GLARE: The Generalized Lab Architecture for Restructured optical Experiments

Fowler[1], J., Noss[1], J., Laginja[1], I., Soummer[1], R., Perrin[1], M., Pogorelyuk[2], L., Sun H.[3]

Space Telescope Science Institute[1], Princeton University[2], California Institute of Technology[3]

Experiment-agnostic Hardware Control

While the experiments in the Makidon Lab are unique, the hardware we use is shared by optics labs around the world. GLARE provides a standardized library of these common hardware controllers, hence forth called `catkit` (the Control and Automation for Testbeds Kit.)

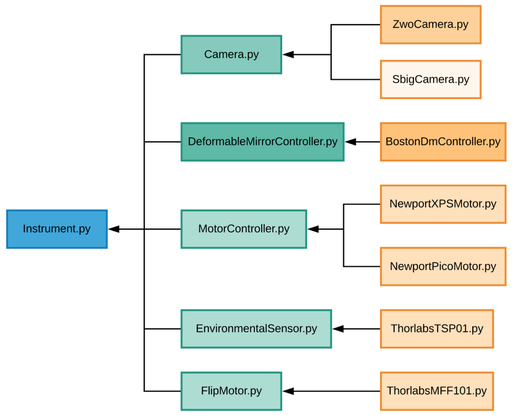

Figure 1: *A sample set of our controller interfaces. Every hardware controller includes three levels of abstraction: (1) `Instrument.py`, which contains functionality shared by every hardware controller, e.g. safely opening and closing connections to devices (with context managers), (2) a layer specific to the type of hardware, e.g. `Camera.py` with imaging functions, and (3) a connection to the specific hardware brand, e.g. `ZwoCamera.py` with functions that operate the ZWO Cameras. In this diagram higher opacity maps to more complete interfaces.*

Optical Simulators and Hardware Emulators

HiCAT has the ability to run entirely in simulation, which allows for continuous integration testing, as well easy as collaboration.

- We have developed a moderately high fidelity optical model and simulator for HiCAT, built using the POPPY Python Fourier optics toolkit (Perrin, 2012).
- We have written custom hardware emulators that respond like hardware, by emulating the manufacturer provided hardware library.

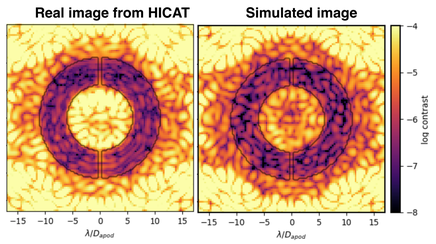

Figure 2: *Left: Real data from the HiCAT testbed. Right: Simulated data run offline.*

```python
# in hicat
if testbed_state.simulation:
      return PoppyBostonDMController(...)
else:
      return BostonDmController(...)

# in catkit
class BostonDmController(DeformableMirrorController):

      instrument_lib = bmc
      ...

class PoppyBostonDMController(BostonDmController):

      instrument_lib = PoppyBmcEmulator
```

Figure 3: *Whereas `bmc` is the manufacturer provided hardware library for Boston Deformable Mirrors, `PoppyBmcEmulator` emulates the library. The `PoppyBostonDMController` serves only to point to the simulated library, otherwise inheriting all functionality from `BostonDMController`, allowing the two classes to have identical functions and calls.*

GLARE, the Generalized Lab Architecture for Restructured optical Experiments, is a restructuring of the Makidon Lab systems and software to (1) write `catkit`, a general library of hardware interfaces, (2) develop HiCAT-like autonomous experiments for every testbed, and (3) incorporate control algorithms from our collaborators into our experiments and run them on the HiCAT testbed. We present a summary of our work on this project, including the development of additional hardware interfaces, the design of the controller software, and the initial results of our collaborators' algorithms on the HiCAT testbed.

Observing Exoplanets with High Contrast Imaging

Direct observations of earth-like planets will require a contrast of 10^{-10} between a planet and its host star; technology development and laboratory demonstrations are key to enabling these future space missions. With HiCAT, the High-contrast imager for Complex Aperture Telescopes testbed (Soummer, 2018), the Makidon Lab aims to demonstrate segmented aperture coronagraphy in air with goal of 10^{-7} to 10^{-8}. Using advanced coronagraphs, innovative methods for deformable mirror control, and autonomous experiment software to run our testbeds and data processing without requiring physical people in the lab, we pave the way for missions like LUVOIR (the Large UV, Optical, and IR telescope.)

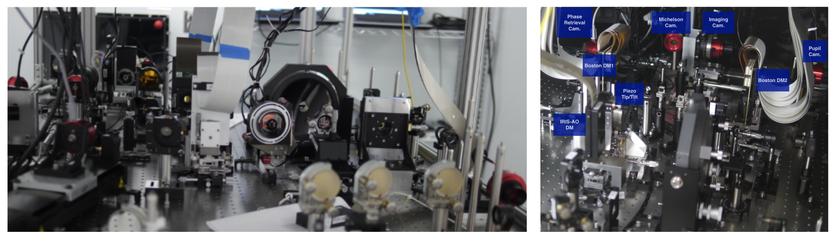

Figure 4: *Left: HiCAT testbed in our clean room at the Makidon Lab. Right: Main hardware components of the HiCAT testbed.*

GLARE in Practice

In practice, hardware interfaces are called by a package specific to the experiment – in our case HiCAT, as part of a `hicat` software package.

```python
# in hicat

from hicat.hardware.testbed import dm_controller

with dm_controller as dm:
      dm.apply_shape()


from catkit.hardware.boston.BostonDmController import BostonDmController

def dm_controller(...):
      if testbed_state.simulation:
            return PoppyBostonDMController(...)
      else:
            return BostonDmController(...)


# in catkit

from catkit.interfaces.DeformableMirrorController import DeformableMirrorController

class BostonDmController(DeformableMirrorController):
      ...


from catkit.interfaces.Instrument import Instrument

class DeformableMirrorController(Instrument)
      ...


class Instrument():
      ...
```

Figure 5: *Command operation: `hicat` applies a shape to a Boston MEMs deformable mirror. This example describes the full path of the command operation (in `hicat`), which calls to the `BostonDeformableMirror` class in `catkit`, which inherits the `DeformableMirrorController` hardware interface, which inherits `Instrument`.*

Collaborators' Algorithms on the HiCAT testbed

With more flexible software and a fully simulated testbed we have incorporated our collaborators' control algorithms, both physically on our testbed, and via simulation.

- We are actively collaborating with He Sun at CalTech, who is using simulated HiCAT experiments to test machine learning DM Control solutions (Llop-Sayson, 2019; Sun, 2018).
- With a recent visit from Leonid Pogorelyuk at Princeton, we sucessfully incorporated a dark hole maintenance algorithm (Pogorelyuk, 2019) into the HiCAT testbed, which uses non-linear filtering to estimate and correct for drifts in the testbed state after a dark hole is achieved.

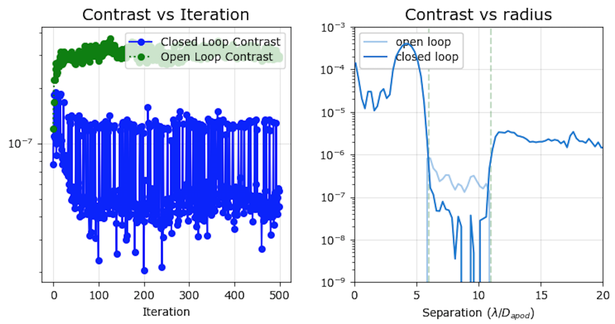

Figure 6: *Initial results of a dark hole maintenance algorithm run on the HiCAT testbed. Left: Contrast with iteration. Green demonstrates the testbed state without maintenance, where an applied drift alters the contrast, as opposed to the blue, where the contrast is not affected by drift. Right: Final contrast over the dark hole. For more information see Leonid Pogorelyuk's talk, # 320.05*

Next Steps

- Finish writing and restructuring hardware interfaces and controllers.
- Write a new suite of autonomous experiment software for a UV Multi Object Spectroscopy Testbed, characterizing deformable mirrors, the Space Telescope UV Facility (STUF), as a new implementation of GLARE.
- Public release of GLARE for collaborators and other lab facilities.

Acknowledgements

Support for HiCAT is provided by NASA APRA (NNX12AG05G, NNX14AD33G), NASA SAT TDEM (80NSSC19K0120), and by the STScI Director's Discretionary Research Fund. Support for GLARE is provided by the STScI Director's Discretionary Research Fund and the STScI Data Science Innovation Initiative.

References

Jorge Llop-Sayson et al., "The high-contrast spectroscopy testbed for segmented telescopes (HCST): new wavefront control demonstrations," Proc. SPIE 11117, Techniques and Instrumentation for Detection of Exoplanets IX, 111171W (9 September 2019);

Marshall D. Perrin et al., "Simulating point spread functions for the James Webb Space Telescope with WebbPSF," Proc. SPIE 8442, Space Telescopes and Instrumentation 2012: Optical, Infrared, and Millimeter Wave, 84423D (21 September 2012);

Pogorelyuk, Leonid et al. (2019). Dark Hole Maintenance and A Posteriori Intensity Estimation in the Presence of Speckle Drift in a High-contrast Space Coronagraph. The Astrophysical Journal. 873. 95. 10.3847

Rémi Soummer et al., "High-contrast imager for complex aperture telescopes (HiCAT): 5. first results with segmented-aperture coronagraph and wavefront control," Proc. SPIE 10698, Space Telescopes and Instrumentation 2018: Optical, Infrared, and Millimeter Wave, 106981O (21 August 2018);

He Sun et al., "Identification and adaptive control of a high-contrast focal plane wavefront correction system," J. Astron. Telesc. Instrum. Syst. 4(4) 049006 (8 December 2018)