accelerator lattice layout of KOMAC.

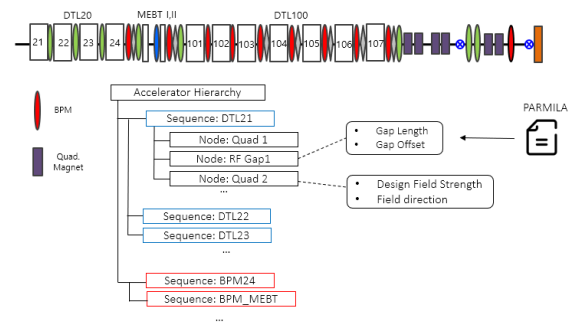


Fig. 1. The Accelerator Lattice layout of KOMAC.

Reflecting KOMAC's EPICS-based control environment, Handle-Channel Binding was performed to directly link each element with its corresponding Process Variable (PV). Additionally, the beam kinetic energy at the entrance of each sequence was specified, enabling the precise tracking of beam characteristics throughout the different sections of the accelerator. This comprehensive configuration, including specific node attributes and channel bindings, successfully replicates the operational environment of the KOMAC proton linac within a simulation.

#### 2.3 EPICS-Based Virtual Testbench

To verify the control logic of PASTA and prevent potential operational risks, an EPICS-based virtual testbench was established. An Input Output Controller(IOC) was created to host a standalone control environment independent of the actual accelerator.

Virtual Process Variables (PVs) were generated to simulate the accelerator's response behavior. The core of this testbench is a data feedback loop utilizing Analog Output (AO) and Analog Input (AI) records. Specifically, when PASTA updates the phase or amplitude setpoints via the AO record, it triggers an internal logic within the IOC to retrieve corresponding beam responses from a pre-recorded CSV file containing DTL experimental data. These values are then reflected in the AI record, which PASTA reads as real-time beam data. This mechanism

ensures that the application operates under conditions nearly identical to the actual machine environment.

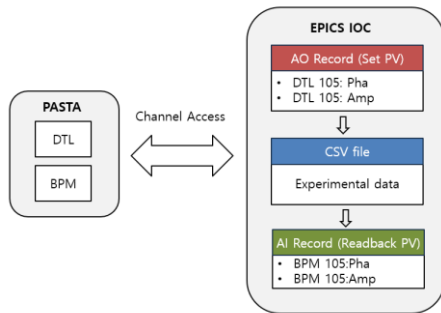


Fig. 2. Schematic of the EPICS-based testbench

The operational workflow of PASTA proceeds in four main stages. First, the PASTA application is launched via a JAR file, and then the user selects the XML lattice file representing the KOMAC accelerator structure. Second, the target DTL cavity and downstream BPM nodes are selected, mapping each element to its respective virtual PV. Third, the Phase Scan is performed (Fig. 3). During this stage, the previously described data feedback loop provides the necessary response curves based on actual experimental results. Finally, the collected data is compared with model predictions. The model's phase response curve is fitted to the measured data to derive the optimal phase and amplitude for actual operation.

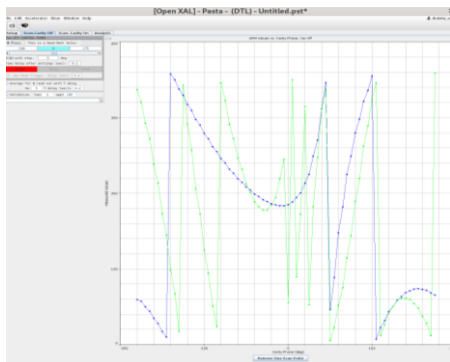


Fig. 3. Phase Scan panel of the PASTA application.

### 3. Conclusions

PASTA was implemented to automate phase scan procedures for the accelerator. To verify its functionality, a virtual testbench was established. Using a Soft IOC, the control logic was successfully validated against experimental data. This configuration confirmed that PASTA effectively performs phase scans within a virtual environment. Future work will involve connecting PASTA to actual accelerator PVs to evaluate its performance during real-time operation.

### REFERENCES

- [1] ESS GitLab, <https://gitlab.esss.lu.se/ess-crs>
- [2] J. Galambos et al., "Pasta - an RF Phase Scan and Tuning Application," Proceedings of the 2005 Particle Accelerator Conference, Knoxville, TN, USA, 2005, pp. 1491-1493
- [3] Open XAL reference manual, <https://openxal.github.io/>
- [4] Experimental Physics and Industrial Control System (EPICS), <https://epics.anl.gov/>