DEVELOPMENT OF PATTERN AWARE UNIT (PAU) FOR THE LCLS BEAM BASED FAST FEEDBACK SYSTEM*

K. H. Kim#, S. Allison, D. Fairley, T. M. Himel, P. Krejcik, D. Rogind, E. Williams, SLAC National Accelerator Laboratory, Menlo Park, CA 94025, U.S.A.

Abstract

LCLS is now successfully operating at its design beam repetition rate of 120 Hz, but in order to ensure stable beam operation at this high rate we have developed a new timing pattern aware EPICS controller for beam line actuators. Actuators that are capable of responding at 120 Hz are controlled by the new Pattern Aware Unit (PAU) as part of the beam-based feedback system [1]. The beam at the LCLS is synchronized to the 60 Hz AC power line phase and is subject to electrical noise which differs according to which of the six possible AC phases is chosen from the 3-phase site power line. Beam operation at 120 Hz interleaves two of these 60 Hz phases and the feedback must be able to apply independent corrections to the beam pulse according to which of the 60 Hz timing patterns the pulse is synchronized to. The PAU works together with the LCLS Event Timing system which broadcasts a timing pattern that uniquely identifies each pulse when it is measured and allows the feedback correction to be applied to subsequent pulses belonging to the same timing pattern, or time slot, as it is referred to at SLAC. At 120 Hz operation this effectively provides us with two independent, but interleaved feedback loops. Other beam programs at the SLAC facility such as LCLS-II and FACET will be pulsed on other time slots and the PAUs in those systems will respond to their appropriate timing patterns. This paper describes the details of the PAU development: real-time requirements and achievement, scalability, and consistency. The operational results will also be described.

INTRODUCTION

SLAC timing system generates 360 Hz fiducial which conducts timing events and is synchronized with the 60 Hz AC power line phase[2, 3]. Thus, there are 6 time slots, each of them corresponding to a different AC power line phase. According to the 120 Hz beam repetition, we need to choose 2 different time slots – TS1 and TS4 for the LCLS, then the LCLS beam runs on the two different AC phases. The different AC phase makes different gain or different response on the actuators – magnets and RF power in accelerator. For the stable 120 Hz beam operation on the LCLS, we need to compensate the different response on the actuators. There are other power line variation sources at the SLAC such as the FACET and LCLS-II which are other beam programs, those are pulsed on other time slots and makes different power line variations. Thus, each of the beam programs makes different power line variations, and needs to compensate for these variations for the stable operation. All of the power line variations can be recognized by the timing pattern on the event system. Thus we have developed a new timing pattern aware EPICS controller which is called Pattern Aware Unit (PAU) for the beam line actuators (Fig. 1).

The PAU should be located in the actuator, and receives control values from the beam-based fast feedback or higher level feedback loop. The feedback system has multiple independent feedback loops which correspond to the each timing patterns. The PAU selects a set value from the multiple feedback loops according to the timing pattern, then implement the set value to the actuator.

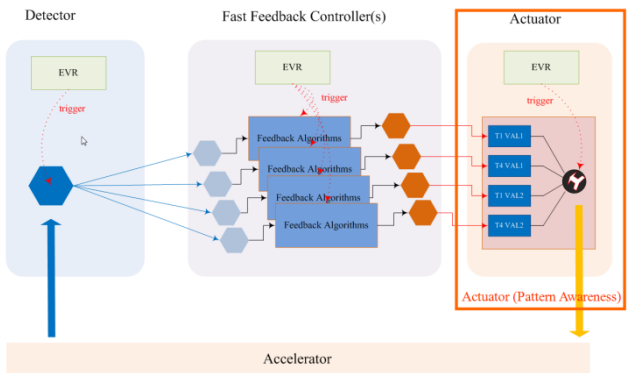

Figure 1: Concept of Pattern Awareness Unit.

REQUIREMENTS

The PAU is a generic software solution for the pattern aware operation. It is required to fit with the SLAC timing system and also required to work with the magnet control system, RF and any other actuator system which requires pattern aware operation.

The PAU is woken up by the fiducial interrupt from the timing system, and performs the pattern matching. The pattern is provided by the event system at every fiducial, and includes the following information: beam code, time slot, and 5 x 32 bits event masks [3].

The PAU performs the pattern matching two times: advanced pattern matching to set the actuator for the next beam pulse, and current pattern matching to get the control data from higher level feedback or to get the beam measurement data from the instruments. For the pattern matching, the PAU utilizes the event pipeline in the event system. This pipeline provides the pattern information two fiducial advanced. That is how the PAU takes action prior to the next beam pulse.

The PAU requires an accurate adjustable time delay due to the various delay times for the data gathering from the high level feedback and the beam measurement

*Work supported by the U.S. DOE Contract DE-AC02-76SF00515
#khkim@SLAC.Stanford.EDU

instrument. Thus, the PAU needs to handle the high-resolution timer on the CPU board. The PAU needs to support various actuators which have very different method to be handled. Thus, the PAU requires an extendability way to handle the different method for different actuators.

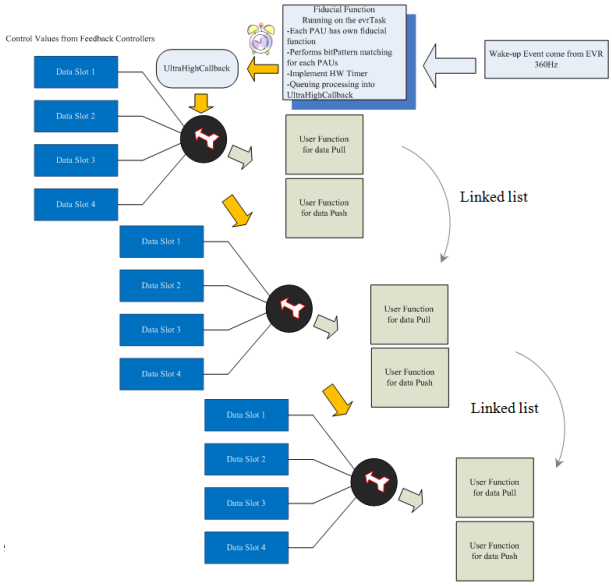

Figure 2: PAU processing for the Multiple Muxes.

IMPLEMENTATIONS

The event system provides a fiducial thread which is woken up by the 360 Hz fiducial interrupts. The PAU utilizes the fiducial thread for the advance/current pattern matching. Actually, the fiducial thread has a hook to register a user function, the PAU registers its hook function as the user function which has the pattern matching codes for the fiducial hook during the initialization. The pattern matching code performs the advance and current pattern matching with the event information from two fiducials ahead, and from the current fiducial, both of which areprovided by the event pipline in the event system.

If the PAU gets the matched pattern, the PAU sets up the high resolution timer on the CPU board which can be adjusted with sub nano-second resolution. The PAU has a user variable which is used for the adjustable time delay. User can configure the PAU delay with this variable. Actually, the delay value is decided by trial experiments to fit into the correct timing for the data gathering.

There is a high priority callback thread, the PAU's own thread called the UltraHighPriority Callback, which is woken up by the high resolution timer. This callback thread has a priority just lower than the fiducial thread and higher than any other EPICS thread. Thus, it can not be blocked by other tasks except the fiducial.

A PAU governs multiple multiplexers, or muxes, each mux corresponds to a physical quantity and a control variable: for example, RF phase, RF amplitude, or strength of corrector magnet. The multiple muxes in the same PAU share the same pattern matching. Thus, the PAU does the pattern matching once instead of the multiple muxes (Fig. 2).

This idea can reduce the pattern matching effort for individual control variable, and improves the real-time performance. This idea also saves the high-resolution timer which is a limited resource: there is finite number of high-resolution timer on the CPU board, in our case it has four. Each PAU occupies one high-resolution timer. Thus, it only allows maximum four PAUs on a CPU board with the high-resolution timer but, the mux idea allows it to handle a larger number of control variables. Actually, the muxes in a PAU has been implemented with the epics linked list (ellList), thus, there is no limit to create multiple instances.

The UltraHighPriority Callback visits each mux in the linked list one by one, and processes the user functions in the muxes. There are two kinds of user function: data pull

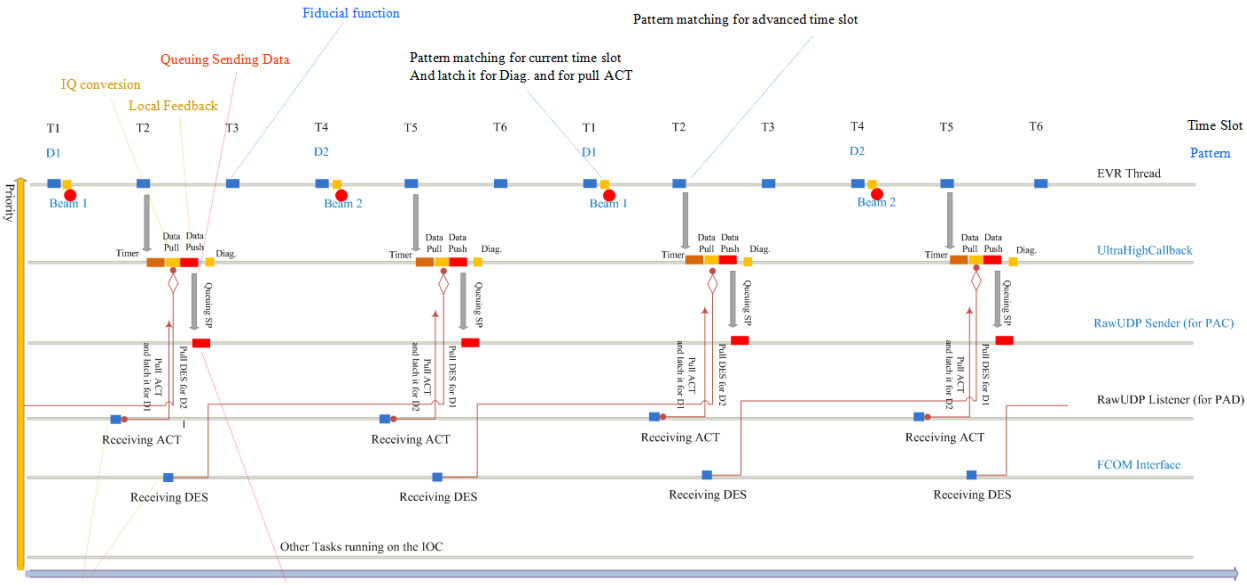

Figure 3: Time line for the 120 Hz Interlaced mode.

function for the current pattern matching, and data push function for the advanced pattern matching. System engineer who is using the PAU software for their system can register those functions for each of muxes.

The data pull function should perform data gathering from the higher level feedback or from the measurement instrument. The incoming data should be moved to the correct data slot which is decided by the current patter matching. The data pull function should perform set data to the actuator, this function brings the data from a correct data slot which is decided by the advanced pattern matching and implements the data into the actuator.

Some systems require more complexity for the data push function. For example, the RF system. This system has local regulation loop for phase and amplitude. The regulation loop requires additional calculations compare to others and also requires both physical quantities: phase and amplitude. Thus, we could not implement both muxes independently for phase and amplitude [4, 5].

The PAU has a concept namely 'Association' to make a relationship between both of the muxes, and we can use an iocsh command to make the association after creating the PAU and the muxes and before the ioc initialization. The concept for the association is not only used for making relation between phase and amplitude, also can be used for the more complex configuration. The local control loop for the LCLS laser RF system, there are three different sources: 2856 MHz oscillator, 119 MHz, and phase cavity. The local feedbacks for each source are related to each other to avoid jumping to a different bucket when switching it to a different source. We also utilize the association for the coupling for these three sources. The PAU has the other callback function namely diag function which has been implemented as a part of PAU software package and is performed very last time of the UltraHighPriority Callback.

After finishing every muxes' user function, the PAU processes a self-diagnostic function. This function does some house keeping work for PAU itself and measures the PAU performance: which data slot has been chosen by the pattern matching, how much time spent by the pattern matching, user functions including the self-diagnostic function itself, and what is the ISR delay for the high-resolution timer. The diag function updates the performance data every time when the PAU is activated by the fiducial, and this information is monitored by EPICS PVs.

Even if the beam based fast feedback supports the pattern aware operation, we still need to consider the non-pattern aware software. There is a longitudinal feedback system which has been written with MATLAB, and controls L2 and L3 section in LCLS, called 6x6 feedback. This 6x6 feedback does not support the pattern aware operation and is still being used for the LCLS. The static offset mode turns this non-pattern aware feedback into a pseudo-pattern aware (Fig. 4). The mux has offset PVs for the each data slot. When the mux implement the set value on the actuator, add up the offset value into the master set value. Thus, the non-pattern aware feedback provides the master set value, and operator or other software can provide pre-defined offset values for each patterns. The pre-defined offset value compensates differences between the different time-slots and patterns. Each mux has a PV to select the fast feedback mode (pattern aware) and static offset mode (pseudo-pattern aware).

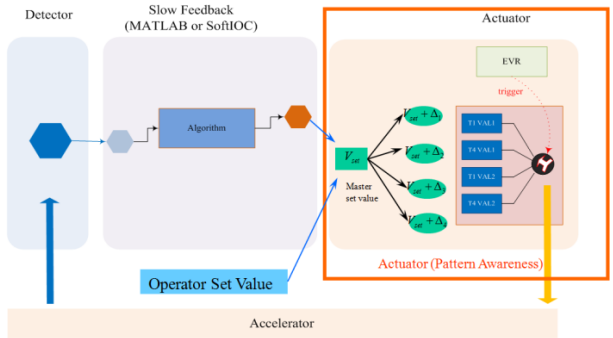

Figure 4: Static offset mode – backward compatibility for the non-pattern aware software.

APPLICATIONS

We have utilized the PAU software package for the RF system and magnets in the LCLS. These systems are controlled by the beam-based fast feedback system. The fast feedback system communicates with the actuator system through FCOM which is a dedicated communication protocol based on UDP/IP-Multicasting. The user pull function cares the FCOM communication with the fast feedback system, and the user push function works for the implement of the actuator set value.

Table 1: PAUs for RF System

PAU	MUXes and Associations	Remarks
PAU0 for Thales Laser System	1 station, 2 associations 3 muxes	2856 MHz PDES 119 MHz PDES Delegate PDES
PAU1for Coherent Laser System	1 station 2 associations 3 muxes	2856 MHz PDES 119 MHz PDES Delegate PDES
PAU2 RF Feedback for IN20	6 stations 6 associations 12 muxes	PDES/ADES for gun, L0A, L0B, TCAV0, L1S, L1X
PAU3 Feedback for LI24	7 stations 9 associations 18 muxes 2 virtual layers (L2/L3 abstraction)	PDES/ADES for L2Ref, TCAV3, KLY24_1, KLY24_2, KLY24_3, S29, S30, L2Abstr, L3Abstr

Especially, the user push function for the RF system is more complicated. It processes the local regulation loop for phase and amplitude, and network communication to the phase&amplitude controller (PAC). RF system has 4

PAUs and each PAU has various number of muxes. We need to use the association concept for the RF system, to make relation among different sources in the laser system, and to make relation between phase and amplitude.

We are using MATLAB based non-pattern aware longitudinal feedback for the L2 and L3 instead of the beam-based fast feedback. The PAU software provides pseudo-pattern aware for this feedback with the static-offset mode. We have implemented an abstraction layer between delegate muxes and individual klystron muxes. The abstraction layer controls the individual muxes to make a collective behaviour on L2 and L3 which reflects the delegate muxes. The non-pattern aware feedback is dealing with the delegate muxes. The abstraction layer has been implemented as the user functions for the muxes.

Table 2: PAUs for Magnet System

PAU	MUXes and Associations	Remarks
PAU0 Corrector in LTU0 area	1 corrector 1 mux	xcor_548
PAU1 Correctors in LTU1 area	3 correctors 3 muxes	xcor_488 xcor_493 xcor_593

PERFORMANCE MEASUREMENTS AND REAL-TIME DEADLINE

The PAU has self-diagnostics which provides accurate measurements for processing time and delay for the every important step. We have utilized this self-diagnostics for the real-time deadline analysis. Table 3 shows the processing time, the worst case is RF system and it takes ~ 105 µsec. According to the Figure 5, PAU wakes up 2 fiducials prior from the next beam. But, PAU need to wait ~ 200 µsec for the communication delay from the beam-based fast feedback or higher level feedback. Some of the actuators require a settling time, worst case is magnet. It requires almost 6 msec. We can consider fiducial interval 2,778 µsec and beam delay from fiducial ~ 1 msec, then the PAU has 251µsec margin from the real-time deadline.

Table 3: Real-Time Performance Measurement

PAU Processing	Processing Time or Delay
Pattern Matching in Fiducial	2 µsec
ISR delay	3 µsec
User Pull Function	25 µsec
User Push Function	60 µsec
Self-diagnostics and house keeping	15 µsec

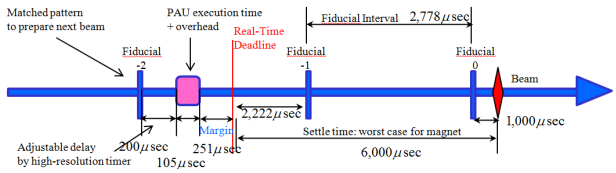

Figure 5: Performance Measurement and Real-Time Deadline.

SUMMARY

The PAU software has been designed as a generic software module for the LCLS beam line actuator. It was implemented to get adaptability to fit into various systems which have particular and unique requirements. We applied PAU software to the RF system, and magnet system, and it contributed for the stable 120 Hz beam-rate operation on LCLS. During this work, we accomplished the followings.

- Integrate with the beam-based fast feedback system for the RF system, and Magnet System
- Achieve the pattern aware operation
- Support non-pattern aware longitudinal feedback for L2 and L3, and make it to pseudo-pattern aware operation
- Implement the various user pull/push functions for the particular/unique requirement: RF local regulation loop, Amplitude/Phase to I and Q conversion, Abstraction Layer for L2 and L3.

REFERENCES

[1] D. Fairley et. al., "Beam Based Feedback For the Linac Coherent Light Source," ICALEPCS 2011 in Grenoble, October 2011.

[2] P. Krejcik and et. al., "LCLS Timing System Requirements," LCLS Physics Requirements Document #1.1-305, 2005.

[3] S. Allison, "Timing and Event System for the LCLS Electron Accelerator," EPICS Collaboration Meeting in Vancouver, May 2009.

[4] A. Akre and et. al., "RF Control Requirements," LCLS Engineering Specific Document #1.1-301, 2005.

[5] D. Kotturi, "LCLS LLRF Distributed Control System," EPICS Collaboration Meeting in Vancouver, May 2009.