

DEVELOPMENT OF THE RF PHASE SCAN APPLICATION FOR THE BEAM ENERGY MEASUREMENT AT KOMAC*

SungYun Cho[†], Jeong-Jeung Dang, Jae-Ha Kim, Young-Gi Song

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

Abstract

The Korea Multi-purpose Accelerator Complex (KOMAC) has been operating the 100MeV proton linear accelerator. The output beam energy from each drift tube linac (DTL) can be changed by the operation RF phase. The KOMAC linac is consisted of 11 tanks that need to proper setting of RF phase. The phase scan application was developed on the Java eclipse. On the other hand, the analysis program was developed on the Matlab. Since the data analysis processes as soon as finished the scan processing, the application integration need. This paper describes the implementation of the integrated application based on python and Experimental Physics and Industrial Control System (EPICS).

INTRODUCTION

The KOMAC control system has been operating based on the EPICS framework [1]. The distributed system that is an input output controller (IOC) controls the subsystem for each local system. The process variables (PVs) are operation parameters for control and monitoring from the IOC. Several programs can be access to EPICS IOC's parameters by channel access (CA). The RF phase scan application has been developed based on python and can be access to IOC using pyepics library [2]. The scan I/O variables were designed using that. The data processing step has been developed based on python numba library [3]. A scan step and data processing are described separately in order. The below figure is the schema regarding the KOMAC control system.

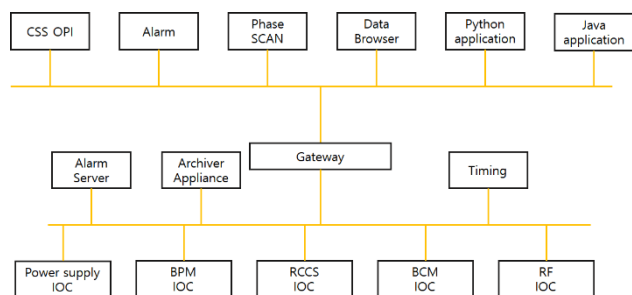


Figure 1: KOMAC distributed control system.

THE STRUCTURE OF THE PHASE SCAN APPLICATION

The basic structure of the application is based on Python Display Manager (PyDM) [4]. The PyDM is a building user interfaces tool based on PyQt5 with plugged in pyepics. The PyDM uses Signal & Slot for calling events. The

I/O variables are connected with Slot functions. Some calculation function are developed based on numba. Figure 2 shows the working structure of the phase scan application.

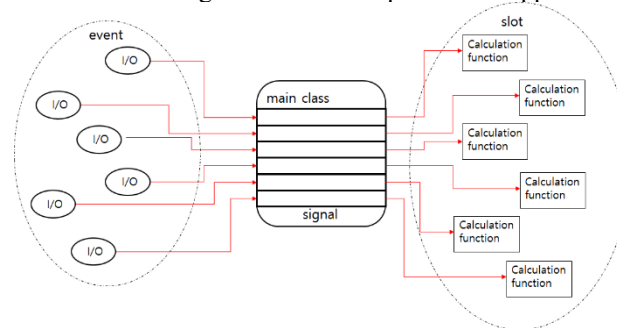


Figure 2: The mechanism of the application.

SCAN CODE DEVELOPMENT

The moving parameter is a phase set value that is different for each tank. After the parameters regarding scan settings are determined, the phase measure experiment will start. The measured phases are from the prior tank and current tank. So the moving parameter is just one and the monitoring parameters are two. Figure 3 shows the concept of process.

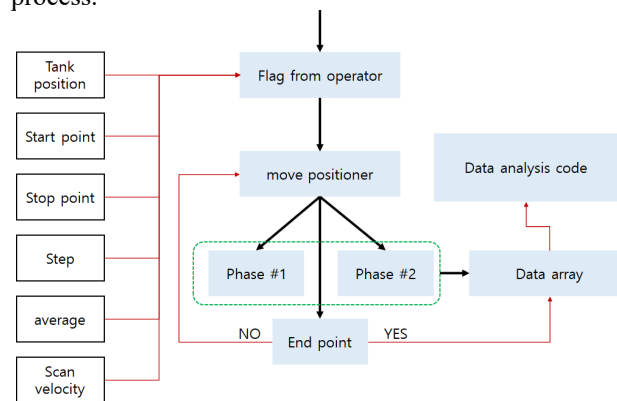


Figure 3: The schematic of scan algorithm.

The original program was developed by the Java language. Since the scan mechanism has to be transplanted to the new program, the EPICS sscan record that's similar to the original algorithm was selected [5]. The sscan record has a function to move positioners and record detector data at each of the positions. The positioner is the moving parameter for the phase setting. When the EXSC field that's a flag signal is executed, sscan record start the process. Many fields work organically during the process. The output signal is sent to positioner PV. If the changing value is finished, the sscan waits the delay time by PDLY field and the detector PV records the target value. The moving step

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2022). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

is determined by the NPTS field. The step can be changed by the custom function using calcout record, since the NPTS is the number of points in a range of between the P1SP and P1EP fields. The monitoring data is saved on the ai record. I/O signals control and monitor the PVs via pyepics library. Figure 4 shows the I/O signals with the IOC.

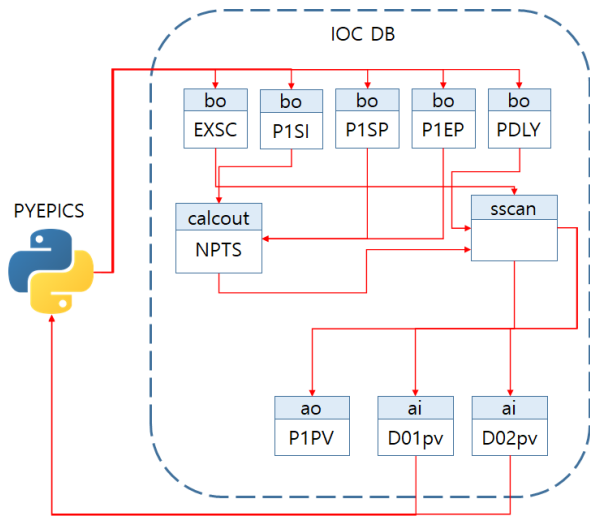


Figure 4: I/O signals from the python application.

The sscan record access to ao record via Channel Access because the set variable works other IOC. When the scan is finished, the monitoring data group is saved to the array and it is output to the python application.

THE DATA ANALYSIS

Figure 5 is overall schematic regarding the system mechanism.

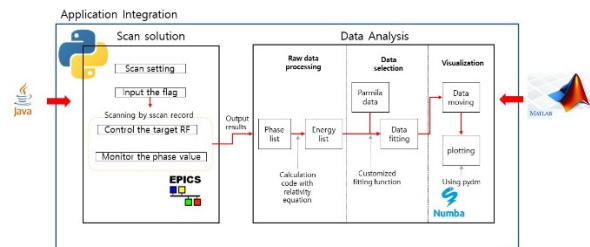


Figure 5: The schematic of the application integration.

The original analysis code was developed by the Matlab. The goal of development is high implementation and speed similar to the original. Because Matlab can provide a rapid calculation process more than python language, the calculation process needs to be rapid platform. The numba library is selected because of it. The numba is a just-in-time compiler for python, it is suited to use with the numpy arrays and functions, and loops.

If the data inputs to the application, the data processing is started. Accumulated data is the phase values that are each target tank and prior tank. The first data processing is that phase values convert to energy data. The velocity of the beam can be calculated from the phase difference. The

energy of the beam can be calculated according to einstein's equation. Figure 6 describes the phase scan method [6].

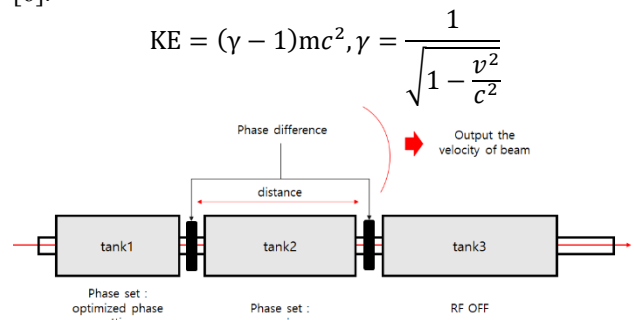


Figure 6: The phase scan signature matching.

The diagnostic device is a beam position monitor (BPM) that has offset value because it isn't synchronous with timing system. So each measurement point has to be calculated considering the offset value between -180 and 180 degree. It is implemented with 360 loops for each step based on numba environment.

After the energy list is accumulated, it can be called measured results. The measured results need to compare with the calculated results in order to determine the proper RF operating point. The calculated results can get from PARMILA calculation. Because the calculated results have to match the number of data for comparison, the data is selected the same RF operation phase with measured results. The optimized RF operation phase can be determined from the minimum chi-square value. Chi-squared test is the method that is determining whether the difference between the observed data with expected data.

$$\chi^2 = \sum \frac{(\text{observed value} - \text{expected value})^2}{\text{expected value}}$$

The calculated results have several data group depending on RF amplitude. The result of chi-square is for each amplitude of parmila data. The results can be fitted to a quadratic equation using scipy library that provides the curve-fit function. Figure 7 shows the result of fitting.

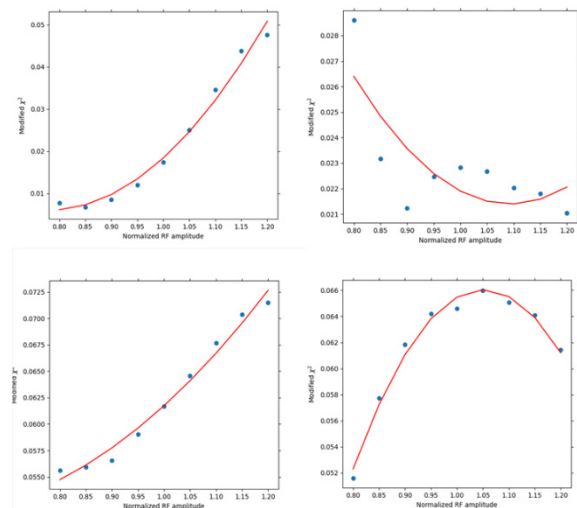


Figure 7: The results of the fitting.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2022). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

However, if the scipy library is used, the calculation speed is slow remarkably. Since the scipy library can't be used in the environment of numba, the custom fitting function needs to be built. The fitting function can be similar to the shape of a general equation. The quadratic gradient can determine whether it is concave or convex. The RF operation phase is selected when the result of fitting is a concave. The key of time reduction is just to do discrimination action in the loop except set up an equation. This method is approximately 200 times rapidly more than scipy function on the numba environment. But the overall result is a little broadly than the original result. Figure 8 shows the result of comparing.

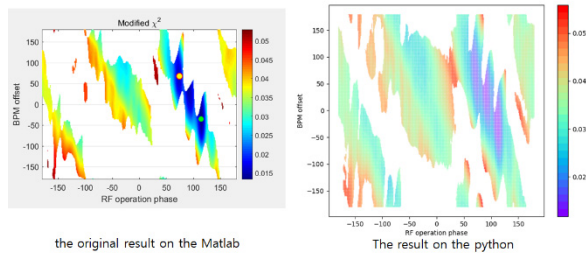


Figure 8: The overall result of the chi-square compared with original result.

Because the purpose of the fitting is determination the minimum chi square, the determining of the optimized point isn't problem on the new calculation code whether the data fitting function was accurate.

THE GRAPHICAL INTERFACE

The graphical interface was implemented using qt-designer. The interface was created using a grid layout for the responsive interface when the size of the display is changed. The scan variables are created using pydm widgets in order to direct use epics access. Figure 9 shows the graphical interface.

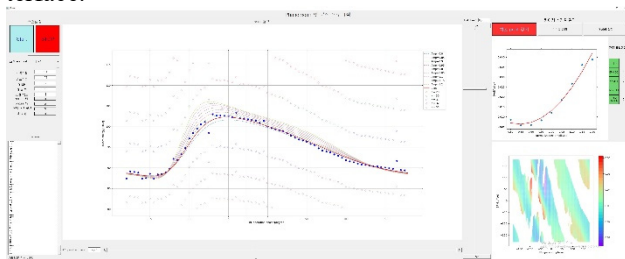


Figure 9: The graphical interface based on pydm.

CONCLUSION

The integrated application is implemented based on python language. The python is selected because of rapid development speed and high scalability. Rapid calculation process is replaced from the matlab function to python numba function and the scan algorithm is replaced from the java scan code to EPICS sscan record. Because the scan and the data analysis was integrated on the one application, the development or the maintenance is more flexibility and the interface is more user-friendly.

Furthermore, the future object of development is implement that's the auto scan function. If this function is implemented, the scan time will be significantly reduced.

REFERENCES

- [1] Experimental Physics and Industrial Control System (EPICS), <http://www.aps.anl.gov/epics>
- [2] Epics Channel Access for Python (PyEpics), <https://cars9.uchicago.edu/software/python/pyepics3/>
- [3] numba, <http://numba.pydata.org/>
- [4] Python Display Manager (PyDM), <https://slacslab.github.io/pydm/>
- [5] The sscan record, <https://epics.anl.gov/bcda/synApps/sscan/sscanDoc.html>
- [6] Han-Sung Kim, Hyeok-Jung Kwon, Seong-Gu Kim, Seok-Geun Lee, Young-Sub Cho, "RF Phase Scan for Beam Energy Measurement of KOMAC DTL", in *Proc. The Korean Nuclear Society Autumn Meeting Conf. (KNS'16)*, Gyeongju, Korea, Oct. 2016.