

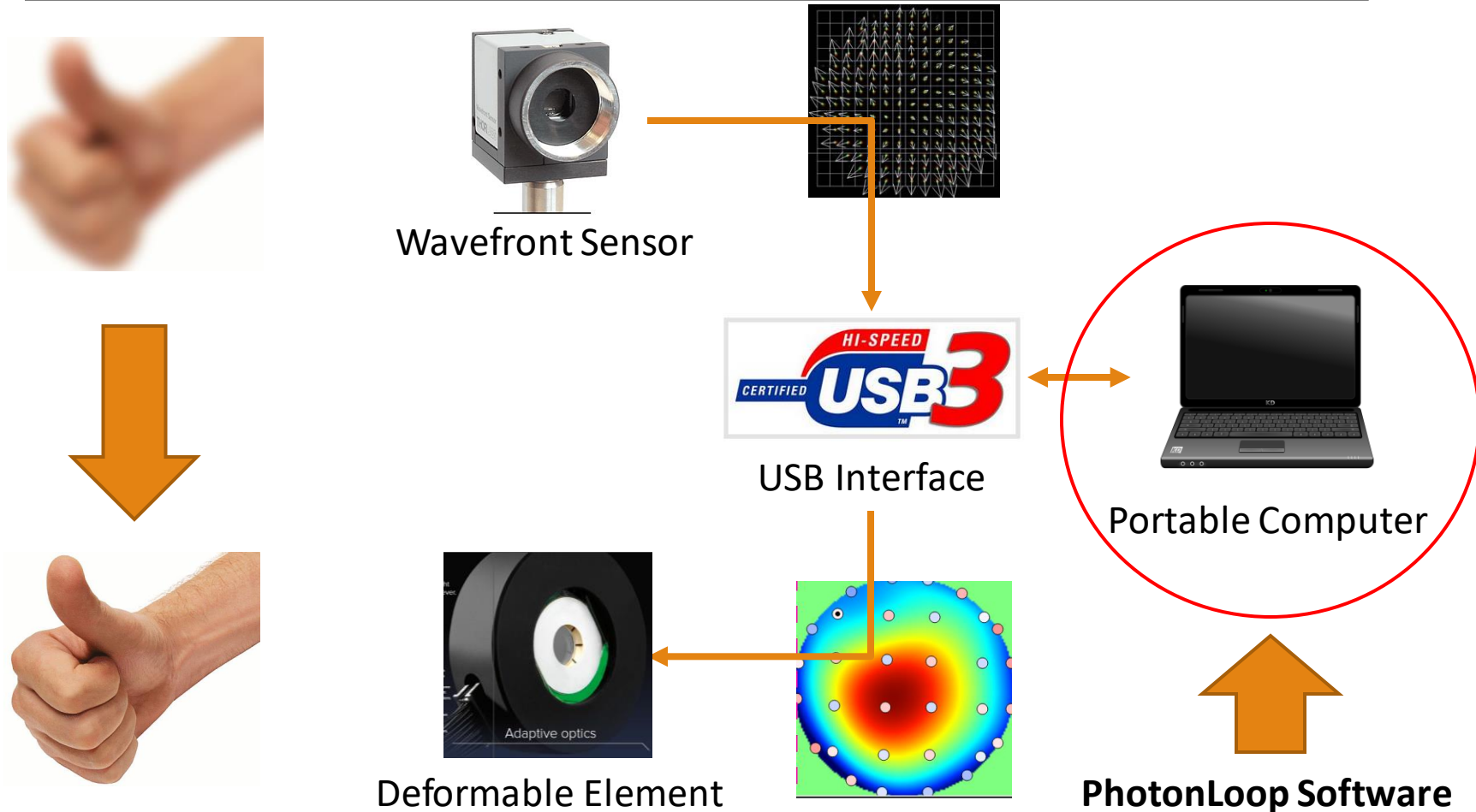
A flexible and user friendly CPU-based AO software

J. MOCCI, M. QUINTAVALLA, C. TRESTINO, S.
BONORA AND R. MURADORE

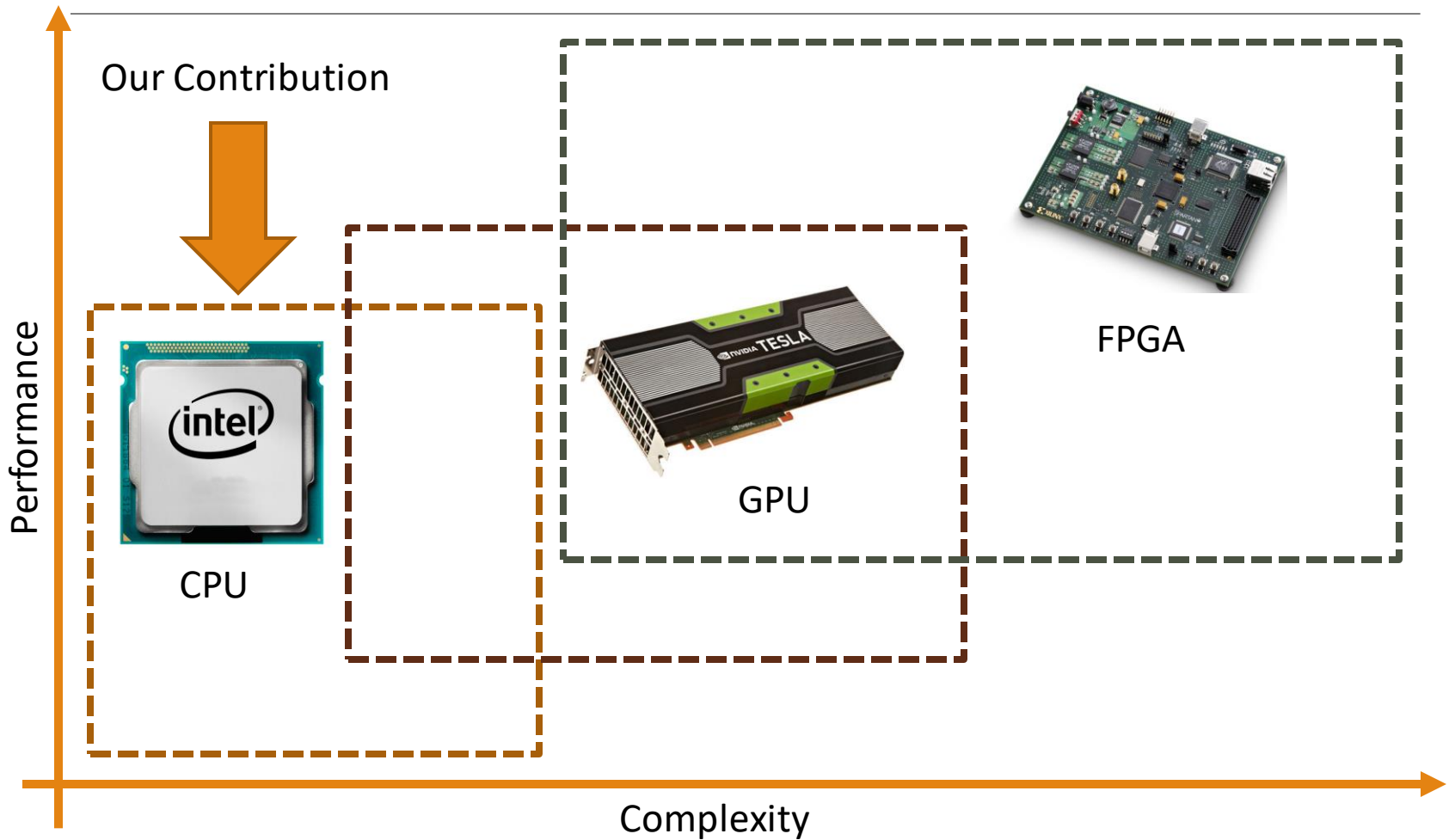
UNIVERSITÀ DEGLI STUDI DI VERONA
CNR-IFN, PADOVA



Adaptive Optics – Basics



Adaptive Optics – Architecture



PhotonLoop – Introduction

PhotonLoop is a **flexible** and **user-friendly CPU-based Adaptive Optics** software.

Measures:

- **Zernike polynomials** and **wavefront shape** from wavefront slopes (using a Shack-Hartmann WFS).

Corrects:

- **closed-loop** and **open-loop** with PI control; **Hadamard** calibration; can apply **Zernike offset** to be generated by the controller.

PhotonLoop – Introduction

Flexible:

- **any size** of WFS aperture (any number of centroids); **any number** of DM actuators; fully scriptable.

User Friendly:

- **graphical representation** of all data structures; advanced tools for WFS and DM calibration; **real-time** performance monitors.

Fast:

- closed-loop frequency **up to 500Hz**; total closed-loop latency of 2-3 ms (**2 frames**).

PhotonLoop – Development

C++:

- guarantees **high performance** and promotes **Object-Oriented Programming (OOP)**.

Qt Framework:

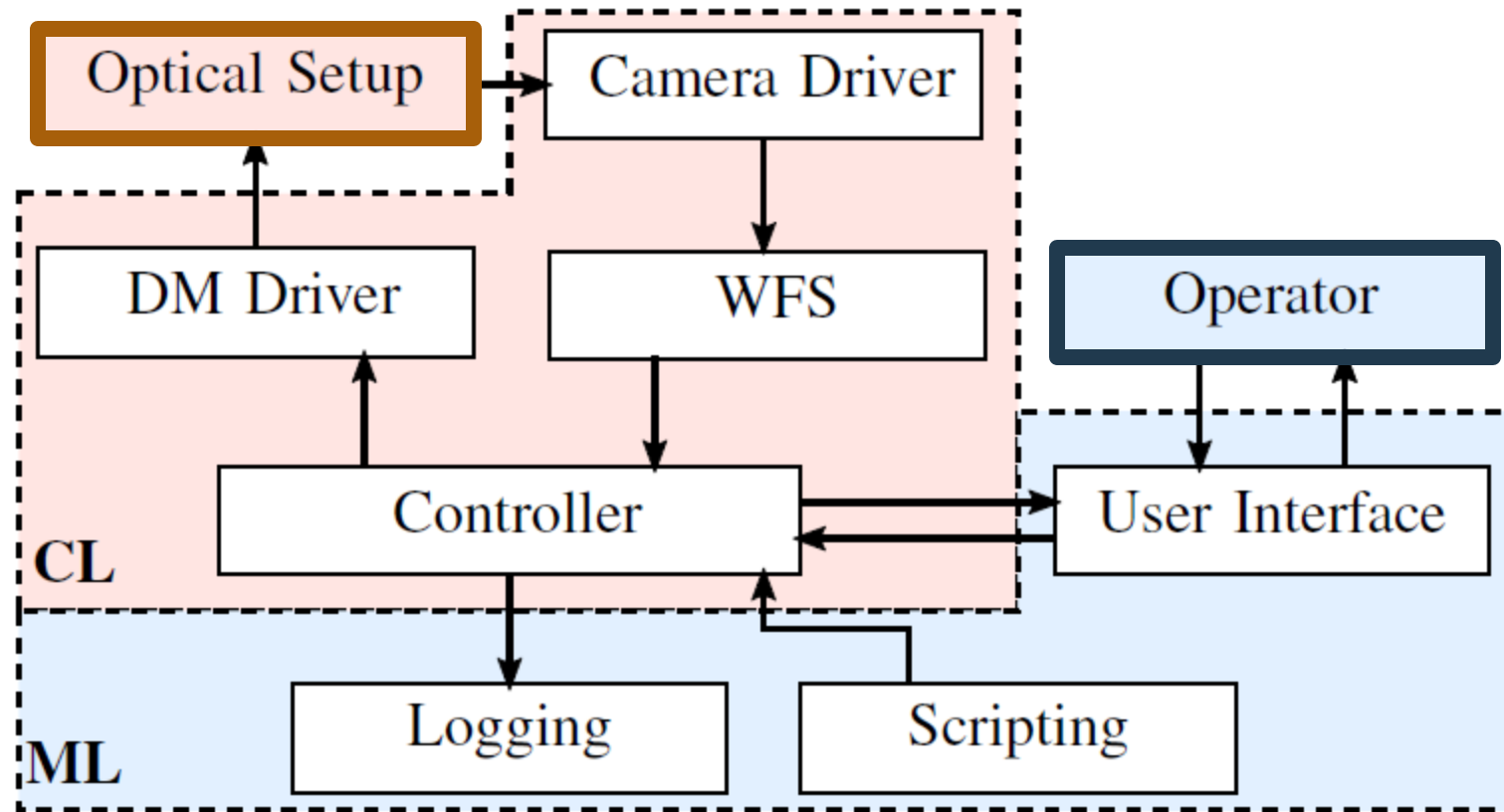
- **cross-platform** utility libraries for GUI and OS-dependent operations.

Eigen:

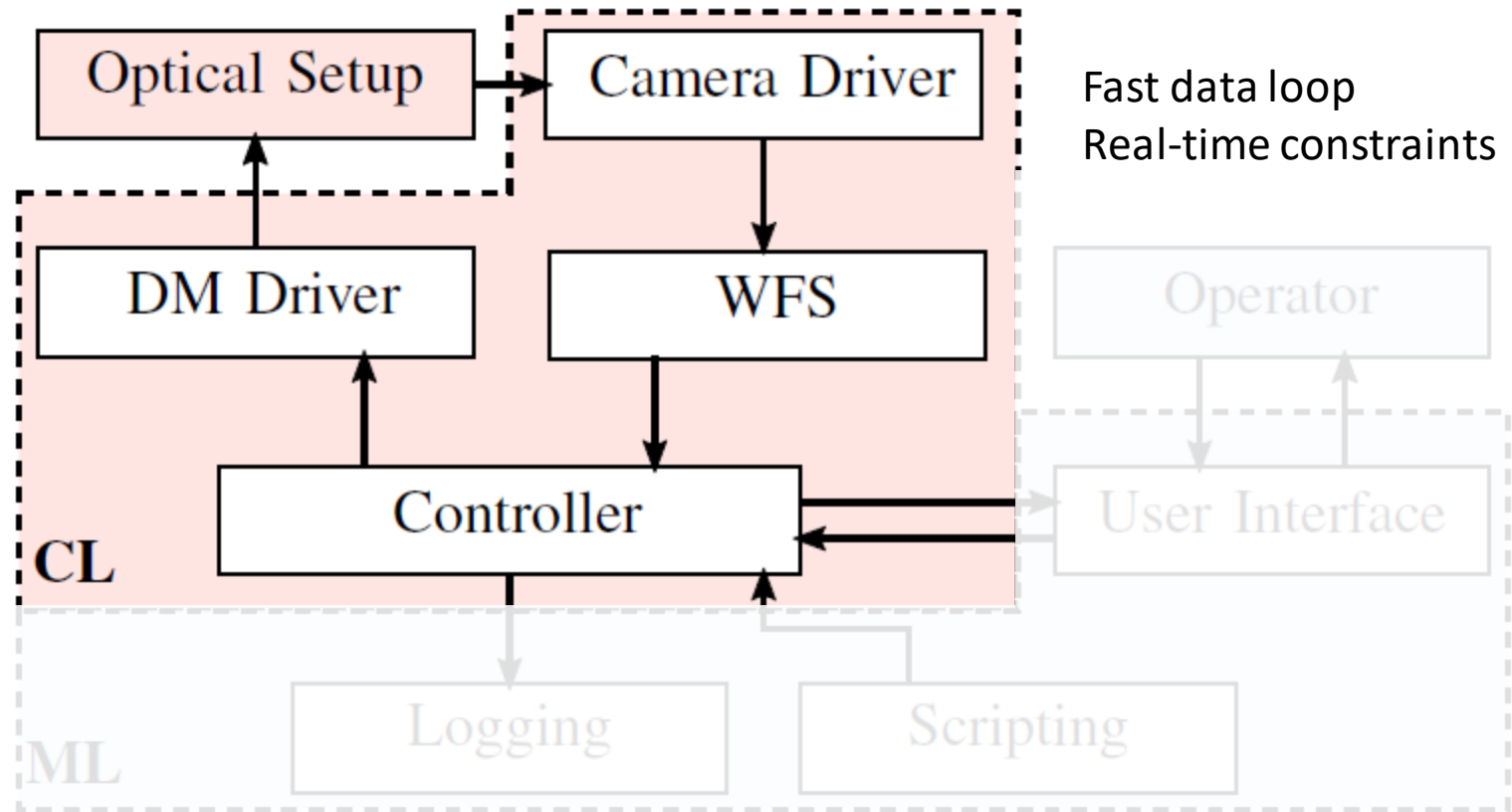
- **efficient** vector and matrix operations.



PhotonLoop – Architecture



PhotonLoop – Control Layer



PhotonLoop – WFS

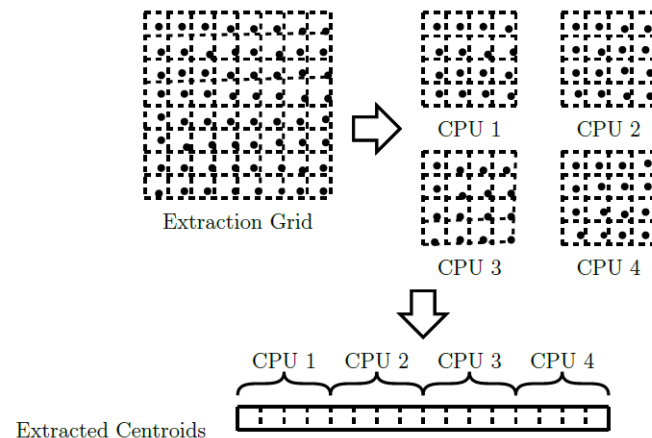
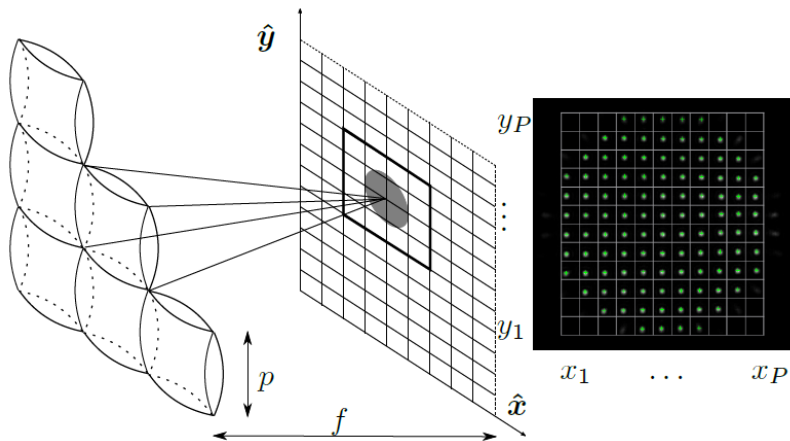
Fast centroiding algorithm:

- Uses Thresholded Weighted Center of Gravity.

Operation on **integers** for speed.

Parallellized:

- independent centroids are offloaded to the single CPU cores



PhotonLoop – WFS

Reference

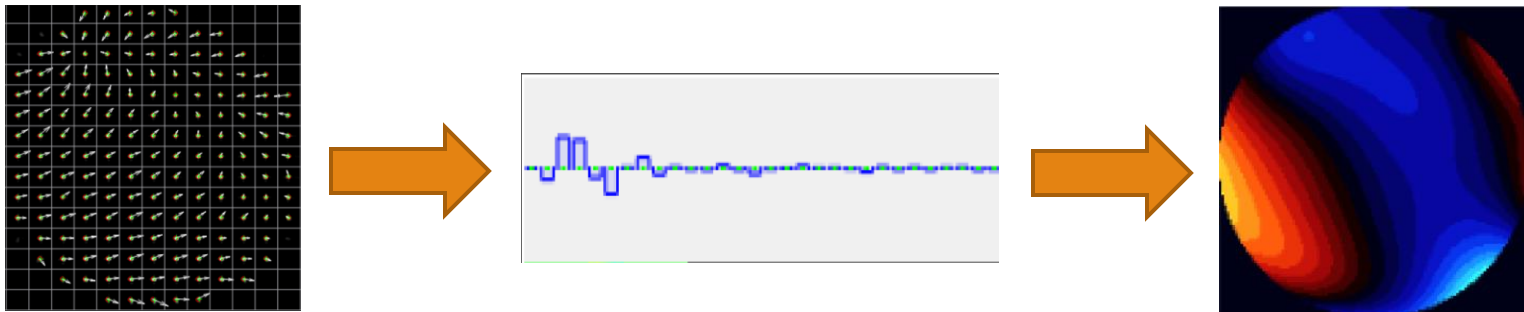
- Can remove **global tilt**; can be set **relative** to centroids.

Zernike

- Decompose the **wavefront slopes** to any number of **Zernike terms** with least-squares methods.

Reconstruction

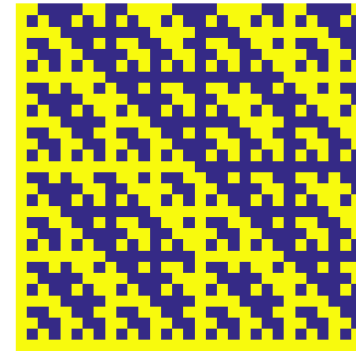
- Uses **modal reconstructions** from Zernike terms



PhotonLoop – Controller

Calibration:

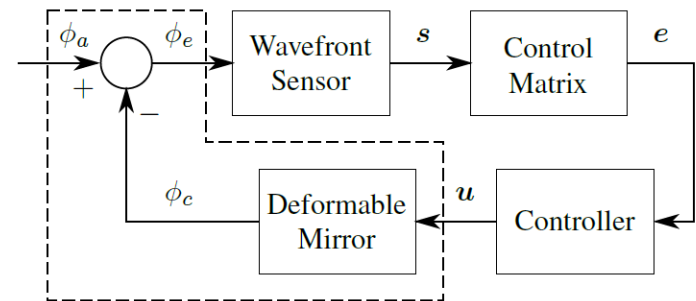
- DM is calibrated with **Influence Functions** or **Hadamard patterns**
- the interaction matrix is processed with **SVD decomposition**



Hadamard Matrix

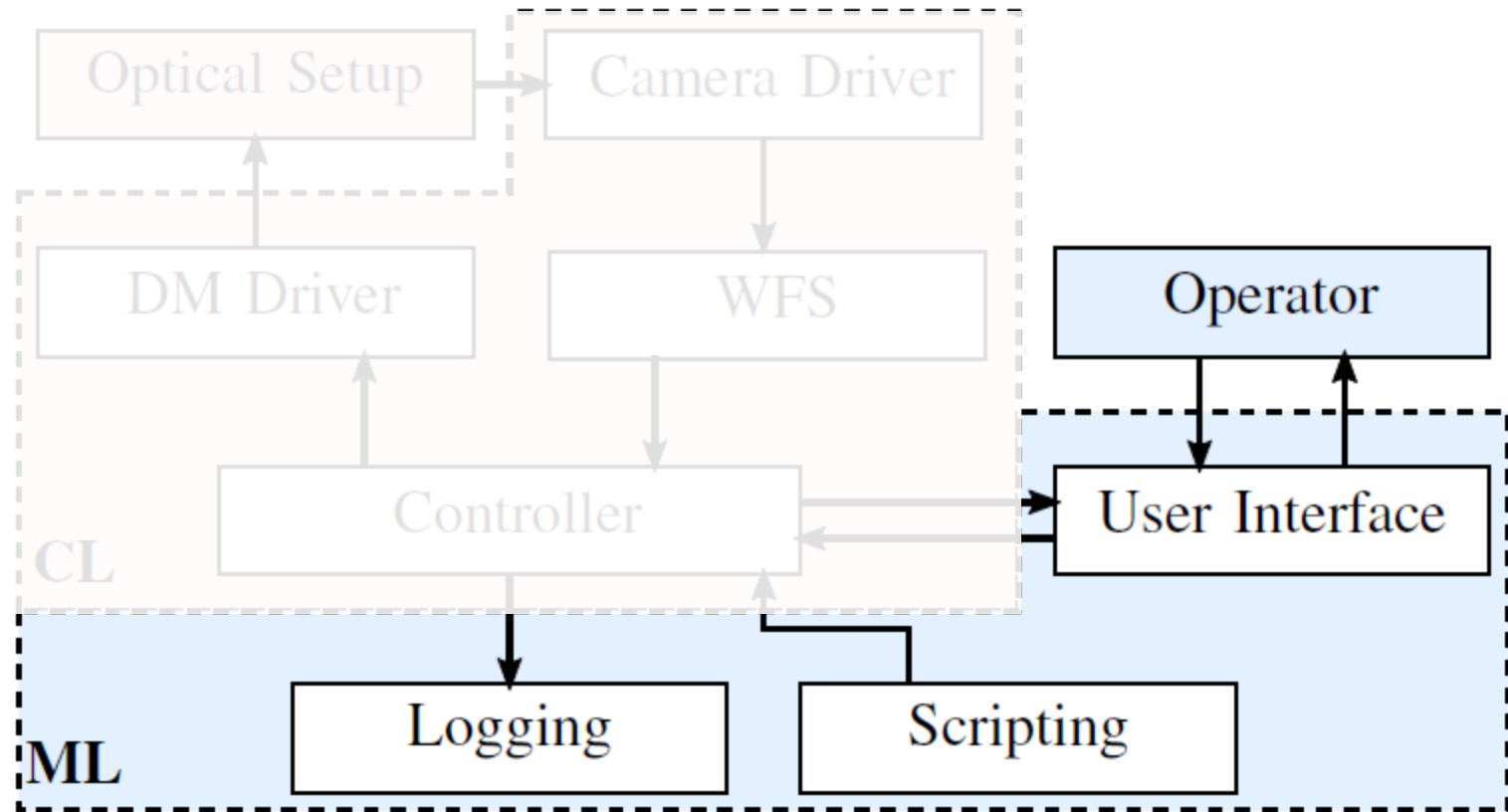
Closed Loop:

- parallel array of **PI compensators** with **anti wind-up** to avoid actuators saturation



Controlled Plant Scheme

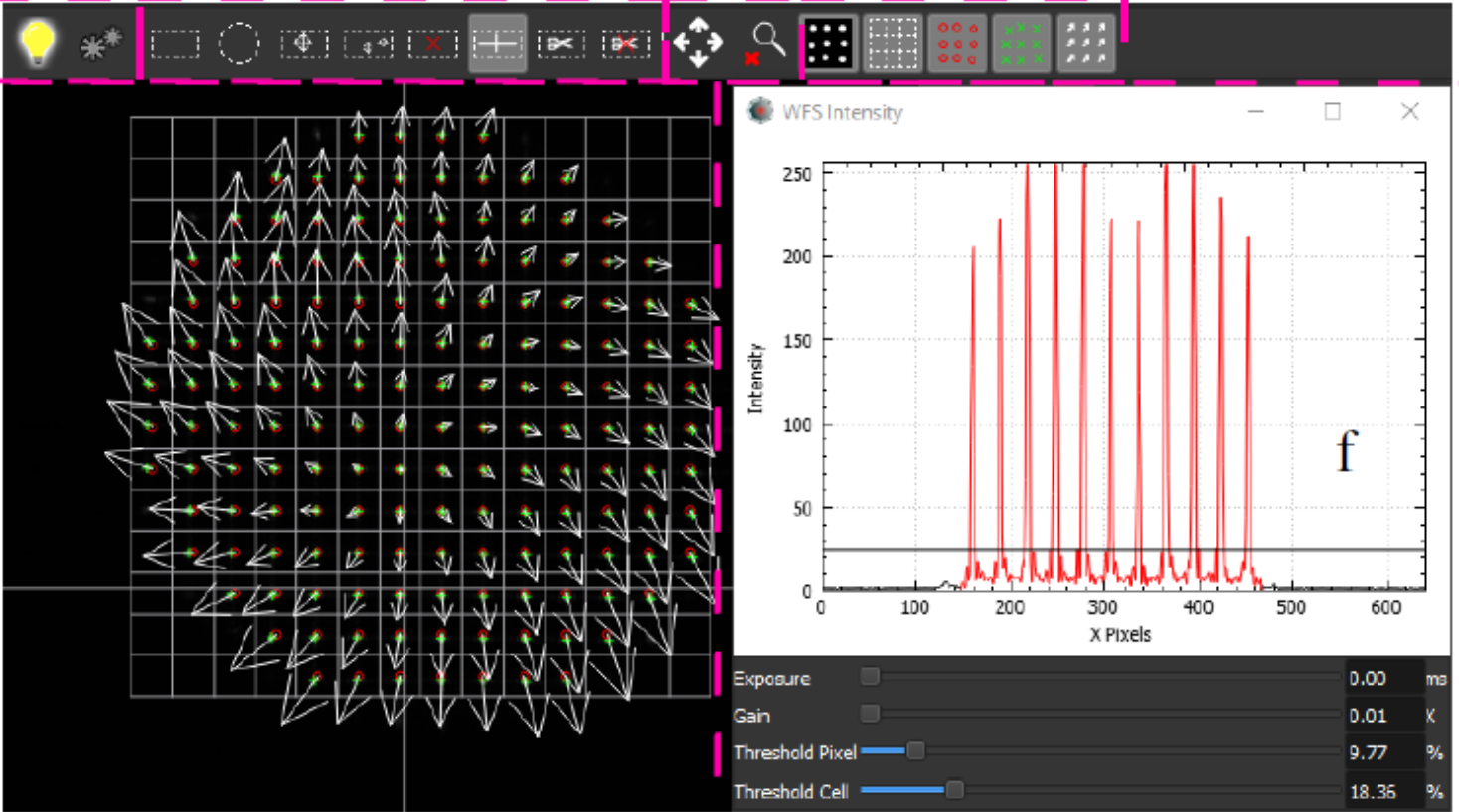
PhotonLoop – Monitor Layer



PhotonLoop – User Interface

Tools **Grid Tools** **Full Screen** **Visualization Toggles**

Slopes

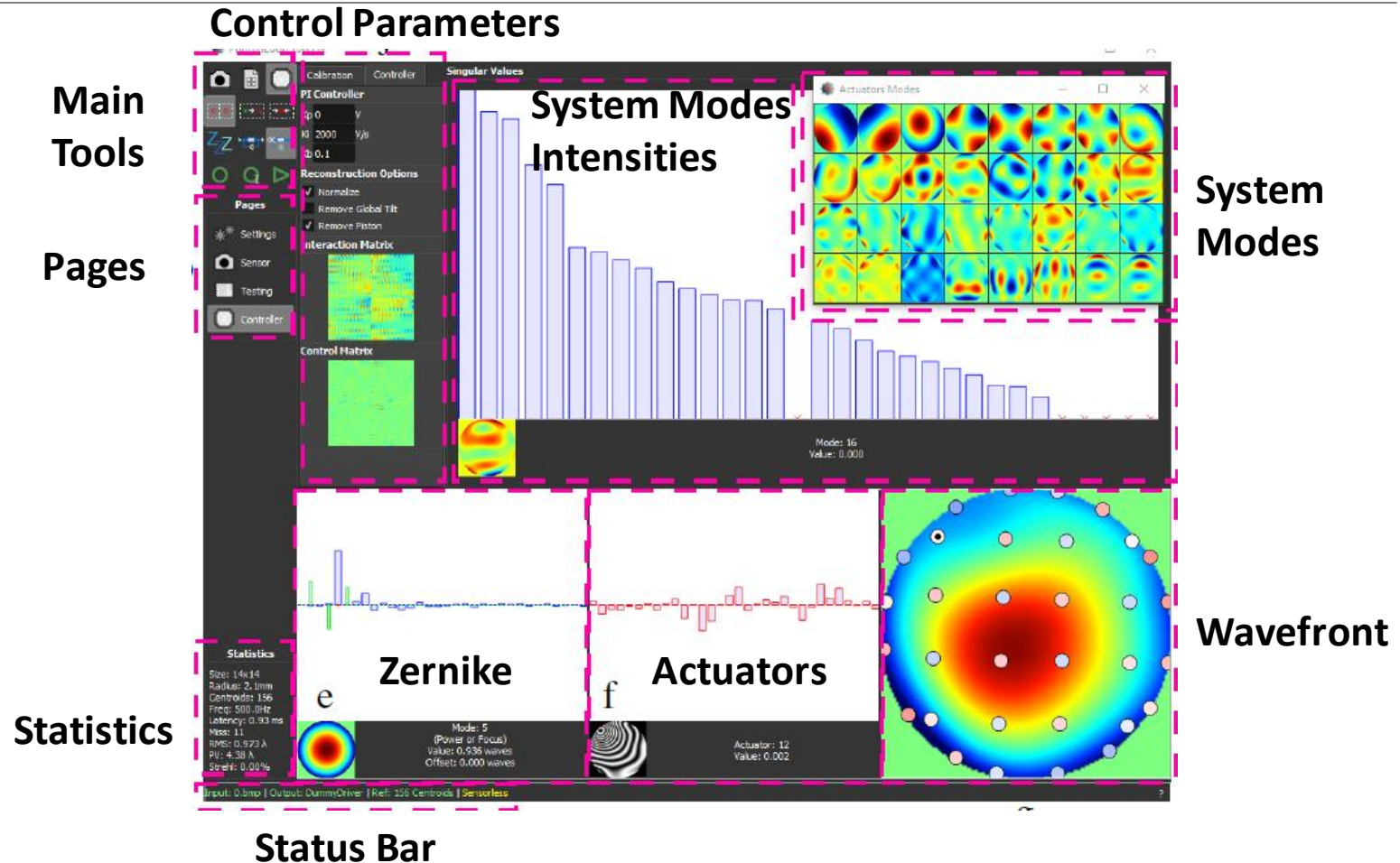


Intensity Graph

The screenshot displays the PhotonLoop software interface. At the top, there are four main menu categories: Tools, Grid Tools, Full Screen, and Visualization Toggles. The Tools menu includes icons for a lightbulb, a starburst, a rectangle, a circle, a crosshair, a plus sign, a minus sign, a double-headed arrow, a magnifying glass, and a grid. The Grid Tools menu includes icons for a grid, a grid with a red dot, a grid with a green dot, and a grid with a blue dot. The Full Screen menu includes a double-headed arrow icon. The Visualization Toggles menu includes a grid icon, a grid with a red dot, a grid with a green dot, and a grid with a blue dot. The main window is divided into two panes. The left pane, labeled 'Slopes', shows a grid of white arrows pointing in various directions, with some arrows having small colored dots (red, green, blue) at their tips. The right pane, labeled 'Intensity Graph', shows a plot of Intensity versus X Pixels. The plot has a y-axis from 0 to 250 and an x-axis from 0 to 600. The data is represented by a series of sharp red peaks. A label 'f' is present in the bottom right of the plot area. Below the plot, there are four sliders with their corresponding values: Exposure (0.00 ms), Gain (0.01 X), Threshold Pixel (9.77 %), and Threshold Cell (18.35 %).

| Parameter | Value | Unit |
|-----------------|-------|------|
| Exposure | 0.00 | ms |
| Gain | 0.01 | X |
| Threshold Pixel | 9.77 | % |
| Threshold Cell | 18.35 | % |

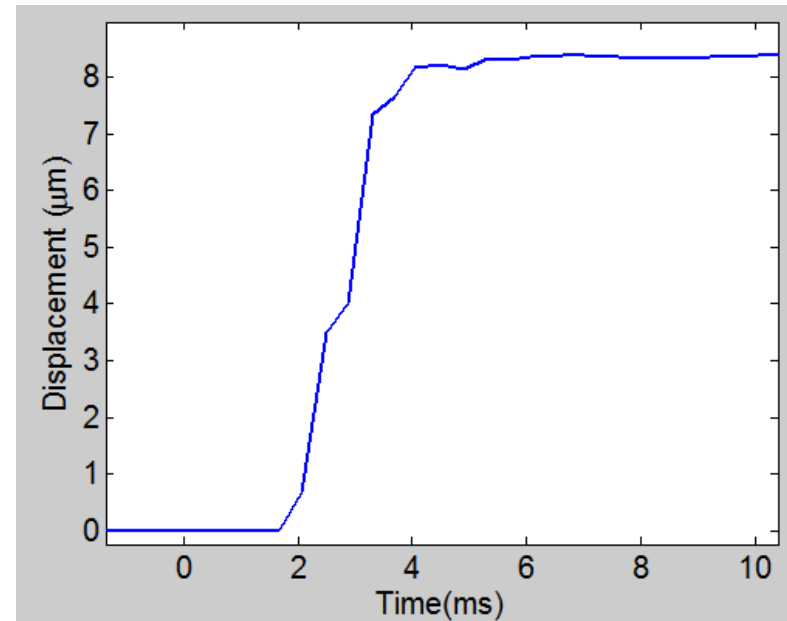
PhotonLoop – User Interface



PhotonLoop – Logging

Logging

- Flexible log engine to record **time series** of the internal **data**
- Can be saved to disk as **contiguous** or **segmented sessions**

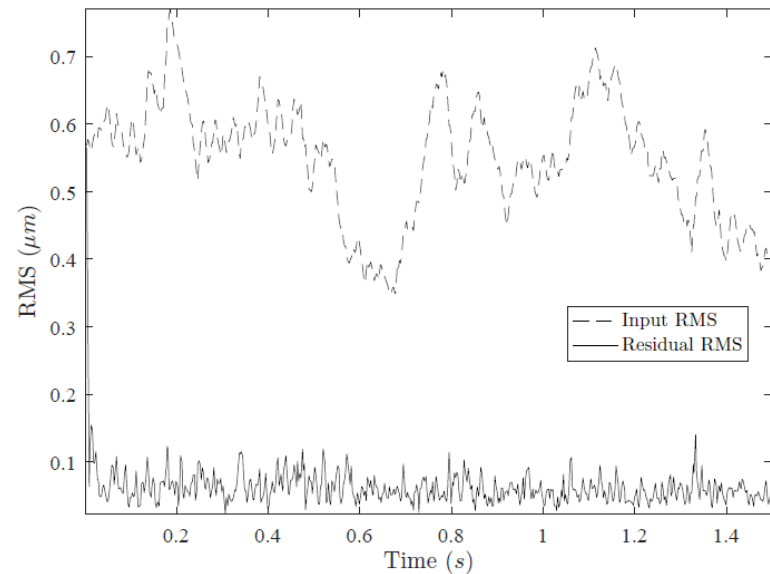


Actuator Rise Time @ 2.5kHz

PhotonLoop – Playback

Playback

- Recorded time series can be **given as input** for testing purposes
- Especially useful when **testing correction performance** amongst devices with **fair comparison**



Correction of a pre-recorded Zernike time series

PhotonLoop – Scripting

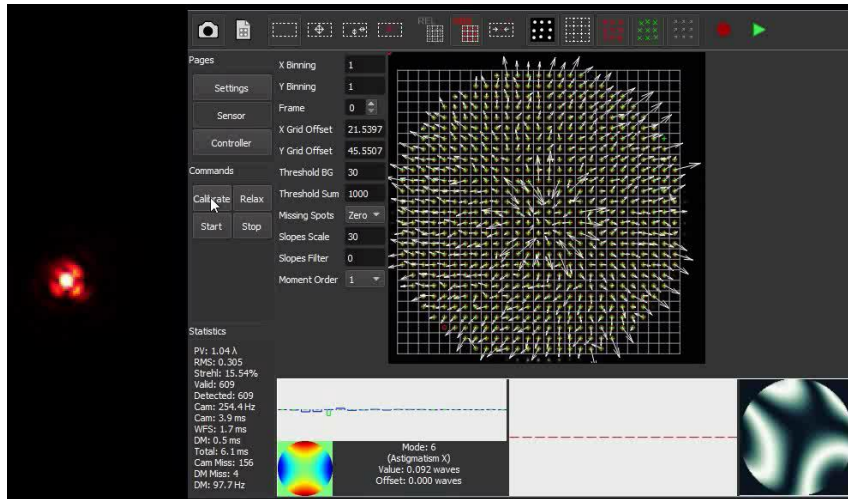
Scripting

- PhotonLoop can be **commanded** by **Javascript files**
- Can connect to **Matlab** or other softwares by **TCP-IP communication**

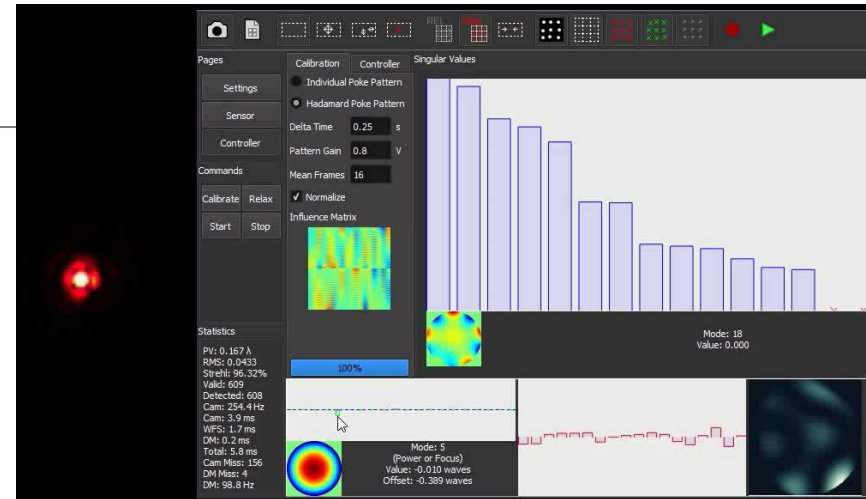
```
10 title = "Udiny";
11 description = "1hr long experiment";
12
13 logIterations = 3;
14 logTime = 1;
15 logPause = 2;
16
17 sc.initLogSession();
18 sc.initLog(title, description);
19
20 for (i=0; i<logIterations; i++)
21 {
22     →sc.playSound("ping")
23     →sc.startLog();
24     →sc.pause(logTime);
25
26     →sc.playSound("okay");
27     →sc.pauseLog();
28     →sc.pause(logPause);
29 }
30
31 sc.stopLog();
```

Sample Javascript File

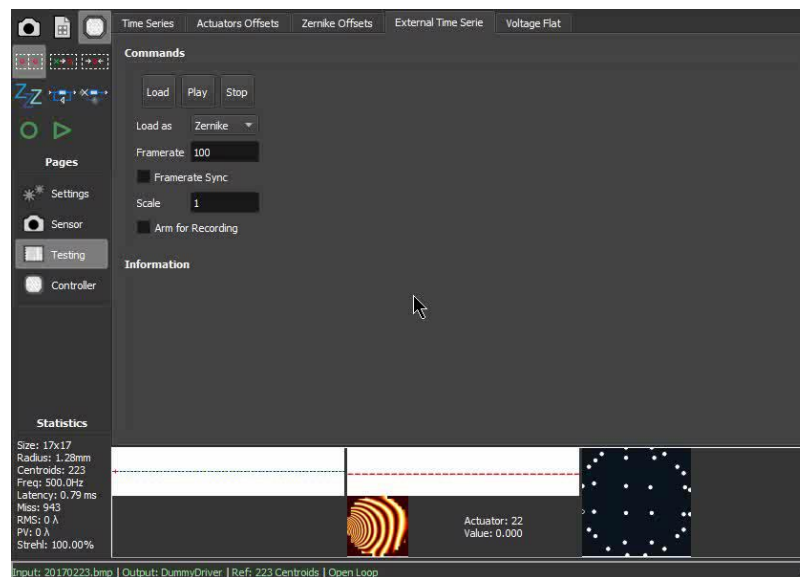
PhotonLoop – Demonstration



Hadamard Calibration



Zernike Generation



Aberration Playback and Correction

Conclusion

We have presented an **AO software controller** which is

- **Fast:** limited by WFS FPS; **500Hz; 2 frames** closed-loop latency
- **Flexible:** record, playback and script engines
- **Friendly:** responsive GUI with plenty of **calibration tools**

We tested it in **real world optical setups**

- Atmospheric turbulence eval (MBDA, La Spezia)
- High power laser (Udyni, Politecnico di Milano)
- Deformable Mirror/Lens comparison framework (CNR-IFN, Padova)