ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA129

AUTOMATED CONTOLS FOR THE HARD X-RAY SPLIT & DELAY SYSTEM AT THE LINAC COHERENT LIGHT SOURCE

A.P. Rashed Ahmed, M.C. Browne, D.L. Flath, K.L. Gumerlock, T.K. Johnson, L. Lee, Z. Lentz, T.H. Rendahl, H. Shi, H.H. Slepicka, Y. Sun, T.A. Wallace, D. Zhu

Abstract

The hard x-ray split and delay (HXRSnD) system at the Linear Coherent Light Source (LCLS) was designed to allow for experiments requiring two-pulse based x-ray photon correlation spectroscopy. The system consists of eight silicon crystals split between two optical branches, with over 30 degrees of freedom. To maintain system stability and safety while easing system operation, we expand the LCLS Skywalker software suite to provide a python-based automation scheme that handles alignment, operations and engineer notification. Core safety systems such as collision avoidance are processed at the controller and Experimental Physics and Industrial Control System (EPICS) layer. Higher level functionality is implemented using a stack of open-source python packages (ophyd, bluesky, transitions) which provide a comprehensive and robust operational environment consisting of virtual motors, plans and finite state machines (FSM).

INTRODUCTION

Between the various operational modes of the hard x-ray free electron laser (FEL) at the Linac Coherent Light Source (LCLS), delays in the 1 ps to 1 ns regime have been unattainable using multi-bunch techniques[1]. To help fill this gap, x-ray optics which split the beam while adding a predefined delay must be used. The hard x-ray split and delay (HXRSnD) system at LCLS fulfills this role by splitting the beam using Si(220) crystals, passing both halves through variable and static delay branches, then recombining them at the end of the enclosure.

The new HXRSnD system consists of almost 30 axes of motion and eight diagnostics for alignment and poses a significant challenge for basic operations since the most desired parameters such as energy and delay require coordinated motion between multiple motors in the system. Additionally, exposing the system in its entirety rather than in discrete states makes it prone to failure by permitting access to previously untested system states.

The HXRSnD system requires a fully automated controls system including the alignment, operations, and engineer notification. Introducing automation is especially prudent when considering projects such as the systems for LCLS-II, the future laboratory upgrade which will result in three new experimental soft x-ray hutches. To adequately prepare for LCLS-II, investing in automation opportunities such as the HXRSnD system proves invaluable as a test-bed for new controls frameworks that allow for full automation.

SYSTEM DESIGN AND HARDWARE

The HXRSnD system is divided into the towers, diagnostics, and pneumatics. There are four towers in the system labeled 1 through 4, with 1 and 4 being the farthest upstream and downstream towers respectively. Diagnostics are present before and after the enclosure, and between each of the silicon (Si) crystals. A system of pneumatics handles the flow and temperature of nitrogen (N_2) and helium (He) into the system.

After the beam has been split at the start of the enclosure, each half will travel through one of two paths: the delay, or channel cut branch. The delay branch is comprised of towers 1 and 4, along with the diagnostics between the tower crystals, and is capable of producing a delay range of -30 ps to 500 ps at 8 keV. The channel cut branch is comprised of towers 2 and 3, along with the diagnostics between them and remains at a fixed delay for a given energy.

Tower System

The towers require a high level of performance and are composed of a combination of servo motors and piezo stages (see Fig 1). Each delay arm is built on top of an Aerotech ANT180 linear stage for insertion/removal of the arm from the beam, with an Aerotech APR150DR-135 positioning the arm on the granite table. The Si crystals rest on top of two Aerotech ANT95-180-R stages which adjust the crystal angles with respect to each other. One of the ANT95-180-R stages is placed on a custom stage built using an Aerotech BMS35 motor, while the other is static, allowing for the time delay between the two branches to be adjusted. Each of the crystals is mounted on top of an Attocube ECGt5050 goniometer and Attocube ECSz5050 vertical translation stage. An Attocube ECSx5050 linear translation stage is used for insertion of a Hamamatsu S3590-19 PIN diode for measuring the beam intensity at the delay crystal. The channel cut branch towers are built using Aerotech ANT95-100-L linear stages for insertion and removal of the crystals from the beam, and ANT95-180-R rotation stages for angular adjustment.

Diagnostics

In addition to the servo motors and piezo stages making up the delay and channel cut branches, beam diagnostics can be inserted and removed along each branch, as well as at the input and output of the system. Hamamatsu S3590-19 PIN diodes are used for beam intensity measurements, while Mako G192B PoE CCD detectors are used for beam profile

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA129

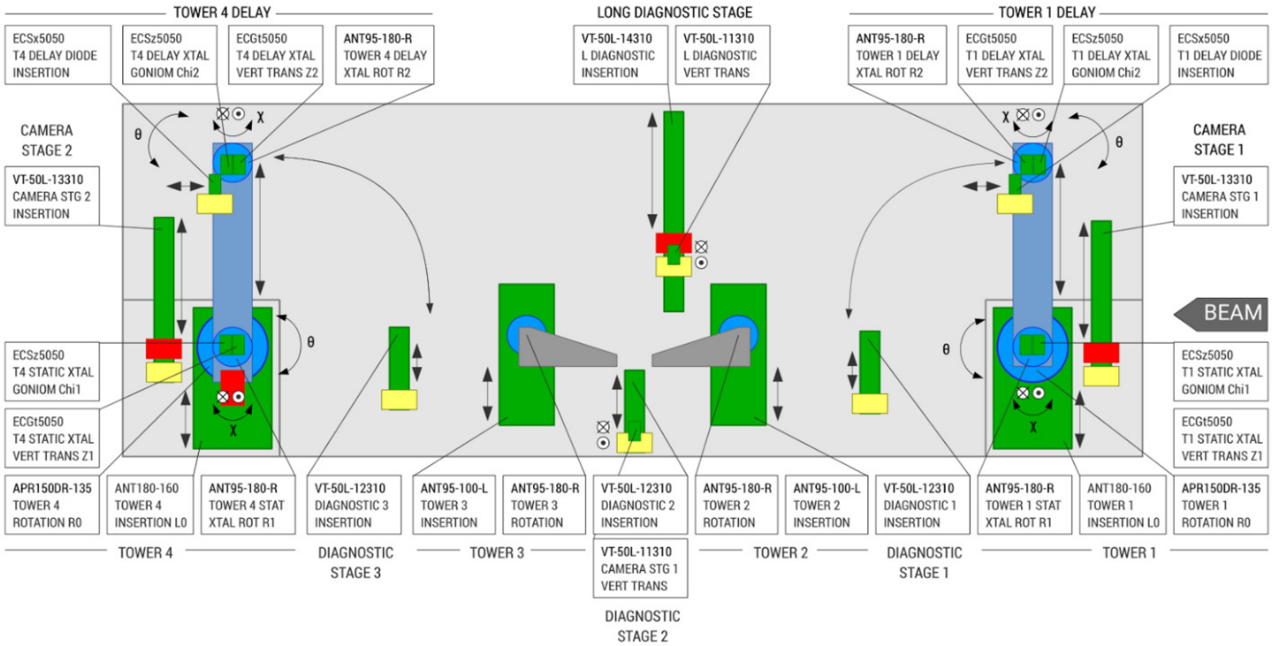

Figure 1: Detailed overview of the HXRSnD motion system for the towers and diagnostics.

measurements. These diagnostic stages do not require the same high level of performance the delay towers do, and as such are positioned using stepper motors. These diagnostics are mounted on top of Micronix VT-50L stages of varying lengths, allowing for horizontal and vertical positioning.

Pneumatics

N_2 is required to move the large theta motor and linear delay stage on the delay towers. More specifically, the N_2 is used to float the towers 5 μm above the table and open a U-channel aluminum casing on the delay stage, permitting linear motion. The granite table has a vacuum system that produces a suction force on the towers, holding them in position. SMC digital solenoid valves are present on the N_2 lines and the vacuum lines for flow control as well as SMC ISE30A and ZSE30A digital pressure switches for feedback. Additionally, there are gas heaters for the N_2 and He along with resistance temperature detectors (RTDs) to temperature control the system.

SOFTWARE ARCHITECTURE

The software stack designed for the HXRSnD is comprised of the Aerotech controllers, Experimental Physics and Industrial Control System (EPICS), ophyd[2], bluesky[3], and transitions[4]. The stack hierarchy has the Aerotech controller at the lowest level, EPICS, providing the transport layer, then ophyd, bluesky, and transitions providing the python device, procedure, and state interface. User-level operation is done using an IPython shell, a Jupyter notebook, or PyDM[5], a Python-based graphical user interface (GUI) developed at LCLS.

Aerotech Ensemble

Each Aerotech controller is loaded with several programs, called Tasks, which can be used to automate motion routines, collect and output data, or monitor the various axes on the controller. Due to the small working space of the HXRSnD system and the large number of axes involved, collisions are possible. Additionally, the air bearings the delay arms glide on can run off of the granite table at certain stage position combinations. To mitigate these issues Aerobasic tasks were added to each Aerotech controller to monitor the positions of the relevant axes and keep them within defined safe zones.

EPICS

For our hardware communication layer, we used EPICS, a distributed control system providing supervisory control and data acquisition (SCADA) capabilities, and implements client communication using the channel access (CA) network protocol. Safety systems were implemented at the IOC level using soft and operating limits which were set to be just within limits set at the Aerotech controller. Additionally, interlocking between the motors and pneumatics were also added to prevent motion while the system is in the parked state. IOCs were also used to host process variables (PVs) outputted by the higher-level python interface.

Ophyd

A python package called Ophyd was used to provide a python interface to the hardware exposed in EPICS. Ophyd provides a framework to implement hardware as control-system agnostic classes with a uniform high-level API. Additionally, their hierarchical nature allows lower level components to be seamlessly combined into aggregate devices.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA129

Thus, all the lower-level devices such as the motors, diagnostics, and pneumatics were implemented using Ophyd, along with the towers, virtual motors, and the HXRSnD system as a whole.

Bluesky

State-transitions and scans required by the system were all written in python using a package called Bluesky. The package provides a library for control and data collection using AsyncIO - the python module which implements coroutines for writing single-threaded concurrent code. Under this framework, sequences of instructions (for example, a scan) are implemented as plans that are executed by a Bluesky run-engine.

Transitions

At the highest level, the system is controlled using finite state machines (FSMs) written using the transitions package in python. Transitions provides the necessary utilities needed to implement state machines with defined states and allowed or conditional transitions. The FSM was used to minimize the number of different system configurations available to the operator, simplifying usage while elucidating the debugging process.

PRIMARY FSM

User level control of the HXRSnD system is done using the primary FSM which handles state transitions between all the system states. In the FSM, there are four main states: "normal operation," "online calibration," "parked," and "error latch" (see fig. 2). Each of these states has secondary states that are only accessed by being in its corresponding primary state. Transitions between states are implemented as bluesky plans that carry out all the necessary changes to the system required by the destination state, transitioning to the "error latch" state in the event of a transition error.

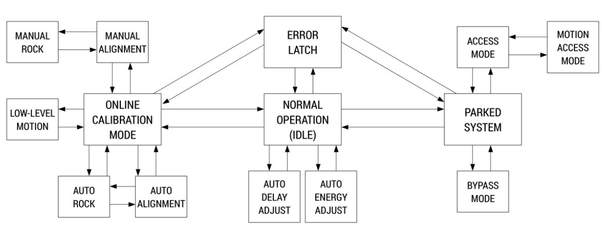

Figure 2: Primary FSM overview with secondary states and allowed transitions.

Normal Operation

The expected operational model is that under most circumstances, the only necessary function is the ability to change the overall energies and delay. For the HXRSnD system, the "normal operation" state is characterized by towers 1 and 4 being in the "flying" state, restricted low-level motion, full access to the high-level virtual motors (E1, E2 and Delay), enclosure state being "sealed", x-ray permission being in the "permitted" state, referenced, and the system be either manually or automatically aligned. From "normal operation," access to the two other states is permitted and all plans for transitioning into "normal operation" must ensure the conditions above are met.

Parked System

When the HXRSnD system needs to be placed in a configuration meant to stay static, the "parked system" state is used to guarantee immobility. In this state, the air flowing through the air bearing is shut off, and the granite vacuum is turned on, anchoring the bearings of the tower, and linear delay stage and all motors are disabled. Other characteristics include the system alignment, referenced, in a sealed state, and in a permitted x-ray state.

Secondary states are present to physically or programmatically give access to the system as necessary. Transitioning into the "access mode" state prohibits -rays and all motors remain immobile. If access and motor motion are required, then transitioning to the "motion access" state grants permission to do so.

For experiments where the HXRSnD is not needed, all components must be placed into the retracted position to allow the beam to bypass the system. The "bypass mode" state indicates that the system is fully removed from the beam path, and drift-tube has been installed in its place. Transitioning into or out of this state initiates the retraction or insertion procedure.

Online Calibration

If the system needs to be recalibrated or loses its alignment or reference, the "online calibration" state gives access to the manual and auto-alignment, referencing plans or lower level motors for operator use. The only restriction for this mode is that towers 1 and 4 must be in the "flying" state.

Secondary modes for this state are "manual alignment," "auto-alignment," and "low-level motion." In the "manual alignment" state, only the alignment motors such as the crystal rotations are permitted for user motion. From this state, the user will have access to the rocking-curve procedure taking the individual crystals and diagnostics as inputs. The user then specifies the alignment has completed and the system alignment will change to "manually aligned."

Transitioning to the "auto-alignment" begins execution of the alignment procedure, returning to the "online calibration" state upon completion. For control of individual motors, "low-level motion" provides this functionality, in addition to allowing the air bearings to transition into the "locked" and "landed" state.

Error Latch

In the event an error is encountered, the "error latch" state ensures system safety by preventing any further interaction without acknowledgment and notifies the relevant engineers using the LCLS Easiest Alarm System Ever[6] (EASE) EPICS monitoring tool. EASE is a utility for monitoring EPICS PVs and sends notifications via email to the

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA129

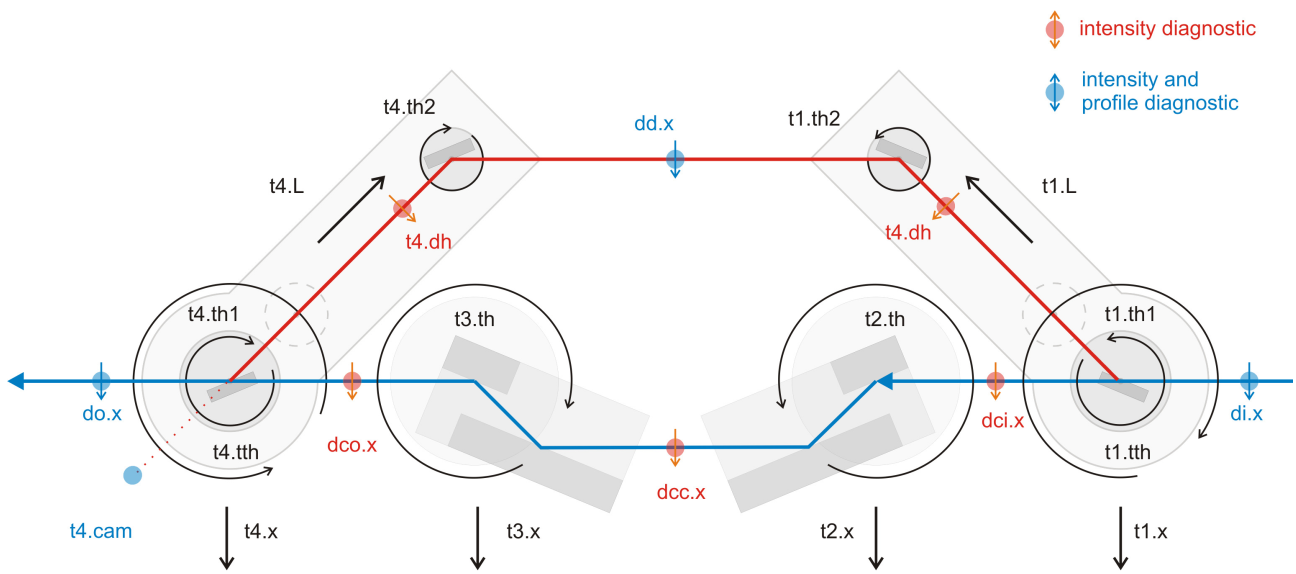

Figure 3: Outline of alignment components. Delay line in red, channel cut line in blue.

specified personnel depending on the alarm values configured for a particular PV.

Auxiliary FSMs

Within the HXRSnD system, there are some auxiliary FSMs the primary FSM depends on. Notable auxiliary FSMs are the referenced and air bearing FSMs, which determine if the system needs to be referenced, and how to transition the towers between pneumatic states. For all of the axes of motion throughout the HXRSnD system, to reliably perform any motion all the involved motors must be referenced or homed.

Each of the delay towers has a separate internal state machine that ensures the proper air-flow procedure is followed when attempting to transition between locked and flying states. The FSM consists of four states with two alternating parameters: the N_2 air supply and table vacuum.

VIRTUAL MOTORS

Within the "normal operations" state, the primary requirement is tuning the system energies or delay. Each branch was combined to form virtual motors representing the delay branch energy, channel cut branch energy, and system delay to remove low-level interaction. Tuning any of these system parameters involves at least three motors, with the delay branch energy requiring eight but with Ophyd, these motors are abstracted as a single device with motor-like functionality such as relative motion, scanning, and waiting.

System Delay

Setting the system delay (Delay) first requires a conversion from the inputted delay in time to a linear stage position, L, where Ł is defined in Eq 1 as:

$$L = \frac{1}{(1 - cos(2\theta_1))}\left(\frac{ct}{2} + \frac{d_G(1 - cos(2\theta_2))}{sin(\theta_2)}\right) \quad (1)$$

Where t is the desired delay, c is the speed of light in vacuum, d_G is the gap between channel cut crystals, θ_1 is the delay branch Bragg angle, and θ_2 is the channel cut branch Bragg angle. The delay stages t1.L and t4.L are then moved to position Ł . To maintain beam position on the dd diagnostic, dd.x must be moved to dd_L, which is defined in Eq 2:

$$dd_L = Lsin(2\theta_1) \quad (2)$$

Delay Branch Energy

Tuning the energy of the delay branch (E1) requires motion of all the crystals and middle diagnostic on the line (see fig 3). After calculating the Bragg angle, θ_1 using the desired delay branch energy, the static crystals t1.th1, t1.th2, t4.th1, and t4.th2 are moved to theta1, while t1.tth and t4.tth are moved to $2\theta_1$. The dd diagnostic is then moved to the position calculated using Eq 2.

Channel Cut Branch Energy

The channel cut branch energy (E2) requires motion on both branch crystals as well as the dcc diagnostic. After calculating the Bragg angle, θ_2 for the desired energy, crystals t2.th and t3.th are moved to θ_2. The diagnostic dcc.x is then moved to dcc_L, which is defined in Eq 3:

$$dcc_L = 2d_G cos(\theta_2) \quad (3)$$

AUTOMATIC ALIGNMENT

Performing an automated alignment of the system requires the coordination of all the motors in the delay and channel cut branches in addition to all the diagnostic motors. Figure

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA129

3 outlines the position and orientation of the motors relevant to system alignment.

Rocking-Curve Intensity Maximization

The rocking curve intensity maximization is the procedure used to fine tune the angles of the crystals to find the angle that best satisfies the Bragg condition. The scheme involves a reflecting surface (Si crystals), an upstream and a downstream diagnostic. The basic procedure is as follows:

- Set up a linear scan around a target position.
- Perform a linear scan using the reflecting surface, recording the beam intensity at the downstream and upstream diagnostic.
- Fit a Lorentzian curve to the ratio of downstream beam intensity to upstream beam intensity.
- Move the reflecting surface to the position that maximizes the fit.
- Repeat as necessary using a finer scan.

Alignment Procedure

Aligning the system for operation can be broken into two independent sections: alignment of the delay line, and alignment of the channel cut line. Below is the procedure for aligning both in series.

- Move t1.x, t4.x, dd.x, t2.x, t3.x, dci.x, dcc.x, and dco.x to their inserted positions.
- Align the beam to the HXRSnD enclosure at di using the Skywalker[7] beam alignment suite.
- Adjust t1.y1 by finding the position where the static crystal cuts the beam intensity in half between di and dci
- Move t1.y2, t4.y1, t4.y2, and dd.y to the position found for t1.y1.
- As a rough alignment, move virtual-motors E1 and Delay to the desired delay line energy and system delay. Verify that the beam is traveling through the system.
- Perform a rocking-curve procedure using t1.th1 and t1.chi1 measuring the beam intensity at t1.dh and di.
- Perform a rocking-curve procedure using t1.th2 and t1.chi2 measuring the beam intensity at dd and t1.dh.
- Perform a rocking-curve procedure using t4.th2 and t4.chi2 measuring the beam intensity at t4.dh and dd.
- Ensure the beam is correctly aligned to t4.th1 using t4.cam, adjusting the t4.y1 accordingly.
- Perform a rocking-curve procedure using t4.th1 and t1.chi1 measuring the beam intensity at do and t4.dh.
- As a rough alignment, move virtual-motor E2 to the desired channel cut energy. Verify that the beam is traveling through the system.
- Perform a rocking-curve procedure using t2.th measuring the beam intensity at dcc and dci.
- Perform a rocking-curve procedure using t3.th measuring the beam intensity at dc and dcc.

CURRENT PROGRESS

The HXRSnD system is currently in the process of fully implementing the automation architecture put forth in this paper. The system has the motor safety systems completed in both the Aerotech controller and the EPICS layer, in addition to the ophyd high and low-level devices including the virtual motors. Additionally, the rocking curve procedure has been fully implemented in bluesky, leaving the finite state machines, transitions, and alignment and homing procedures to be completed. In the upcoming fall beam-times we expect the system to reach complete implementation, resulting in full automation.

CONCLUSIONS

Extending the Skywalker framework to include FSMs allowed for more comprehensive system control and represents a significant step on the path to automation at LCLS. Using a software stack consisting of controller logic for safety, ophyd for low and high-level device abstractions, bluesky for procedures and state transitions and transitions to create the state topology, we demonstrated an automation scheme for the HXRSnD. And while the system has not been fully implemented, the upcoming beam-times will allow for the completion of the full software architecture.

ACKNOWLEDGMENT

Use of the Linac Coherent Light Source (LCLS), SLAC National Accelerator Laboratory, is supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences under Contract No. DE-AC02-76SF00515.

REFERENCES

[1] Zhu, D. *et al.*, "Development of a hard x-ray split-delay system at the Linac Coherent Light Source", Proc. SPIE 10237, Advances in X-ray Free-Electron Lasers Instrumentation IV, 102370R (June 2017)

[2] National Synchrotron Light Source II (December 2014), Ophyd v0.7.0 [Python Package]. Retrieved from http://nsls-ii.github.io/ophyd/

[3] National Synchrotron Light Source II (June 2015), Bluesky Data Collection Framework v0.10.0 [Python Package]. Retrieved from http://nsls-ii.github.io/bluesky

[4] T. Yarkoni, A. Neumann, et al., Transitions v0.6.1 [Python Package]. Retrieved from https://github.com/pytransitions/transitions (October 2014)

[5] M.L. Gibbs *et al.*, “PyDM: A Python-Based Framework for Control System Graphical User Interfaces”, ICALEPCS 2017, Barcelona (October 2017)

[6] N. W. Brown, *et al.*, “Developing EASE, A remote EPICS monitoring tool”, SLAC National Laboratories (2016).

[7] T.H. Rendahl *et al.*,“Skywalker: Python Suite for Automated Photon Alignment at LCLS”, ICALEPCS 2017, Barcelona, Spain, October 2017.