ICALEPCS2017
Barcelona · Spain, October 8-13
Palau de Congressos de Catalunya
www.icalepcs2017.org

Skywalker

Automated Alignment at LCLS

Teddy Rendahl
Alex Wallace
Abdullah Ahmed Rashed

U.S. DEPARTMENT OF ENERGY
Office of Science

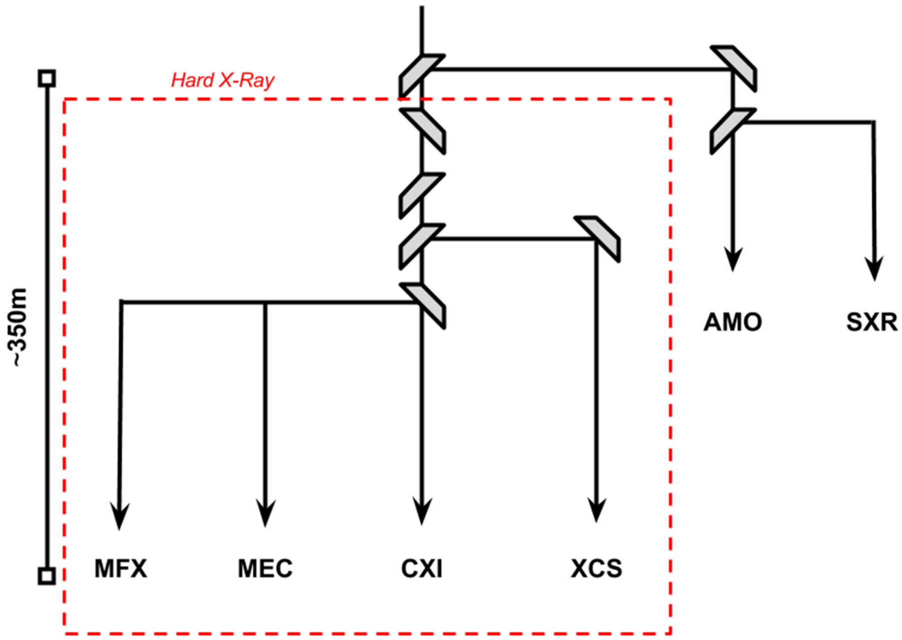
SLAC NATIONAL ACCELERATOR LABORATORY

Outline

- Motivation
- Project Goals
- Stepping Stones
- Aligning LCLS
- Next Steps

Motivation

- Seven experimental hutches, each requiring unique pointing of flat mirror systems
  - Each with sensitive downstream optics
  - Common for experiments to run in serial
- LCLS-II upgrade in 2019 adds more mirrors and more endstations
  - Emphasis on automation



Why Skywalker?

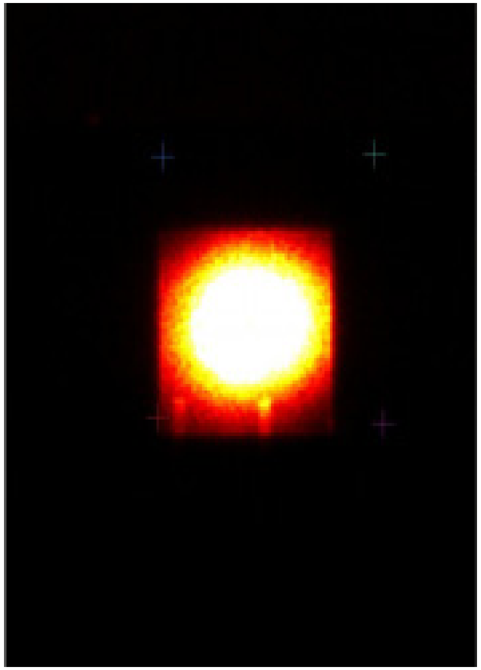
DAQ BCS OCIO DOE FDR CDS SRD API PPS CXI POC TXI XPP MFX MPS ICL PDR TLA ECS CDR ACR ANA SXR

Project Skywalker

- **Manual Alignment**
  - Pointing of the FEL is not repeatable enough for `set and forget` values
    - No diagnostics sensitive enough to determine undulator pointing for optics 350m away
  - Over 190 devices in the common areas of the beamline
- **Skywalker Deliverables**
  - Operators should be able to `single-click` align to any of Hard X-Ray endstations
    - Should be done faster than manual alignment (or time claimed by operators for manual alignment)
  - Deal with dynamic target selection
  - Full automation
    - Watch for drops in FEL energy
    - Clear the beamline of obstructions
    - Durable against day to day operation
  - Create a suite of tools for future automation projects

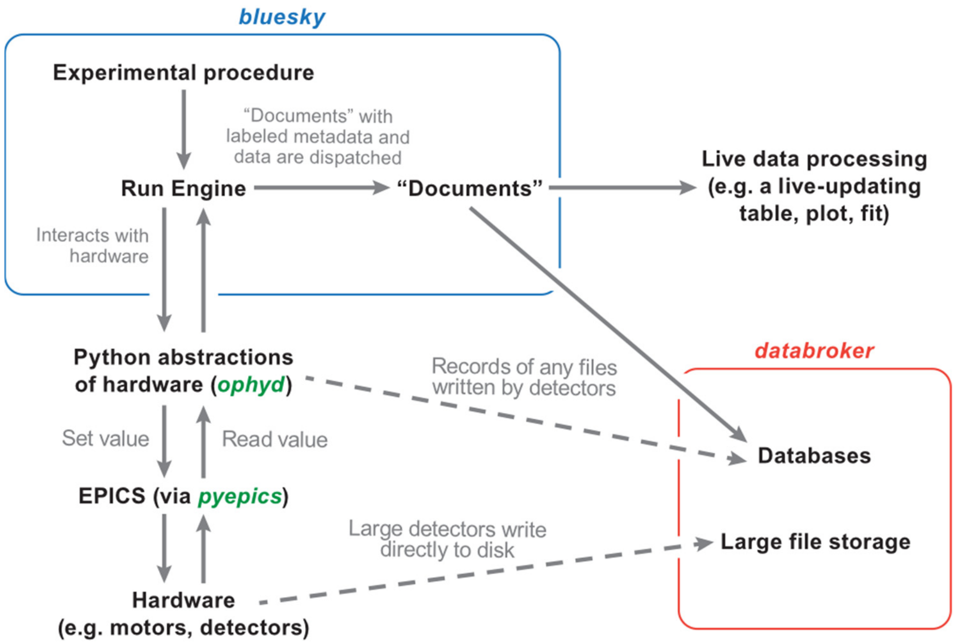
The Tools

- Flat Mirror System
  - Controlled by ELMO Drives with a Beckhoff PLC Ethercat Master
  - Stepper and piezo in series on the pitch mechanism
- 4 Jaw-Slits
  - Produced by JJ-XRAY
  - Use EPICS motor record
- Imagers
  - YAG crystal fluoresces when X-Rays are incident
  - EPICS AreaDetector
- Software
  - Python 3.5 and up
  - Ophyd and Bluesky

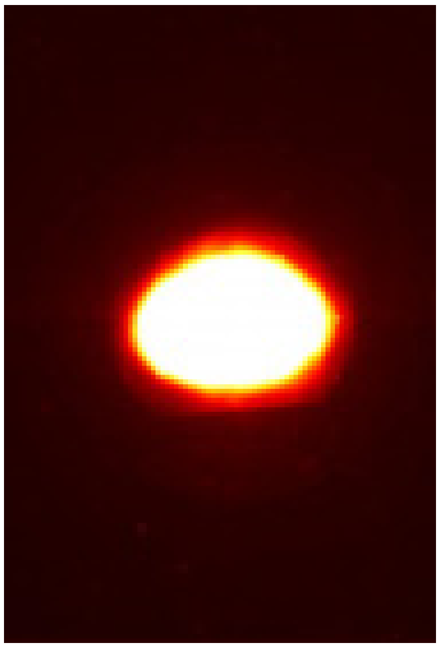


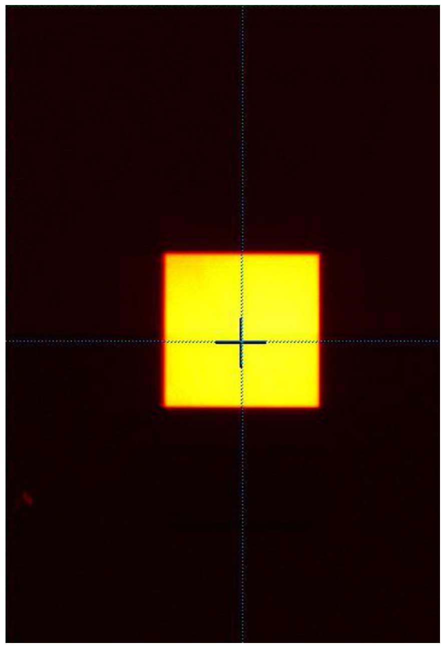
Bluesky

- Developed at NSLS-II
- Abstracts experimental plans into Python generators
  - Allows for adaptive plans
- RunEngine controls plan execution
  - Start / Stop / Pause
  - Emits live scan data as `Documents` for long term storage and live feedback for operators



Defining Alignment

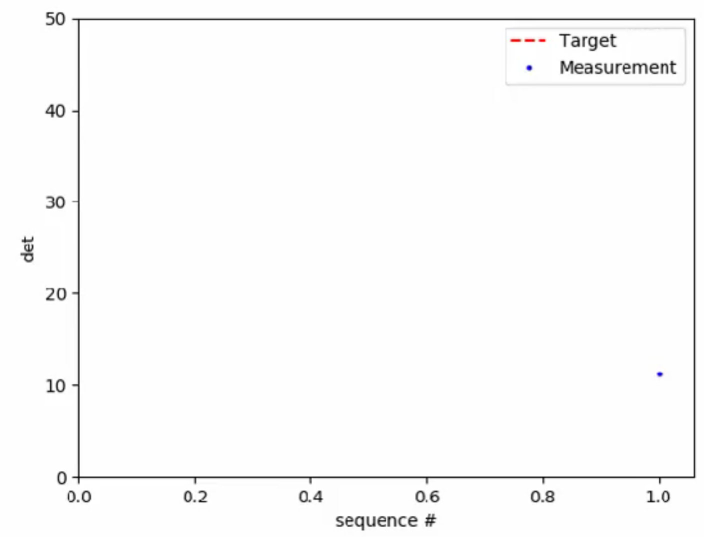




- Fiducialize imager by using previously aligned 4-Jaw slits
  - Trim the beam down to only a small subsection that is aligned
  - Calculate the centroid of this subsection
  - Expand slits and difference between the open and closed centroid is the error in our pointing

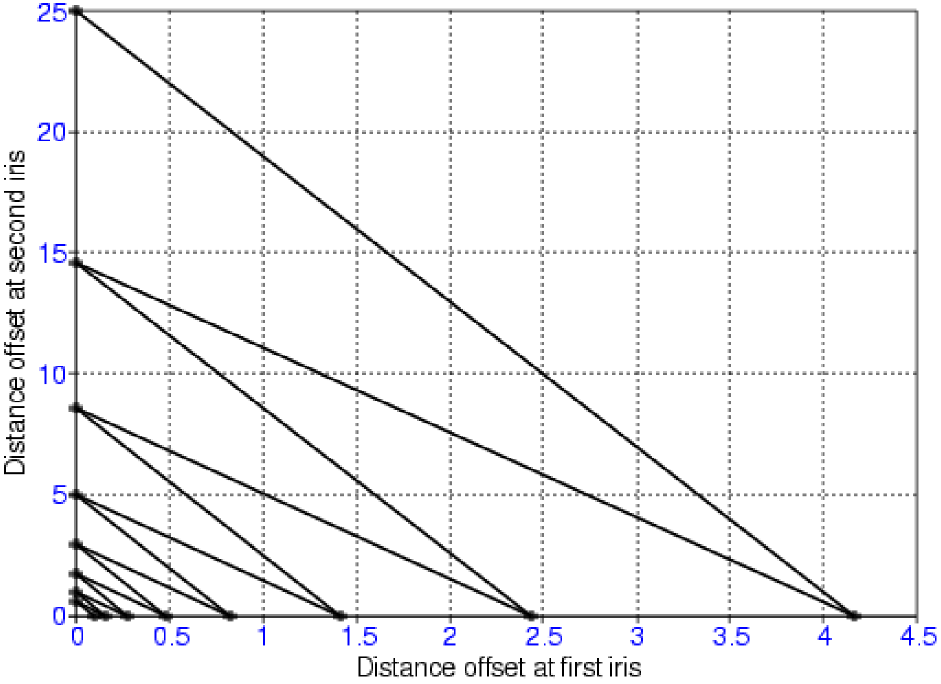
One Step At a Time …

- Require as little prior knowledge as possible
- Formulate a mathematical relationship between a motor position and our detector signal
  - Prior knowledge can be included as our 'initial guess' for the model
- Take a single `naïve step` to explore the parameter space, feeding all information into our model
- Query the model for the motor position that meets our target value
- Repeat until our detector readout is within the operator specified tolerance

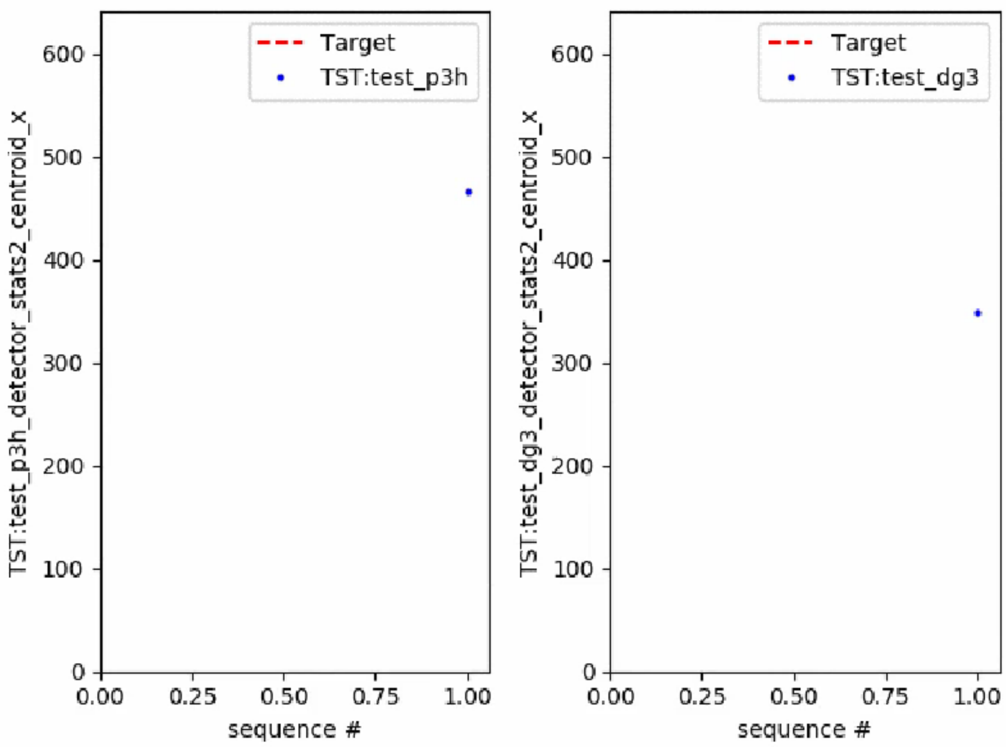


Two Mirror Alignment

- Used to create a shared central alignment
- Iterate between two different mirrors and two different fiducialized imagers to create an aligned axis
  - Fastest rate of convergence is found by choosing one imager as close as possible to the two mirror system and one as far away
- Scaling tolerance accelerates convergence, but still demands high degree of pointing accuracy



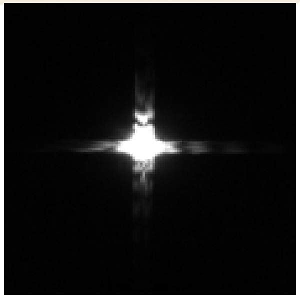
In Action …

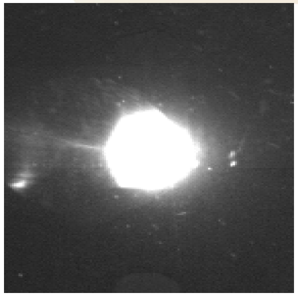


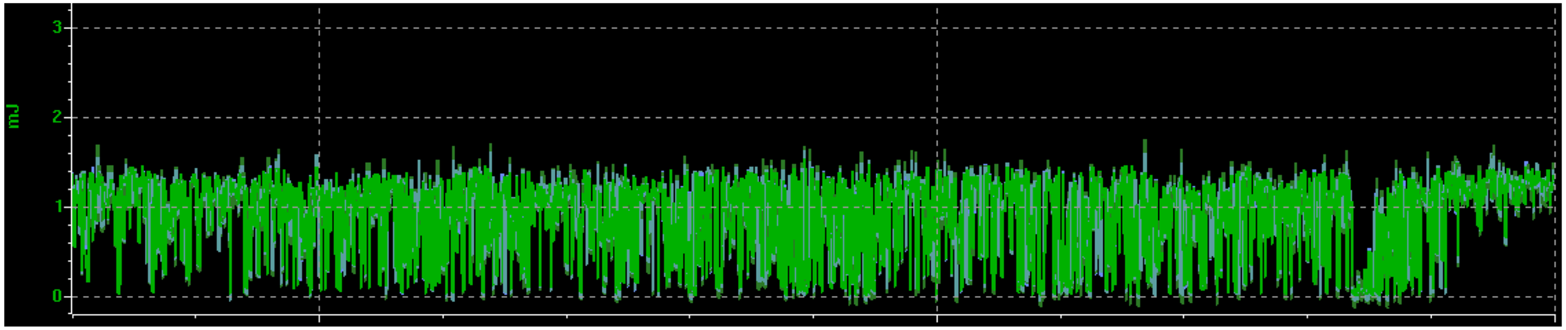
Target
TST:test_p3h
TST:test_p3h_detector_stats2_centroid_x
600
500
400
300
200
100
0
0.00
0.25
0.50
0.75
1.00
sequence #
Target
TST:test_dg3
TST:test_dg3_detector_stats2_centroid_x
600
500
400
300
200
100
0
0.00
0.25
0.50
0.75
1.00
sequence #

Hurdles

- Unstable FEL
  - Heavy use of Bluesky `suspenders`
  - Filter event data before it gets to our model
- Optical Phenomena
  - Homegrown image processing library `psbeam` allows more complex image filtering







Commissioning Results

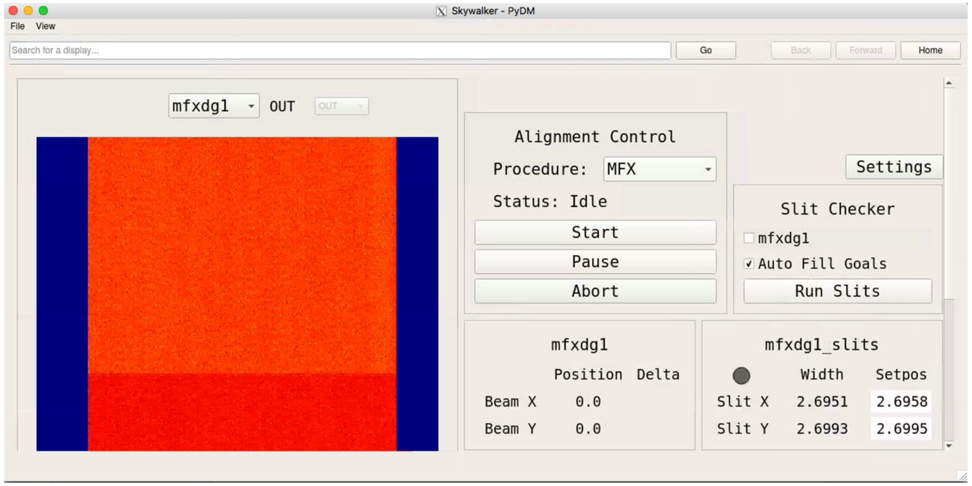
- Single Mirror Alignment
  - No prior knowledge
  - Capable of pixel precision (3.4 um)
    - Accomplished in roughly 60 seconds
  - Majority of time spent during fine adjustment
    - Within 10 pixel tolerance after first naïve step
  - Jupyter Notebook
- Two Mirror Alignment
  - Able to reliably solutions within two pixels
  - Starting with no prior knowledge, no beam on imagers
    - 7 minutes
  - Seeding the run with approximate values
    - 90 seconds

Clearing the beamline ….

- Happi (*Heuristic Access to Positions of Photon Instruments*)
  - Database that contains device positions
  - Stores relevant metadata for alignment
  - Flexible backend support
- Lightpath
  - Interpret complex devices to report a transmission
    - Simple device are binary, others require more care
  - Alert operators to blockages in the beamline
  - GUI and underlying Python objects generated on the fly from happi

Immediate Agenda

- PyDM UI beta-test begins next week
  - *PyDM Speakers Corner-> Thursday 5:45*
- Create long term storage for alignment results
- Hard X-Ray Split and Delay System
  - *HXRSnD Poster -> Thursday 4:45*
  - System of eight crystals
  - Maximization of signal instead of point to point
- Full separation of LCLS specific routines and modelling toolkit
  - Either as separate module or as an extension of bluesky



Future Improvements

- Automated tuning
  - Add a short diagnostic run at the beginning of the scan to inform parameter choices
- Easy to imagine continued deployments
  - Feedback system for mirror curvature
  - Visual light lasers
- Automated mirror centering
- Passive monitoring system to alert the operator when a realignment is needed
- Inclusion of LCLS undulators to include vertical pointing

Links

Conda Channel

- **skywalker-tag** -> Most recent tagged build
- **skywalker-dev** -> Bleeding edge
- **lightsource2-tag** -> Ophyd and Bluesky

Documentation

- https://pswww.slac.stanford.edu/swdoc/releases/skywalker
- https://nsls-ii.github.io/ophyd
- https://nsls-ii.github.io/bluesky

Source

- https://github.com/slaclab/skywalker
- https://github.com/slaclab/pswalker
- https://github.com/slaclab/lightpath
- https://github.com/slaclab/happi
- https://github.com/slaclab/psbeam
- https://github.com/NSLS-II/ophyd
- https://github.com/NSLS-I/bluesky

Special Thanks

- Daniel Flath
- Alex Wallace
- Zachary Lentz
- Abdullah Ahmed Rashed
- Hugo Slepicka
- Diling Zhu
- Karl Gumerlock
- Daniel Ballan
- Thomas Caswell
- Ken Lauer
- Matt Gibbs