THE LCLS-II EXPERIMENT CONTROL SYSTEM PREEMPTIVE MACHINE PROTECTION SYSTEM [*]

A. Wallace[†], SLAC National Accelerator Laboratory, Menlo Park CA, USA

Abstract

The LCLS-II Preemptive Machine Protection System (PMPS) safeguards diagnostics, optics, beam-shaping components and experiment apparatuses from damage due to excess XFEL average power or single shots. The dynamic nature of these systems requires a somewhat novel approach to a machine protection system design, relying more heavily on preemptive interlocks and automation to avoid mismatches between device states and beam parameters. This is in contrast to standard reactive machine protection systems deployed at other laboratories. Safe beam parameter sets are determined from the combination of all integrated devices using a hierarchical arrangement and all state changes are held until beam conditions are assured to be safe. This machine protection system design utilizes the Beckhoff industrial controls platform and EtherCAT. It is woven into the LCLS subsystem controllers as a code library and standardized hardware interface.

INTRODUCTION

Machine Protection Systems (MPS) are ubiquitous in high-energy physics machines around the world. Typically this term is reserved for the subsystem which protects the physical machine hardware from the very phenomenon it produces. Machine protection in the context of the LCLS Experiment Control Systems (ECS) is protection from damage by the x-ray photon beam (XFEL). It is distinguished from other terms such as Personnel Protection Systems (PPS), and Equipment Protection Systems (EPS). EPS typically refers to interlock logic for actuators, or vacuum equipemnt to protect those devices from damaging themselves.

LCLS consists of two control system domains, characterized by different operating requirements. The electron beam production aspect consists of accelerator and undulator technology. Physical access in this domain is more rare and the configuration is generally more stable. Once produced, the XFEL photon x-ray beam proceeds through a front-end where it is measured, attenuated, steered, shaped and focused by x-ray photon devices. After the front-end the beam is delivered to experiment interaction points where it is typically used as a probe. This part of LCLS is more accessible, and requires more in the way of experimental reconfiguration. This dynamic area of the LCLS thus lends itself to a different control system architecture - that of the ECS.

The ECS of LCLS-II is a SCADA type system consisting of Beckhoff PLCs which integrate mechatronics and vacuum system components using a common framework of hardware templates and IEC61131-3 Structured Text libraries [1]. These libraries provide standardized EPS and MPS functionality as well as a uniform EPICS interface. The ECS machine protection functionality is completely implemented within this PLC framework.

REQUIREMENTS

MPS are typically designed to rate-limit or zero-rate beam as fast as possible in reaction to anomalous or fault conditions. Some MPS for pulsed machines are designed to react to an error by turning beam off within the time between pulse bunches (macro-pulses) or even individual pulses [2]. This approach is not technically feasible with the LCLS-II machine parameters.

LCLS-I uses the SLAC Normal Conducting (NC) linear accelerator (linac) with a maximum repetition rate of 120 Hz, leaving ample time between pulses for fault detection and mitigation. LCLS-II, in contrast, has a nominal repetition rate of just less than 1 MHz, while the electronic signal propagation time along the length of the linac along is at least 20 µs.

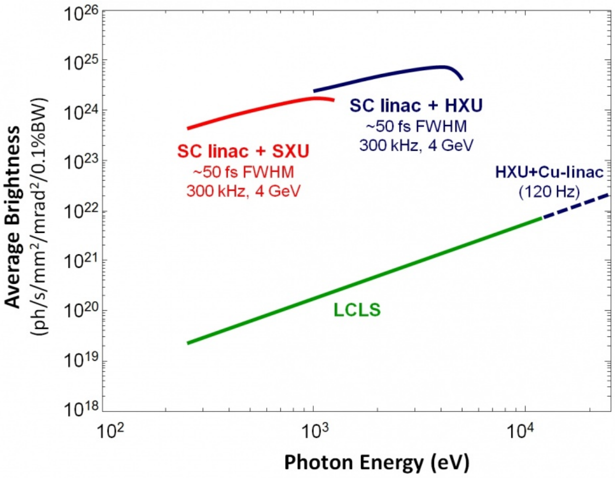


Figure 1: The increase in brightness and average power from LCLS-I to LCLS-II.

Inter-pulse mitigation is not necessarily required for the ECS MPS. Instead, the primary challenge is preventing even a single pulse of the wrong wavelength and energy from entering the experiment system domain, see Fig. 1. Mismatches between optic coatings or diagnostic elements and XFEL pulse wavelength can cause damage. Additionally, with the increased repetition rate, average power delivery can cause harm on a longer timescale. Finally, an additional

[*] Work supported by U.S. D.O.E. Contract DE-AC02-76SF00515
[†] awallace@slac.stanford.edu

challenge of optic thermal stability[1] means that the PMPS must take an active role in optimizing which attenuator to use for downstream components.

Concept and Operation

At a high level, the PMPS[2] tries to avoid ex post facto faults. The concept of preemption is to reduce design requirements and increase safety margins by operating the machine as a diligent and knowledgeable human operator would. That is, by limiting beam intelligently prior to moving a device (and thus changing the configuration of the overall machine), and checking that it is safe (in all respects) to remove restrictions when at a new target position, see Fig. 2.[3]

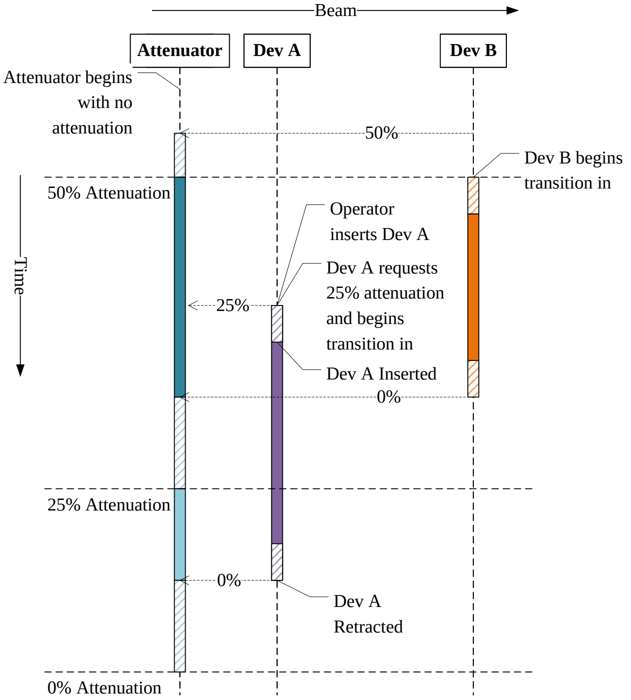


Figure 2: The preemption concept as implemented for two devices with different beam power limits.

With this functionality in place, an MPS would be able to reduce its dependence on reaction time which becomes increasingly expensive and impossible to implement. This translates to lower cost hardware and enables the use of commercial off-the-shelf solutions. While all MPS will require some reactive aspect, with the addition of preemptive checks the reaction time can be relaxed.

ARCHITECTURE

The ECS PMPS architecture starts with a line arbiter PLC (Programmable Logic Controller), which summarizes requests for beam parameter limits and distributes the current beam parameters to subsystem PLCs connected through an EtherCAT fieldbus. Depicted as a loop in Fig. 3, the actual implementation topology is a star, arranged such that branches consist of all subsystems associated with a physical and logical area of LCLS. This way, if there is a disruption of the subsystems of that area, other areas may continue to operate (once the associated faults are either handled or overridden). This functionality is provided by EtherCAT Hot-Connect, and is tolerant to multiple branches being disrupted simultaneously.

Another notable aspect of the PMPS architecture is a requirement for absolute encoding of all motion stages. All devices identified as needing PMPS protection were required to be designed with an absolute position encoding element affixed directly to the load (as opposed to an indirect measurement at the actuator), such as a limit switch, LVDT or a BiSS-C optical encoder. While cost-effective incremental (or relative) encoding solutions could be made pseudo-absolute, the additional implementation complexity, lack of error detection, and impact on machine operation greatly outweighs its cost benefits. Furthermore, the PMPS design and implementation is simplified with absolute encoding.

Reactive System

The reactive system of the ECS PMPS is designed to provide a programmatically easy way to quickly turn off the beam in the event of a fault condition. A fault condition can be generated by any boolean expression (i.e., one which evaluates to true or false) evaluated by PLC logic. Examples include: absolute encoder position evaluation to determine a device state[4], position lag monitoring by motor controls [5], vacuum gate valve position, or even the operational state of a PLC fieldbus or a single IO point.

The ability to compare device state with the current beam parameters is essential for device protection schemes. Beam parameters are made available through an inter-PLC communication system based on EtherCAT. Inclusion of the PMPS library [1] and the addition of a few lines of boilerplate code (see Fig. 4, BP Fanout and Active Beam Param) provides a global and cyclically updated beam parameter structure (see Fig. 3) provided by the line arbiter.

The reactive aspect of the PMPS is implemented by two core function blocks provided in the PMPS PLC library: the Fast Fault Output (FFO), and Fast Fault Block (FFB or equivalently FF). To use this reactive system in a PLC project, an engineer instantiates an FFO, associates its boolean output with a physical digital output, and codes the block to be

[1] Beam power levels in LCLS-II are predicted to cause beam aberrations from thermal induced bumps on optic surfaces. These effects are addressed by engineered solutions but the performance is governed by the thermal response of the optic, so stability in average incident beam power helps, meaning an MPS which minimizes beam-off conditions.

[2] PMPS is overloaded to also mean Photon MPS.

[3] A human operator may also be efficient and recognize when beam is off, everything is safe to move, even before a specified protection device is done reaching a required state for another device to be safe, ie. parallelized operation of mitigation and protected devices. This behavior is designed into the preemptive system.

[4] Another major strength of the PMPS design is the ability to handle complex device states defined by multiple axes or any arbitrary condition. This has been useful in protection of monochromator components from zero-order beam.

[5] Position lag meaning the difference between position setpoint and actual position readback during motion.

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUPDP129

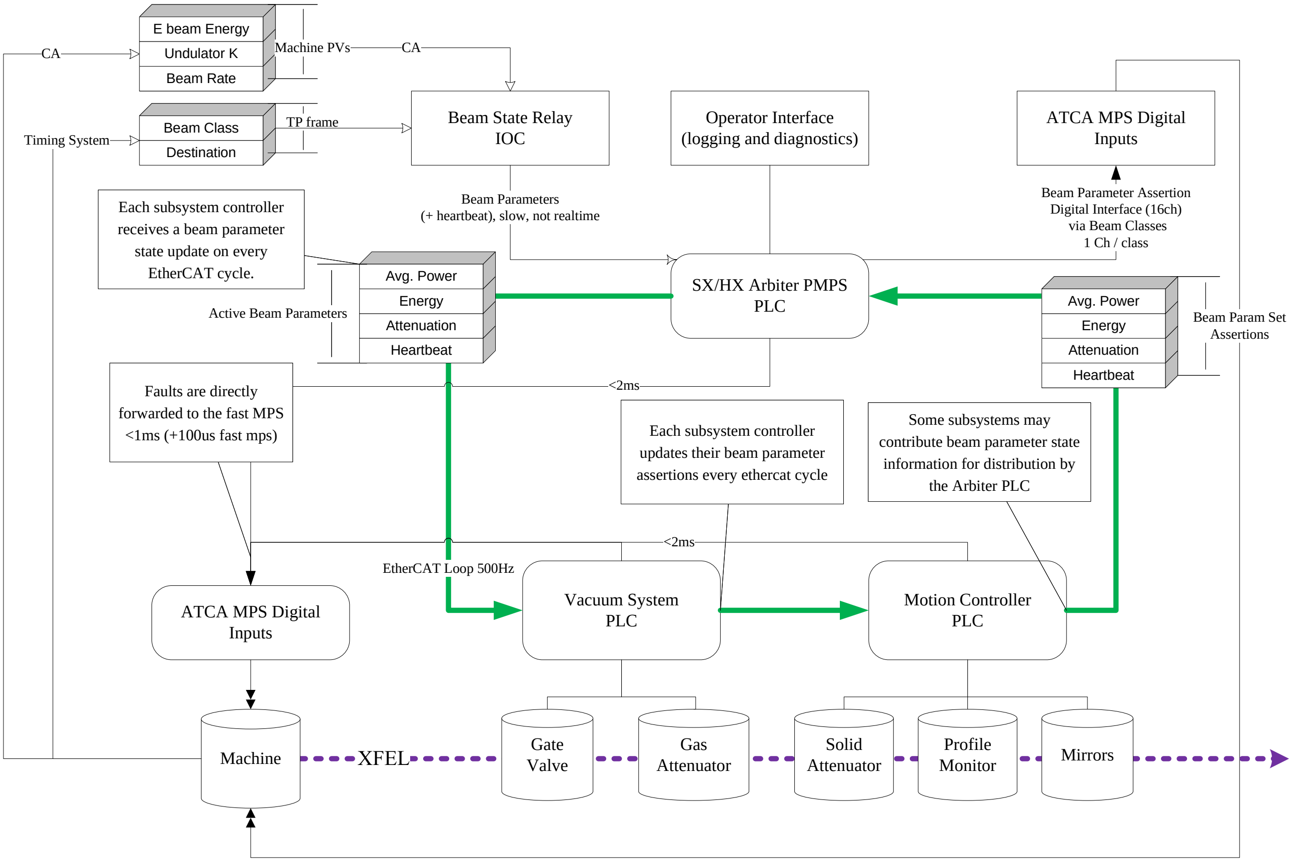

E beam Energy
Undulator K
Beam Rate
CA
Machine PVs
CA
Beam Class
Destination
Timing System
TP frame
Beam State Relay IOC
Operator Interface (logging and diagnostics)
ATCA MPS Digital Inputs
Each subsystem controller receives a beam parameter state update on every EtherCAT cycle.
Beam Parameters (+ heartbeat), slow, not realtime
Beam Parameter Assertion Digital Interface (16ch) via Beam Classes 1 Ch / class
Avg. Power
Energy
Attenuation
Heartbeat
Active Beam Parameters
SX/HX Arbiter PMPS PLC
Avg. Power
Energy
Attenuation
Heartbeat
Beam Param Set Assertions
Faults are directly forwarded to the fast MPS <1ms (+100us fast mps)
<2ms
Each subsystem controller updates their beam parameter assertions every ethercat cycle
Some subsystems may contribute beam parameter state information for distribution by the Arbiter PLC
<2ms
EtherCAT Loop 500Hz
ATCA MPS Digital Inputs
Vacuum System PLC
Motion Controller PLC
Machine
XFEL
Gate Valve
Gas Attenuator
Solid Attenuator
Profile Monitor
Mirrors

Figure 3: This depicts the major interfaces and arrangement of subsystems in the PMPS.

called cyclically. The engineer may also optionally identify a veto condition for the fault output, such as the position of a beam block or an upstream insert-able steering mirror.

FFs are instantiated by providing a description (typically a helpful tag to aid the user in resolving the fault), fault code (for searching for more information), marking as to whether or not it may be overridden by the user (some faults are too risky to override), its latching behavior, and finally the variable name of an FFO. The FFO variable name is used at FF initialization to register the FF with the FFO for diagnostic purposes; the PMPS diagnostic tools connect to these FFO variables. Subsequent invocations of the FF will update the status of a mirror FF structure in an internal-to-PMPS FFO array. Each FF structure has a single boolean indicating if the FFO should signal a fault. This internal FFO array of FF is scanned each time the FFO is called to determine the FFO output state. This scan also updates override status for each FF, applying new overrides requested by operators and calculating expiration of existing ones.

The reactive system reaction time depends on the cycle time of the PLC task, associated digital output and the accelerator MPS response time [3].[6] The reactive function blocks are relatively simple, lending themselves to minimal cycle times. The ECS has a small number of applications of the reactive system (mostly for fast shutters in vacuum systems), with cycle times at or below 500 μs[7]. Most applications of the reactive system are in 1-10 ms tasks, see Table 1.

Table 1: Timescales of ECS Elements

System element	**Reaction time**
BiSS-C Absolute Encoder	325 μs
Fast shutter	1 ms
Best PLC reaction	500 μs
Average PLC reaction	1 ms
Accelerator MPS Response [3]	110 μs

Preemptive System

Fundamentally the preemptive system of the PMPS arbitrates, propagates, and confirms acceptable beam parameter limits for everything in its scope. It does this through a hierarchical arrangement of software objects implementing standard interfaces named ”Higher Authority” and ”Lower Authority”. The preemptive system may span multiple PLCs,

[6] TwinCAT task cycle time in this context is defined as the total reaction time, from input sensing, program execution, through output updates and is dependent upon CPU power and program parameters (complexity and length). Also, cycle time here means both the interval or tick at which a PLC task begins or completes, and is also considered to be the deadline for the task. Ie. the task must be complete before the next interval, so updates to the outputs are occurring at least every task cycle time.

[7] Achieved with ECS standard PLC (CX5010 or CX2020), independent task, EL2022 200 μs T_{off} libraries.

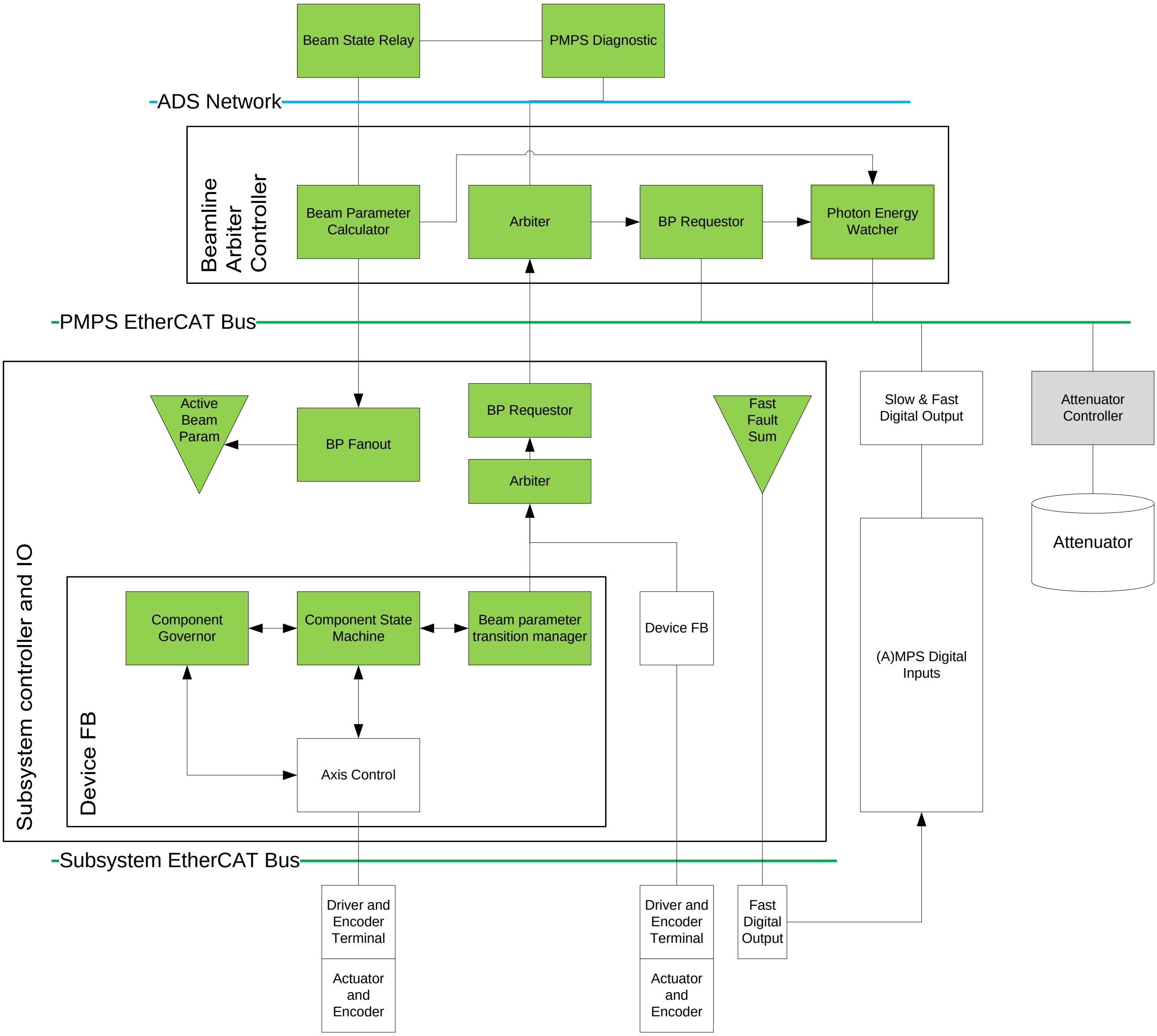


Figure 4: This diagram shows the major components of the PMPS design. Software components are highlighted in green.

operating asynchronously, or be deployed in a single isolated system. Additional functions are provided for inter-PLC communication (see Fig. 4 BP requestor), and the common process of requesting, waiting, and removing beam parameters from arbitration (see Beam parameter transition manager in Fig. 4). The preemptive system is best understood by tracing a beam parameter request through its life cycle, from submission to being honored by the preemptive system and finally to removal.

The preemptive process begins with system startup when all devices that will use the preemptive system request a beam parameter (BP) set with their referenced arbiter (see Fig. 4, Device FB and Arbiter). This initial set is typically a no-beam or beam off request as a matter of precaution. Once the device is fully initialized and it has verified its state, a more permissive BP set may be more appropriate. The device then uses the Beam Parameter Transition Manager (BPTM), to request a new beam parameter set.

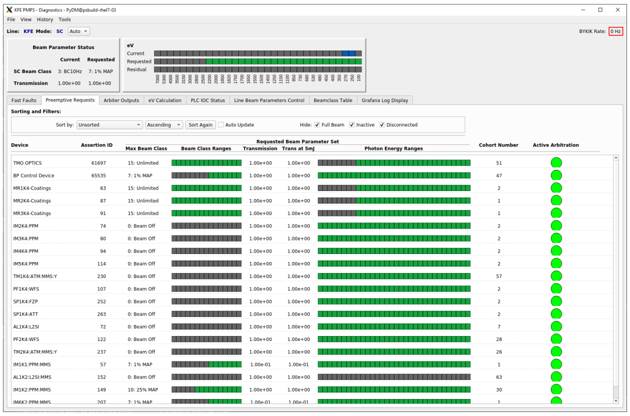
The BPTM uses the arbiter add method to register the new BP set (which may be less or more permissive) with the referenced arbiter. Then the BPTM polls the arbiter cyclically to determine if the new BP set is being honored (see below). Once the arbiter confirms the new request is included in arbitration it removes the earlier (less permissive) request, and signals the device state machine. With the device fully initialized and safe, an operator may make a request to transition to a different state, in which case this process is repeated nearly identically.

The device tells the BPTM the acceptable BP set for the target device state as well as a BP set which is safe for transitioning (usually just beam off, although this is not always the optimal strategy). Since the arbiter and BPTM can handle multiple simultaneous requests from any device, the target state BP set and the transition BP set are requested at the same time. Once confirmed, the Component State Machine (Fig. 4) initiates a device state transition and indicates when it is complete to the BPTM. Then the BPTM removes the transition and original state BP sets from the arbiter. This completes the transition with the only BP set corresponding to the current device state remaining.

Arbiters honor BP requests submitted by associated devices (or BPTM) through a process of escalating an arbitrated BP set to a higher-authority then polling to know when the request is being honored. The higher-authority can be any class that implements the interface. The Arbiter class, as well as the BP Requester class implement the interface, see Fig. 4. In this way Arbiters and BP requesters can stack ad infinitum (within the bounds of reason), and in practice this stack terminates in a BP Requester class which connects to an interface function block. This interface function block also implements the higher-authority interface but instead of polling a higher-authority it immediately marks its request as being honored (in the same cycle).

Diagnostics

PMPS diagnostics consist of an operator graphical user interface (GUI) and Grafana dashboard. The GUI is composed of a parameterized PyDM application and a configuration file which identifies EPICS process variable (PV) prefixes for systems to be included in the display. The operator GUI displays real-time status of all currently active faults and preemptive requests in the PMPS (see Fig. 5). These listings can be filtered based on beam parameters, and whether or not a fault is active, overridden or vetoed.



Beam Parameter Status

	Current	Requested
SC Beam Class	3: BC10Hz	7: 1% MAP
Transmission	1.00e+00	1.00e+00

Sorting and Filters:

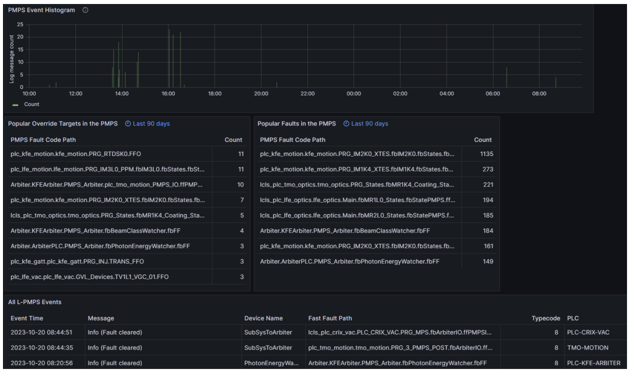
Sort by: Unsorted Ascending □ Auto Update Hide: ☑ Full Beam ☑ Inactive ☑ Disconnected

Device	Assertion ID	Requested Beam Parameter Set					Cohort Number	Active Arbitration
		Max Beam Class	Beam Class Ranges	Transmission	Trans at 5mJ	Photon Energy Ranges		
TMO OPTICS	61697	15: Unlimited		1.00e+00	1.00e+00		51	●
BP Control Device	65535	7: 1% MAP		1.00e+00	1.00e+00		47	●
MR1K4-Coatings	63	15: Unlimited		1.00e+00	1.00e+00		2	●
MR2K4-Coatings	87	15: Unlimited		1.00e+00	1.00e+00		1	●
MR3K4-Coatings	91	15: Unlimited		1.00e+00	1.00e+00		1	●
IM2K4:PPM	74	0: Beam Off		1.00e+00	1.00e+00		2	●
IM3K4:PPM	80	0: Beam Off		1.00e+00	1.00e+00		2	●
IM4K4:PPM	94	0: Beam Off		1.00e+00	1.00e+00		2	●
IM5K4:PPM	114	0: Beam Off		1.00e+00	1.00e+00		2	●
TM1K4:ATM:MMS:Y	230	0: Beam Off		1.00e+00	1.00e+00		57	●
PF1K4:WFS	107	0: Beam Off		1.00e+00	1.00e+00		2	●
SP1K4:FZP	252	0: Beam Off		1.00e+00	1.00e+00		2	●
SP1K4:ATT	263	0: Beam Off		1.00e+00	1.00e+00		2	●
AL1K4:L2SI	72	0: Beam Off		1.00e+00	1.00e+00		7	●
PF2K4:WFS	122	0: Beam Off		1.00e+00	1.00e+00		28	●
TM2K4:ATM:MMS:Y	237	0: Beam Off		1.00e+00	1.00e+00		26	●
IM1K1:PPM:MMS	57	7: 1% MAP		1.00e-01	1.00e-01		1	●
AL1K2:L2SI:MMS	152	0: Beam Off		1.00e+00	1.00e+00		63	●
IM1K2:PPM:MMS	149	10: 25% MAP		1.00e+00	1.00e+00		30	●
IM6K2:PPM:MMS	207	7: 1% MAP		1.00e-01	1.00e-01		1	●

Figure 5: The PMPS operator GUI showing all active preemtpive requests.

The primary Grafana dashboard (see Fig. 6) displays PMPS flagged log messages from the ECS logging system, as well as relevant statistics pertaining to reactive fault and override frequency. PMPS log messages are generated by any subsystem using the PMPS library and detail the events surrounding any reactive fault, override, or configuration change on a single-PLC cycle basis so that nothing is missed.

Another important and notable feature is the usage of a TwinCAT pragma for introspection to generate object instance-paths. The log messages and GUI fault or preemptive request displays contain these object instance-paths to facilitate tracing. The instance-paths show, starting with the network hostname, the complete path to the object instance which is generating a fault or preemptive request in the PMPS. This has been an invaluable feature for ECS logging and diagnostics, and provided further affirmation that Beckhoff's platform was a good choice.



Popular Override Targets in the PMPS Last 90 days

PMPS Fault Code Path	Count
plc_kfe_motion.kfe_motion.PRG_RTDSK0.FFO	11
plc_lfe_motion.lfe_motion.PRG_IM3L0_PPM.fbIM3L0.fbStates.fbSt...	11
Arbiter.KFEArbiter.PMPS_Arbiter.plc_tmo_motion_PMPS_IO.ffPMP...	10
plc_kfe_motion.kfe_motion.PRG_IM2K0_XTES.fbIM2K0.fbStates.fb...	7
lcls_plc_tmo_optics.tmo_optics.PRG_States.fbMR1K4_Coating_Sta...	5
Arbiter.KFEArbiter.PMPS_Arbiter.fbBeamClassWatcher.fbFF	4
Arbiter.ArbiterPLC.PMPS_Arbiter.fbPhotonEnergyWatcher.fbFF	3
plc_kfe_gatt.plc_kfe_gatt.PRG_INJ.TRANS_FFO	3
plc_lfe_vac.plc_lfe_vac.GVL_Devices.TV1L1_VGC_01.FFO	3

Popular Faults in the PMPS Last 90 days

PMPS Fault Code Path	Count
plc_kfe_motion.kfe_motion.PRG_IM2K0_XTES.fbIM2K0.fbStates.fb...	1135
plc_kfe_motion.kfe_motion.PRG_IM1K4_XTES.fbIM1K4.fbStates.fb...	273
lcls_plc_tmo_optics.tmo_optics.PRG_States.fbMR1K4_Coating_Sta...	221
lcls_plc_lfe_optics.lfe_optics.Main.fbMR1L0_States.fbStatePMPS.ff...	194
lcls_plc_lfe_optics.lfe_optics.Main.fbMR2L0_States.fbStatePMPS.f...	185
Arbiter.KFEArbiter.PMPS_Arbiter.fbBeamClassWatcher.fbFF	184
plc_kfe_motion.kfe_motion.PRG_IM2K0_XTES.fbIM2K0.fbStates.fb...	161
Arbiter.ArbiterPLC.PMPS_Arbiter.fbPhotonEnergyWatcher.fbFF	149

All L-PMPS Events

Event Time	Message	Device Name	Fast Fault Path	Typecode	PLC
2023-10-20 08:44:51	Info (Fault cleared)	SubSysToArbiter	lcls_plc_crix_vac.PLC_CRIX_VAC.PRG_MPS.fbArbiterIO.ffPMPSI...	8	PLC-CRIX-VAC
2023-10-20 08:44:35	Info (Fault cleared)	SubSysToArbiter	plc_tmo_motion.tmo_motion.PRG_3_PMPS_POST.fbArbiterIO.ff...	8	TMO-MOTION
2023-10-20 08:20:56	Info (Fault cleared)	PhotonEnergyWa...	Arbiter.KFEArbiter.PMPS_Arbiter.fbPhotonEnergyWatcher.fbFF	8	PLC-KFE-ARBITER

Figure 6: The PMPS Grafana dashboard showing fault and override statistics as well as a live log and histogram of events.

Listing 1: A complete object instance path.

```
plc_kfe_motion.kfe_motion.PRG_IM1K3_PPM
    .fbIM1K3.fbStates.fbPMPSCore.
    fbMiscStatesErrorFFO.ffBeamParamsOk
```

CONCLUSION

The PMPS has been deployed at the LCLS since 2020 and operated successfully with the normal-conducting linac, balancing beam uptime with zero incidents of unauthorized beam. Commissioning and operation with the superconducting linac required additional features implemented in a recently-deployed major revision of the PMPS. This update better accommodates a wider range of machine and XFEL configurations including attosecond pulses and other modes. A new PMPS configuration management tool has recently been deployed enabling authorized operators to directly manage beam parameter thresholds in a traceable fashion and with proper engineering controls, without the intervention of a control system engineer.

There are a number of ideas for improvements and new developments in the context of the PMPS in retrospect. While the EtherCAT communication system tightly couples the entire PMPS system, it may be replaceable with Beckhoff's Realtime TCP/UDP package and this may lead to some simplification in the architecture. This option was not explored due to lack of awareness of its existence. Another idea is the implementation of hierarchical arbiters directly in EPICS. When the PMPS was being designed, PLC-layer implementation was prioritized for reliability and its real-time capabilities. However the preemptive aspect does not need a real-time platform to function. Keeping the reactive and preemptive system together in the PLC layer makes logical sense for PMPS applications, but the arbiter component could be implemented in the form of an EPICS record which could then be used to give EPICS only devices some preemptive protection. This would increase options for implementing a graduated approach to risk mitigation in EPICS, as well as adding another tool to the EPICS-layer automation kit.

The PMPS design philosophy of distributed and tightly integrated machine protection functionality, combined with a dynamic system has served LCLS operations well. The PMPS provides a solid framework for implementing engineered controls to safeguard XFEL devices in highly varied and complex arrangements. This was only possible by addressing the increased complexity in requirements for protection with modern solutions for real-time controls.

ACKNOWLEDGEMENTS

The PMPS was developed as a result of years of effort combined with indispensable input from ECS team colleagues as well as our stakeholders. Margaret Ghaly carefully considered and reviewed the EtherCAT interface system, override philosophy, and took over PMPS development after v0.13.1, releasing v1.0.0 and subsequent versions. Zachary Lentz authored the LCLS Motion library which became the primary application of the PMPS library, consulted on the reactive system kernel and the design of the preemptive system. Without his steely logic I would have been lost many times. Ken Lauer and Michael Browne helped review the overall architecture and pushed for a more rigid confirmation mechanism in the preemptive system. Hugo Slepicka delivered the first iterations of the complex diagnostic GUI application.

REFERENCES

[1] A. Wallace *et al.*, “Photon Machine Protection System *PMPS* TwinCAT library for LCLS-II”, `https://github.com/pcdshub/lcls-twincat-pmps/releases/tag/v2.2.1`

[2] S. R. Norum *et al.*, “The Machine Protection System for the Linac Coherent Light Source”, in *Proc. PAC’09*, Vancouver, BC, Canada, May 2009, paper FR5REP039, pp. 4856–4858.

[3] J. Mock, “Commissioning of the LCLS-II Machine Protection System for MHz CW Beams”, presented at the IBIC’23, Saskatoon, SK, Canada, Sep. 2023, paper TU3I01.