Contents lists available at ScienceDirect

Nuclear Engineering and Technology

journal homepage: www.elsevier.com/locate/net

Original Article

Design of HUST-PTF beamline control system for fast energy changing

Peilun Li, Dong Li[*], Bin Qin, Chong Zhou, Wenjie Han, Yicheng Liao, Aote Chen

State Key Laboratory of Advanced Electromagnetic Engineering and Technology, School of Electrical and Electronic Engineering, Huazhong University of Science and Technology, Wuhan, 430074, China

A R T I C L E I N F O

Article history:
Received 26 December 2021
Received in revised form
9 February 2022
Accepted 21 February 2022
Available online 23 February 2022

Keywords:
Proton therapy
Beamline control system
Dynamic node update
Fast energy changes

A B S T R A C T

A proton therapy facility is under development at Huazhong University of Science and Technology (HUST). To meet the need for fast energy changes during treatments, a beamline control system (BCS) has been designed and implemented. The BCS coordinates and controls various beamline devices by adopting a distributed architecture divided into three layers: the client, server, and device layers. Among these, the design of the server layer is the key to realize fast energy changes. The server layer adopts the submodule programming paradigm and optimizes the data interface among modules, allowing the main workflow to be separated from the device workflow and data. Furthermore, this layer uses asynchronous, multithreaded, and thread-locking methods to improve the system's ability to operation efficiently and securely. Notably, to evaluate the changing energy status over time, a dynamic node update method is adopted, which can dynamically adjust the update frequency of variable nodes. This method not only meets the demand for fast updates on energy changes but also reduces the server's communication load in the steady state. This method is tested on a virtual platform, and the results are as expected.

© 2022 Korean Nuclear Society, Published by Elsevier Korea LLC. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

Proton therapy uses proton beams to irradiate cancer cells, which have the characteristic of the “Bragg Peak,” allowing them to accurately release the dose at the target while reducing damage to normal cells, thus eliminating side effects [1]. The proton therapy facility typically comprises an accelerator, a beamline, and treatment rooms, which places high requirements on the beamline control system (BCS).

Many medical-use accelerators and beamline devices across the world, such as Center for Proton Therapy at the Paul Scherrer Institute, use the EPICS framework to achieve distributed control [2,3]. The EPICS system is built around a distributed real-time database that can be accessed with operator interface (OPI) using PV variables based on the CA protocol. However, owing to the CA protocol's “transparent access” feature, additional methods must be used to manage control permissions. The Austrian MedAustron Particle Therapy Center adopts a mature commercialization strategy, with WinCC OA as the main platform for control system development, and numerous products provided by mainstream manufacturers (e.g., SIEMENS and NI) [4,5]. This not only guarantees the equipment's reliability but also facilitates the replacement and iteration of software and hardware parts in later maintenance and upgrade. However, the system cost is high. The French Centre Antoine Lacassagne frequently uses open-source frameworks. The back end is primarily developed using C++, and the front end is developed using the Qt framework. This reduces licensing costs while improving the transparency and flexibility of the final product [6]. The China Institute of Modern Physics conducts object-oriented programming based on Python. They developed a physics-oriented accelerator control system (PACS) with Python's extensive module library, making the control system highly scalable and maintainable [7].

A proton therapy facility based on a superconducting cyclotron is under construction at Huazhong University of Science and Technology (HUST-PTF). As shown in Fig. 1, the facility includes a 250 MeV superconducting cyclotron, a beamline, and three treatment rooms [8,9]. The beamline connects the cyclotron to the treatment rooms and contains 106 magnets, beam diagnostic elements, and other devices that are primarily used for beam transportation and energy selection to fulfill the clinical needs.

The main treatment mode of HUST-PTF is spot scanning. The beam energy can be continuously modified in range of 70–240 MeV, provided that an energy change occurs less than 150 ms. In the process of energy change, the nozzle directly controls the kicker magnet in the beamline to cut off the beam, and then the

[*] Corresponding author.
E-mail address: lidonghust@hust.edu.cn (D. Li).

https://doi.org/10.1016/j.net.2022.02.023
1738-5733/© 2022 Korean Nuclear Society, Published by Elsevier Korea LLC. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

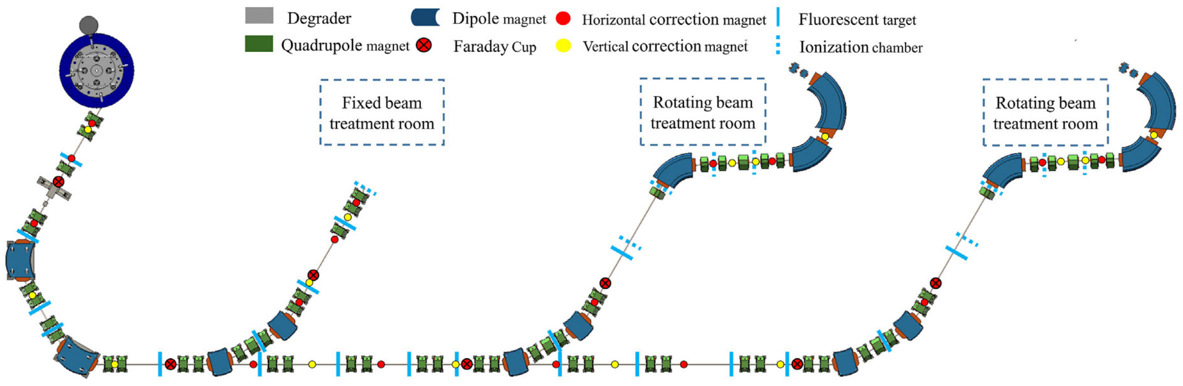
Degrader
Quadrupole magnet
Dipole magnet
Faraday Cup
Horizontal correction magnet
Vertical correction magnet
Fluorescent target
Ionization chamber
Fixed beam treatment room
Rotating beam treatment room
Rotating beam treatment room

Fig. 1. Structure of HUST-PTF.

beamline control system (BCS) controls the magnet power supplies and the degrader according to the treatment requirements. The excitation current adjustment of power supplies and the motion control of the degrader are implemented using their local controllers. The BCS must obtain data, such as the current values of power supplies and position of the degrader, quickly to evaluate if their adjustments have been finished and to realize fast energy changes.

Because the mechanical position movement of the degrader wedges requires longer time than the current adjustment of beamline magnets, we first test the adjustment time of the degrader under BCS control. A typical energy change step corresponds to a water equivalent depth of 5 mm, and the displacement of the degrader corresponding to the typical energy change step is 2.2 mm at lower energy region (70 MeV). When the relative error between the actual and set positions is less than 1 μm for two consecutive times, the adjustment is accomplished. According to Fig. 2, under the condition of maximum displacement, the degrader's adjustment time will reach 144 ms, which matches the design requirements. We also performed tests to validate that the BCS's running time is less than 3 ms and the communication time between the BCS and the degrader is less than 4 ms, both of which have little effect on the total time.

Moreover, many power supplies on the beamline only provide a TCP "command-response" communication method, which can transmit primary variables, such as current and voltage, but not their adjustment status. Therefore, the BCS must continuously scan the current values to evaluate whether its adjustments are accomplished. The specific method is as follows: when the BCS discovers that the current value reaches the set value within an acceptable margin for the first time, it must read the current value again to ensure that the current reaches a steady state exactly; furthermore, while the energy is changed, many power supplies must be controlled simultaneously. For example, the third treatment room must simultaneously regulate up to 61 power supplies. These variables place a considerable load on the communication between the BCS and devices. To satisfy the time requirement of an energy change, adopting a more efficient data update method and improving the BCS's operational efficiency are necessary.

The rest of this paper is structured as follows: Section 2 introduces the BCS's overall design and functions/modules of each control layer. Section 3 introduces the design of the server layer for fast energy changes. Section 4 proposes a method for dynamic node update, which can adjust the update frequency of variable nodes dynamically during energy changes or the steady state. Section 5 contains the results of the dynamic node update method test, and Section 6 draw conclusions.

2. Overview

The BCS adopts an OPC UA distributed architecture with three layers: the device layer, the server layer, and the client layer. As shown in Fig. 3 [10], each layer communicates with another via the OPC UA protocol.

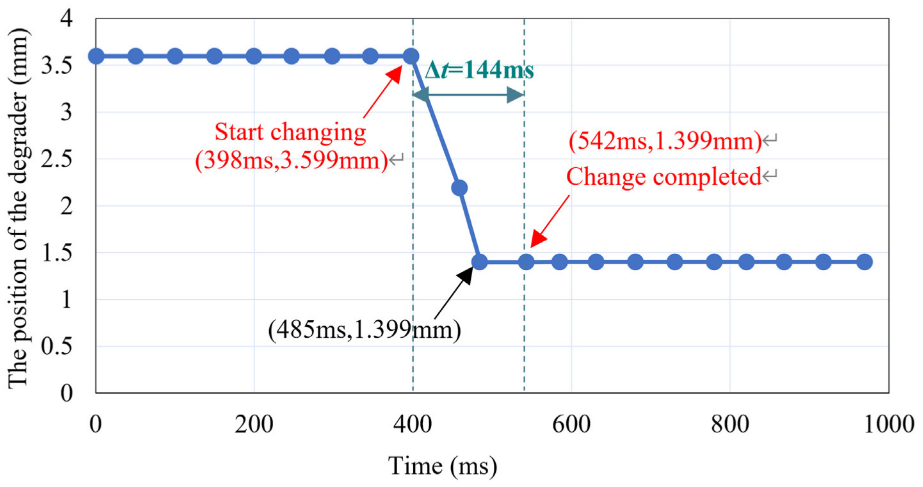

Fig. 2. Adjustment process of the degrader.

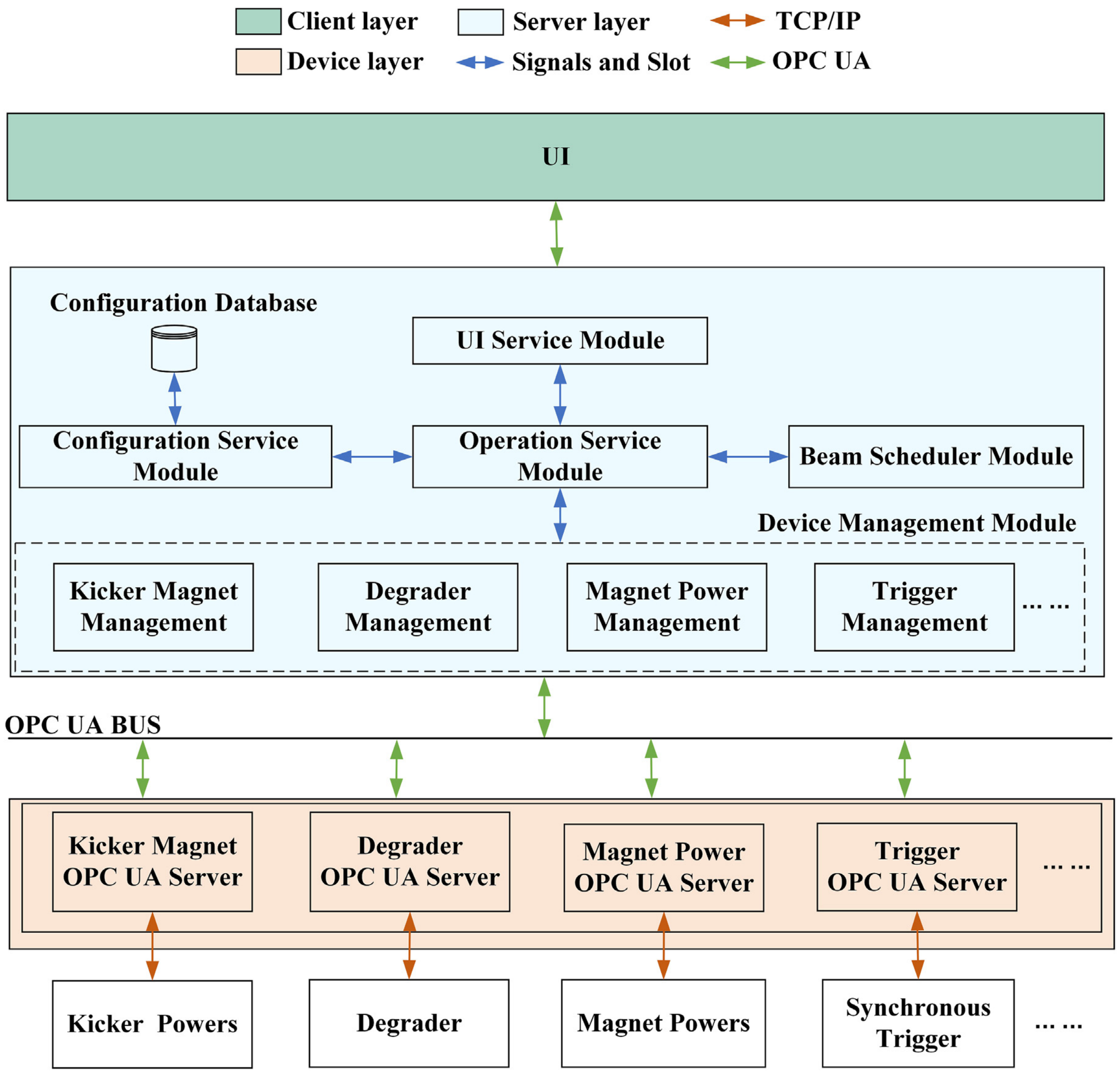

Fig. 3. Structure of the BCS.

The client layer is developed using the Qt framework [11], which also serves as the beamline's GUI. This layer comprises a main interface and several sub-interfaces. The main interface monitors at the master level, whereas the sub-interfaces monitor at the device level.

The server layer, which is the functional core of the BCS, primarily conducts the workflow and maintains the configuration information. To realize the decoupling of modules, the BCS server adopts C++ [12] sub-module programming method. The server layer comprises the configuration service, UI service, operation service, beam scheduler, and device management modules. The configuration service module interacts with the database and manages configuration data; the UI service module provides an interactive interface for the UI; the beam scheduler module interacts with the treatment control system and processes beam request information; the operation service module runs the BCS workflow and dispatches other software modules; and the device management module interacts with specific devices. Separating various modules allows for easy future expansion and maintenance.

The beamline hardware devices and corresponding OPC UA servers are supported by the device layer. Each OPC UA server not only interacts with the relevant device via the TCP protocol but also publishes its control variables on the bus in the form of the OPC UA variables to interact with the server layer. Simultaneously, to ensure the safety of the treatment process, the readable or writable attributes of the OPC UA variable nodes can be set via the UI device module at the server layer, enabling the dynamic control of the device setting authority. During treatment, all variable nodes are set to the read-only status with only the emergency stop function retained; during beam tuning, more beam permissions can be opened to the client layer.

3. Designing the server layer to achieve fast energy changes

As described previously, the server layer is the core of the BCS, which is primarily responsible for running the workflow, including extensive configuration information, and must interact with numerous external systems. Therefore, the server layer's design is the key to achieve fast energy changes.

3.1. Workflow design of the server layer

Considering the functional requirements of the BCS and control features of subsystems, improving operational efficiency is the key to achieve fast energy changes. Therefore, the actual process design adheres to two principles: (1) For operations with tight requirements on the order of operations, the operations are performed in succession. The process moves on to the next operation after completing and validating the previous one. (2) The advantages of computer multithreading are used to achieve parallel execution and to improve the efficiency of the BCS operations for activities that do not have strong sequence restrictions and are time-consuming. Each module runs in its own thread and realizes thread waiting and synchronization using thread locks and condition variables. This not only ensures the system's safety but also improves the system's operational efficiency and allows for the decoupling of various software modules.

As an example, the process of changing energy is shown in Fig. 4. To reduce the energy change time, time-consuming steps, such as modifying the currents of power supplies and position of the degrader, are operated in parallel.

3.2. Architecture design of the server layer

The BCS server is developed in a sub-module way, with each software module implemented as a C++ class object. The operation service module is the most important of the server's software modules. It is the BCS server's “dispatch center” and operates the main workflow. During operation, it must communicate with other software modules on the BCS server to coordinate various devices to complete the workflow simultaneously. Object pointers and callback functions are used in traditional C++ programming to communicate across class objects. To send a message to an object, its object pointer must be obtained first. This method is ineffective for module decoupling and has a negative impact on code readability. This paper is built on the Qt's “signal and slot” technology, which allows for the asynchronous communication and decoupling of class objects. Using this technique results in the decoupling of the operation service module from other modules, as well as the decoupling of the main and device workflows.

Furthermore, to improve the efficiency of the main workflow and to ease future expansion, the data interface for interaction between the main and device workflows is optimized. As shown in Fig. 5, the device's operation status is evaluated in the corresponding device management module. To decide whether to continue treatment, the main workflow only accepts the status evaluation result returned by the associated module. In this manner, the decoupling between the primary workflow and data is also completed, which improves the workflow's operational efficiency even further.

4. Dynamic node update mechanism

During treatment, the energy needs to be changed in accordance with the requirements of treatment. During the energy-changing process, the server should quickly update the OPC UA variable nodes of devices to quickly evaluate their status and inform the main workflow of the energy-changing result. In the steady state, it is envisaged that the update frequency can be adjusted accordingly

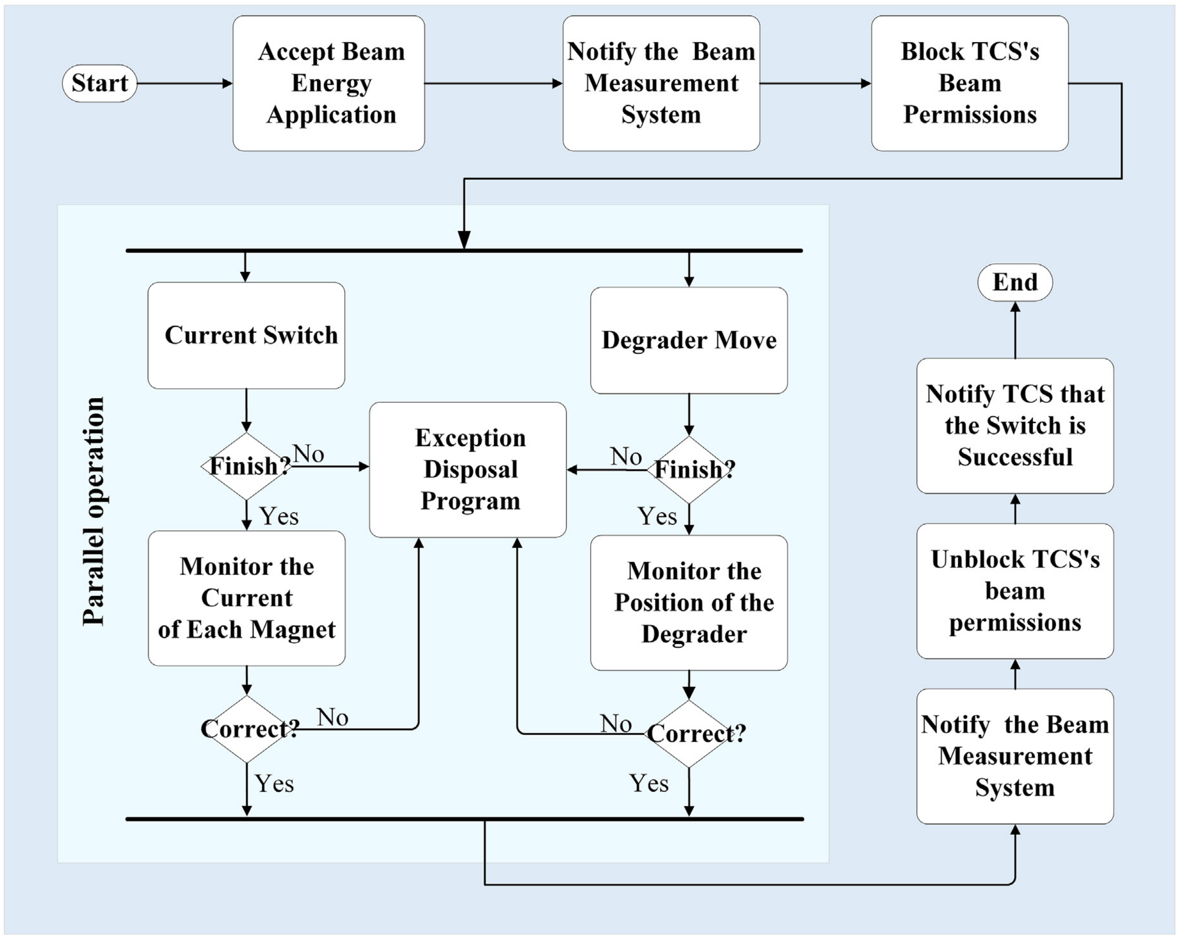
Start
Accept Beam Energy Application
Notify the Beam Measurement System
Block TCS's Beam Permissions
Parallel operation
Current Switch
Degrader Move
Finish?
No
Yes
Exception Disposal Program
Finish?
No
Yes
Monitor the Current of Each Magnet
Monitor the Position of the Degrader
Correct?
No
Yes
Correct?
No
Yes
End
Notify TCS that the Switch is Successful
Unblock TCS's beam permissions
Notify the Beam Measurement System

Fig. 4. Process design of energy change.

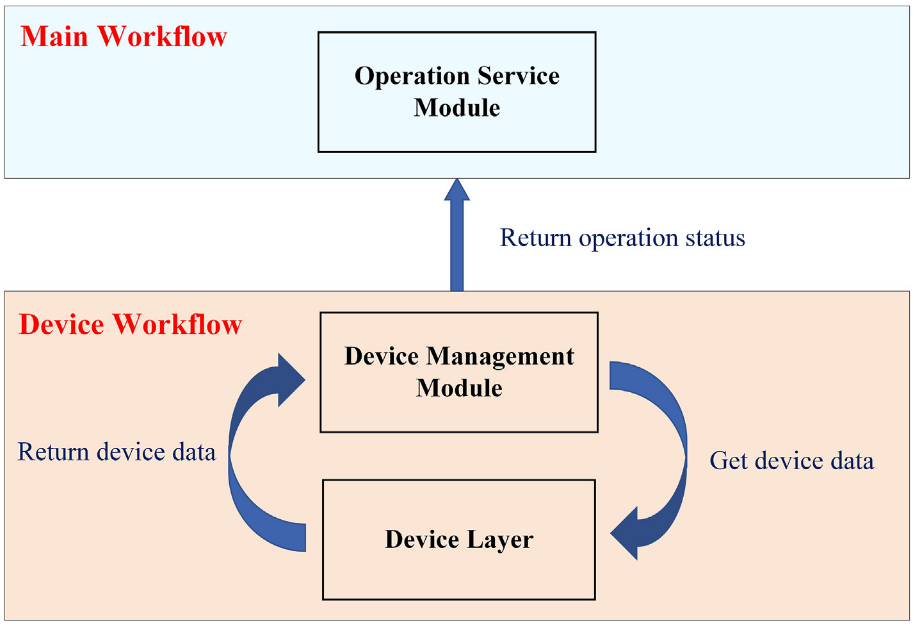

Fig. 5. Optimization of the data interface between the main and device workflows.

to decrease the server's communication load. The OPC UA generally adopts the monitoring method to realize node updates. Its update frequency is insufficient to meet the demand for the fast reading of variable nodes during energy change. Therefore, a mechanism for dynamic node update is adopted in this system.

The update mechanism is shown in Fig. 6: while the system is stable, the BCS monitors the necessary node via the monitor mechanism. When the node is in a monitored state, the client can independently set the update frequency. Therefore, the client can customize the node's update frequency to meet its own requirements. When the system's energy is changing, the BCS uses the ReadValues() function to quickly read the data of the corresponding variable nodes, thereby increasing the node's update rate and obtaining the node's status at a fast speed.

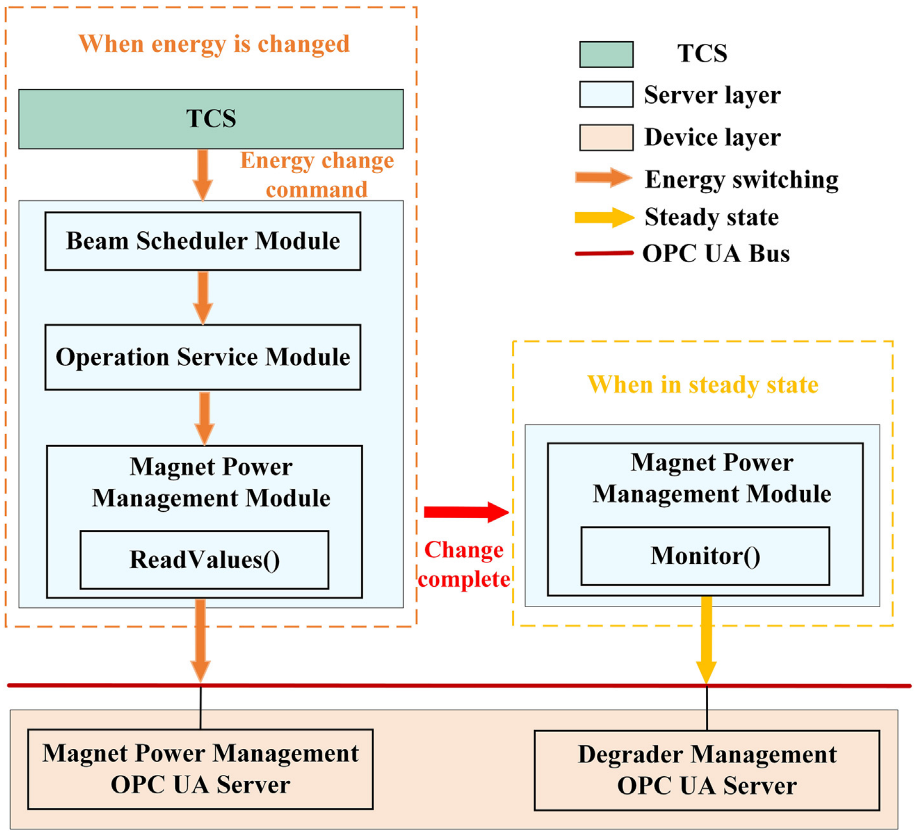

Fig. 6. Dynamic node update mechanism.

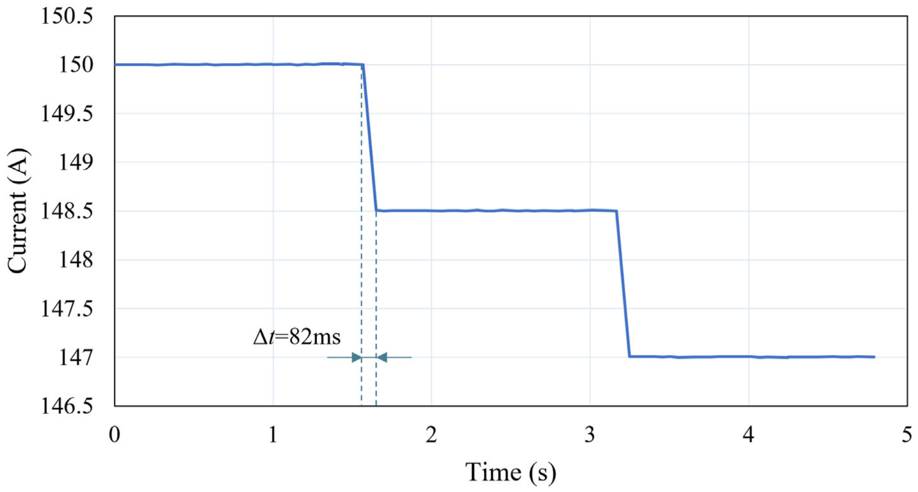

Fig. 7. Current adjustment of a single virtual power supply.

This method not only updates the necessary nodes as soon as the energy changes but also avoids the frequent polling of the server in the steady state, which reduces the communication stress and better meets our needs.

5. Test

In the BCS, the control of power supplies is the main task due to their quantities and the requirement on precise current adjustments, an experiment is designed to test the dynamic update performance of nodes by controlling the magnet power supplies. An experiment is designed on a virtual platform because the power supply is not yet installed at site. According to the actual power supply's test report, the current change completes and enters the steady state at a rate of 1% $*I_N/80$ ms. Further, in the low energy region (70 MeV), the current change corresponding to the typical energy change step is the largest but it does not exceed 1% $*I_N$. Therefore, in the virtual power supply test, the change rate of the virtual power supply current is set to 1% $*I_N/80$ ms and the single current adjustment step is set to 1% $*I_N$. Based on the data and power protocol described above, a virtual power module that meets the requirements is built to substitute the real magnet power supply for subsequent testing.

First, we use a basic power client to test the changing time of a single virtual power supply source. The virtual power supply's rated current is 150 A (I_N), and each change is 1.5 A (1% $*I_N$). Fig. 7 shows the test results. A virtual power supply requires 82 ms to change 1% $*I_N$, of which 80 ms is the time required for the virtual power supply we set to change 1% $*I_N$, 2 ms for the communication time between the client and virtual power supply, and practically 0 ms for the client running time. This result is consistent with the virtual power design goals.

Table 1
The operating environment of the virtual machine.

Software and hardware environment	Attributes
OS	Debian10
CPU	Intel Xeon E5-2650
CPU frequency	2.2 GHz
SCSI disk device	40 GB
RAM	4 GB
Network adapter	Bridge mode
Python	3.8.3

We select the beamline connecting to the third treatment room for the test of node dynamic update performance because it has the most power supplies, up to 61 units. To put the BCS's performance to the test, using the VMware platform, 61 virtual power supplies are placed in the matching 61 virtual machines under the maximum managed load. Table 1 shows the operational environment of each virtual machine.

During this test, the BCS server controlled all 61 power supplies simultaneously. First, through monitoring and fast loop reading, update the “current readback” node 100 times. Because each power supply has the same node update frequency, we may examine the node update frequency of any magnet power supply, as shown in Fig. 8. When the monitoring method is used to update the nodes, the client can independently control the update frequency, with a maximum of approximately 20 Hz. When using the fast loop reading method to update the node, the update frequency can reach approximately 50 Hz and the node update frequency is considerably higher than the former.

This virtual platform is then used to model an energy-changing process for the third treatment room, with the current change being %1 $*I_N$. The update frequency and value change of a power supply's “current readback” node are recorded. As shown in Fig. 9, the BCS server monitors the power supply and acquires the value of the node through the monitoring mechanism before and after the energy change due to the use of the dynamic node update mechanism. However, when the energy is changed, the server updates the node via fast loop reading to determine whether the current changing process is accomplished, considerably increasing the

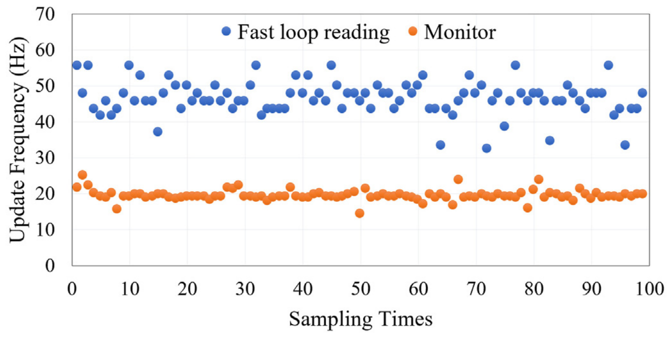

Fig. 8. Comparison of the update frequency of the two methods.

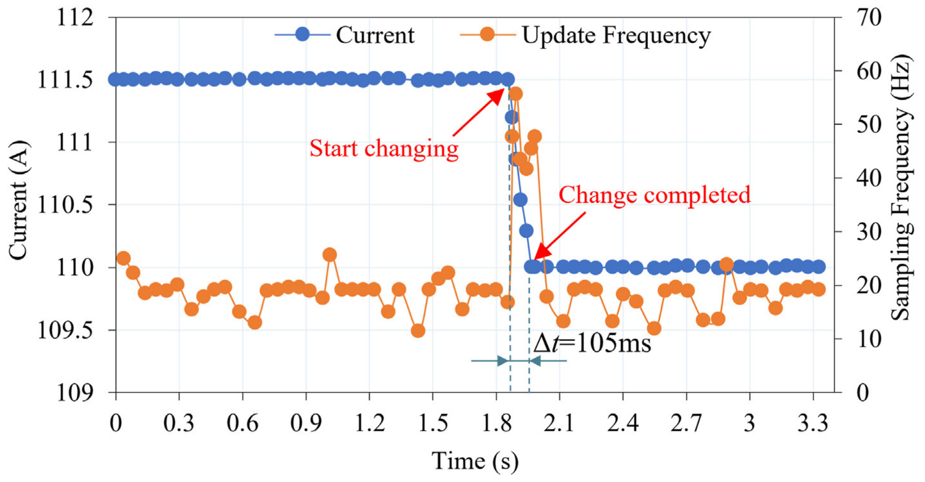

Fig. 9. Values and update frequency of nodes before and after energy change.

update frequency. Furthermore, under the BCS control, the total time for a power supply current change is 108 ms, of which 80 ms is the virtual power supply current change time; 28 ms is the BCS running time consumption and communication time (specifically, the communication time is 21 ms and running time consumption of the BCS is 7 ms).

6. Conclusions

In this study, the BCS is designed and implemented in accordance with the HUST-PTF specifications. To meet the requirement of less than 150 ms for energy changes, the server layer's workflow and architecture are designed and optimized. The system can boost operational efficiency while ensuring safety using asynchronous multithreading technologies to optimize the workflow. The decoupling of the main workflow from the device workflow is achieved in the server architecture design using C++ modular programming and Qt's “signal and slot” mechanism. Simultaneously, the data interface between the main and device workflows is adjusted to improve the workflow's operational efficiency. Notably, a method for dynamic node updating is adopted, enabling the server to quickly update the device variables as energy change. The server switches to the monitoring mode for variable nodes when in a steady state, and the update frequency is reduced. The method is tested on a virtual platform, and the results meet the expectations of the designers.

The BCS is expected to be used in HUST-PTF beam tuning experiments in the near future after additional testing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (No. 2016YFC0105305) and the National Natural Science Foundation of China under Grant 11975107.

The authors would like to thank Enago (https://www.enago.com) for the English polish.

References

[1] R.R. Wilson, Radiological use of fast protons, Radiology 47 (1946) 487–491.
[2] D. Anicic, M. Gasche, H. Lutz, et al., Upgrading the PROSCAN control system to EPICS: a success story, in: Proceedings of ICALEPCS, 2009. Kobe, Japan, Oct 12–16.
[3] M. Grossmann, Therapy control and patient safety for proton therapy, in: Proceedings of the CAS-CERN Accelerator School on Accelerators for Medical Applications, 2017. Vösendorf, Austria, May 26–June 5.
[4] M. Marchhart, A. Brett, M. Hager, et al., Extending WinCC OA for use as accelerator control system core, in: Proceedings of ICALEPCS, 2013. San Francisco, USA, Oct 6–11.
[5] M. Hager, M. Regodic, A modular software architecture for applications that support accelerator commissioning at MedAustron, in: Proceedings of ICALEPCS, 2015. Melbourne, Australia, Oct 17–23.
[6] P. Hofverberg, J.-M. Bergerot, R. Trimaud, et al., The development of a treatment control system for a passive scattering proton therapy installation, Nucl. Instrum. Methods Phys. Res., Sect. A 1002 (2021) 165264.
[7] G.M. Ma, J. Liu, J.C. Yang, et al., The design of PACS (Physics-Oriented accelerator control system) and its implementation in a heavy ion accelerator facility, Nucl. Instrum. Methods Phys. Res., Sect. A 953 (2020) 163170.
[8] B. Qin, X. Liu, M.W. Fan, et al., Progress of the beamline and energy selection system for HUST proton therapy facility, in: Proceedings of IPAC, 2017. Copenhagen, Denmark, May 14–19.
[9] B. Qin, W. Chen, X. Liu, et al., Design of gantry beamline for HUST proton therapy facility, IEEE Trans. Appl. Supercond. 28 (2018) 4400205.
[10] C. Zhou, D. Li, W.J. Han, et al., Development of a power supply control system and virtual simulation based on docker, in: China International Youth Conference on Electrical Engineering, 2020. Wuhan, China, Nov 2–4.
[11] Qt. https://www.qt.io/.
[12] C++. http://www.cplusplus.com/.