UI/UX Development for Common Laser Experiments

Standardized Component Screens for Tabletop Integrated Laser Elements (TILEs)

Moyinoluwa Adelowo[1,2] / Intern / Experimental Control Systems
Adam Berges[1] / Mentor / Experimental Control Systems

1. SLAC National Accelerator Laboratory, USA. 2. Alabama A&M University, USA

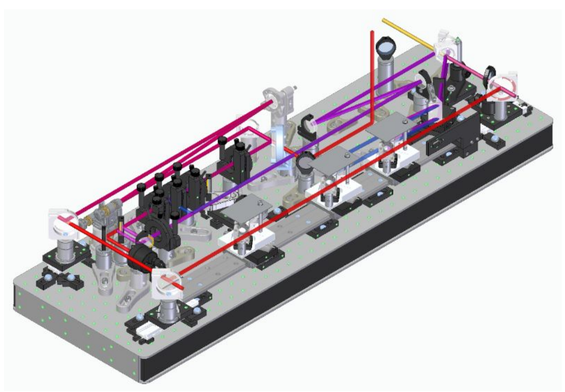
CONNECT WITH ME

BACKGROUND

LCLS hutches pair the X-ray free-electron laser with tabletop ultrafast lasers to make the pump / probe photons each experiment needs.
These Tabletop Integrated Laser Elements (TILEs) pack in many control points required to manipulate optomechanics and detectors and each of these control points needs to be exposed as an EPICS process variable (PV).
Scientists steer those PVs through GUIs built on the lab's PyDM + PyQt stack.

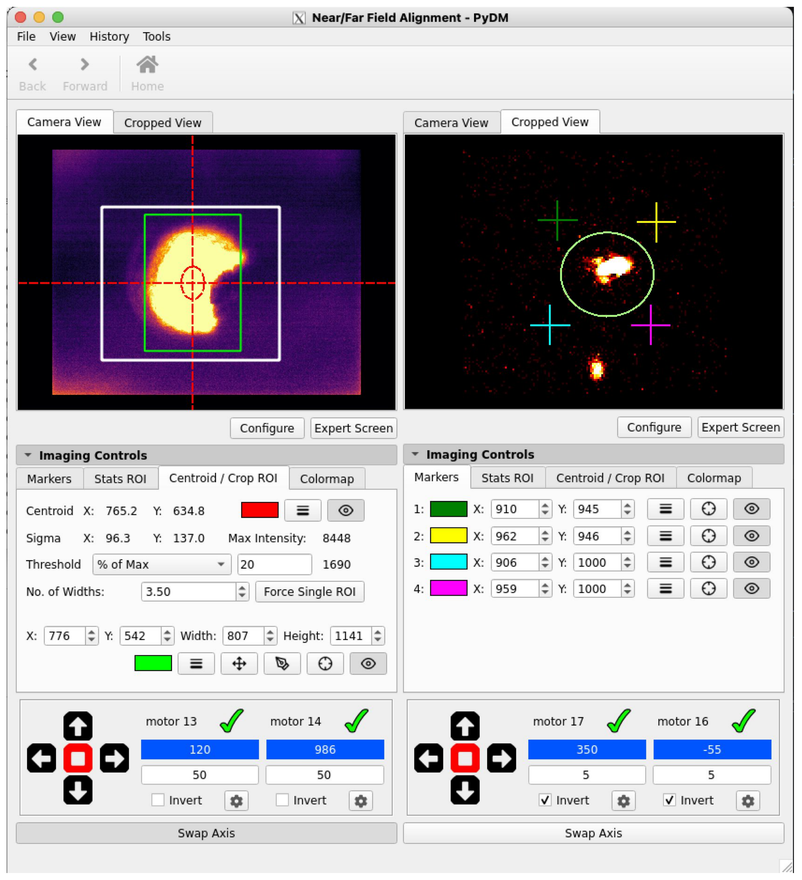
THE PROBLEM

Screens tend to either be one-offs, hard to reuse, tied to one hutch, forgetting their settings on restart, and slow to keep up with what scientists actually need.



Tabletop Integrated Laser Element

FEATURED BUILD: NEAR / FAR-FIELD BEAM-ALIGNMENT SCREEN



Near/Far Field Alignment Screen

DESIGN GOALS

Streamlined PyDM screens that are:

- Deployment-agnostic - the same screen runs in XPP, TMO, any hutch
- User-configurable - pick your own cameras, motors, and PVs
- Session-recoverable - most setting persists across restarts
- Iterable - respond quickly to developing needs from users

WHAT IT DOES

A single app with two screens used to align a mirror in the near-field and far-field at the same time. A scientist watches the live beam, drops reference markers, tracks the beam centroid, tunes the region-of-interest (ROI) and colormap, then steers the mirror by tip/tilt until both fields line up, and the screen store applicable settings between launches.

ANATOMY (per column)

- Live full camera and cropped camera image feed (tabbed)
- Configure button (auto-wires areaDetector plugins)
- Launch camera IOC's expert screen button
- Collapsible tab: Marker Selection, Stats ROI, Centroid Tracker, Crop ROI, Colormap
- Motor tip/tilt controls
- Swap horizontal/vertical motors button

BUILT BY ITERATIONS - DRIVEN BY USER FEEDBACK

The screen didn't arrive fully formed. Every capability below came from a round of feedback from laser scientists and code reviewers, each shipped as a small, testable increment with a safe fallback.

1 **"Changes I make should persist when I re-launch the GUI"**

Built a JSON persistence layer that saves settings to the user's home directory, keyed to the hardware (camera + motor PVs) so they follow the device rather than the screen. ~15 settings now restore automatically on relaunch.

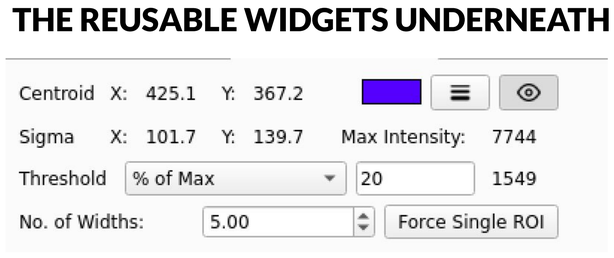
2 **"Markers don't show up on the cropped view"**

Re-architected the shared overlay primitive so one marker can render in two camera views at once, each with its own live offset. Markers and the centroid now mirror onto the cropped view, and can be placed directly from it too.

3 **"On Centroid, it's critical to be able to add a threshold"**

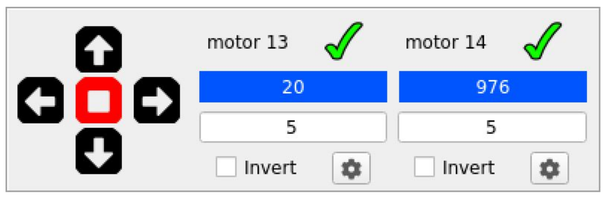
Added a threshold control with three modes (% of max, $1/e^2$ of max, and raw pixel value) with live readbacks. The %-of-max and $1/e^2$ modes track the beam's live max intensity as it changes; raw commits on demand.

THE REUSABLE WIDGETS UNDERNEATH



CENTROID TRACKER WIDGET

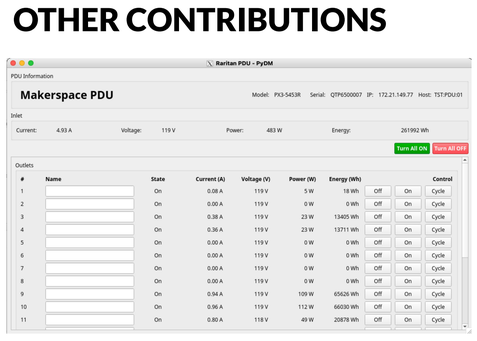
A widget that overlays the beam centroid on a camera image as an ellipse shaped marker (sized by sigma), writes the centroid-derived ROI to EPICS (camera ROI plugin), and has EPICS-based centroid threshold control ($1/e^2$, %, raw).



TIP / TILT MOTOR WIDGET

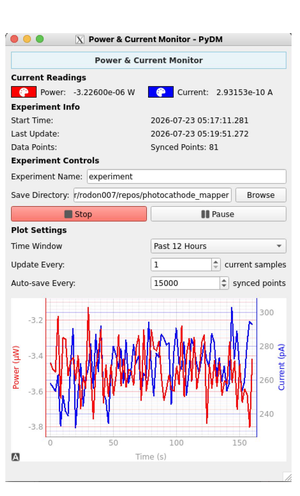
A widget that provides a directional-pad tip/tilt control for a paired vertical/horizontal motor (EPICS motor-record or SmarAct piezo), with step & stop buttons, axis-invert checkboxes, and an expert-screen shortcut.

OTHER CONTRIBUTIONS



RARITAN PDU (IOC + SCREEN)

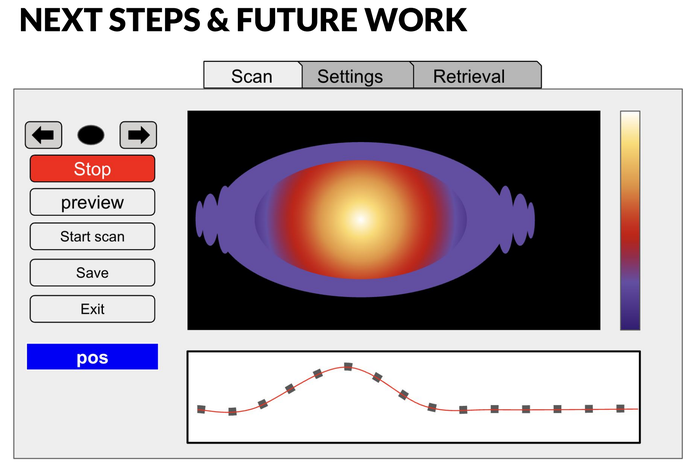
Added EPICS database templates (PV records) for monitoring and controlling Raritan PDUs via SNMPv2c. Developed a PyDM UI screen for Raritan PDU (inlet/outlet status and control), needed for easy PV interactions.



SPARSE PLOTTER

Real-time synchronized logging of two mismatched-rate channels (power + current) with dual-axis plots and auto-data saving. App includes experiment life cycle controls, axis color selection, experiment naming, and plot settings.

NEXT STEPS & FUTURE WORK



Build a Frequency-Resolved Optical Gating (FROG) screen (Scan / Settings / Retrieval), driving one linear stage + one spectrometer for pulse compression & phase retrieval.

TOOLS / TECHNOLOGIES USED

PyDM · PyQt · EPICS · PyQTGraph · areaDetector · Qt Designer · SNMP · Smaract · Python · Motor Record ·

Acknowledgement

I would like to thank my mentor, Adam Berges, for his guidance and support throughout this internship and with my project. I also want to thank Ilia Potanin, and the ECS team for their help throughout the internship. A shout out to all the interns and employees who made the summer interesting. Lastly, I'd like to give a special thanks to Alan Fry, Arturo Garcia, Hillary Freeman for organizing and hosting the LCLS Intern program.

References

[1] EPICS. EPICS — *Experimental Physics and Industrial Control Systems.* https://docs.epics-controls.org
[2] SLAC National Accelerator Laboratory. *PyDM — Python Display Manager.* https://slaclab.github.io/pydm
[3] SLAC PCDS (Photon Control and Data Systems). *pcdswidgets — LCLS PyDM Widget Library.* https://github.com/pcdshub/pcdswidgets