Type Research Paper
Section: Optics in Education

A Flexible AO system for research and educational purposes

Sistema flexible de óptica adaptativa para investigación y docencia

Boris Ayancán [*, 1], Héctor González-Núñez [1], Andrés Guesalaga [1,2]

1. Department of Electrical Engineering, Pontificia Universidad Católica de Chile, Santiago, Chile.

2. Astro-Engineering Center, Pontificia Universidad Católica de Chile, Santiago, Chile.

[*] E-mail: boris.ayancan@live.cl

Received:01/12/2016 Accepted: 21/11/2017
DOI: 10.7149/OPA.50.4.49024

ABSTRACT:

This paper shows how to implement a classic adaptive optics system. It describes the hardware elements and interfaces, it is proposed a software architecture and its implementation with C++ programming language. This AO system is fully implemented using free and open source development tools.

Key words: Adaptive optics; Qt; C++

RESUMEN:

En este artículo se muestra como implementar un sistema clásico de óptica adaptativa. Se describen los componentes e interfaces de hardware, se propone una arquitectura de software y su implementación en lenguaje de programación C++. Este sistema de óptica adaptativa está implementado usando herramientas de desarrollo gratuitas y de código abierto.

Palabras clave: Óptica adaptativa; Qt; C++

REFERENCES AND LINKS / REFERENCIAS Y ENLACES

[1] R. K. Tyson. *Introduction to Adaptive Optics. Washington,* SPIE press (2000).

[2] C. Béchet, A. Guesalaga, B. Neichel, V. Fesquet, H. González-Núñez, S. Zúñiga, P. Escarate, D. Guzman, "Beam shaping for laser-based adaptive optics in astronomy, " Opt. Express 22, 12994-13013 (2014).

[3] Qt Company, https://www.qt.io/

[4] Z. Xianyi, W. Qian, W. Saar, "OpenBLAS", http://www.openblas.net/

[5] Armadillo C++ Linear Algebra Library, http://arma.sourceforge.net/

[6] P-Y , Madec. "Overview of deformable mirror technologies for adaptive optics and astronomy." Proc. SPIE 8447, (2012).

[7] T. Bifano, "Adaptive imaging: MEMS deformable mirrors," Nature photonics 5, 21-23 (2011).

1. Introduction

Adaptive optics (AO) systems are one of the main components of the modern telescopes. These systems allow us to correct atmospheric disturbances in real time, greatly increasing the resolution of high performance telescopes [1]. Besides astronomy, these systems have multiple uses, including medical and military. A basic implementation requires a wavefront sensor (WFS) to measure disturbances, a real time controller (RTC) to calculate corrections and a deformable mirror (DM) to apply these corrections. A Steering mirror is also used to correct the low aberration modes (tip-tilt), helping to increase the dynamic range of the entire system. For real time operation, these components are connected in a closed loop.

In this work, we present the design of the control software, the hardware components selected, and the experimental setup used to validate results. The main characteristics of this implementation are flexibility and open source code. Therefore, other configurations are easy to implement with different optoelectronic devices. As an example, we plan to implement multi-conjugate AO configurations to correct different perturbation layers located at different planes (using more than one WFS and several DMs). Also beam shaping configurations can be implemented in order to generate arbitrary spatial light distributions [2].

The code is written in C++ language, using the free environment Qt [3]. Also, OpenBlas [4] and Armadillo [5] are libraries used to solve linear algebra operations. There are two configurations tested in laboratory, namely:

- The "Classic AO Loop", with one WFS and one DM.
- One WFS, one fast steering mirror to correct tip and tilt, and one DM to correct only the higher aberration modes of the beam.

2. Background

2.a. Adaptive optics overview

Our atmosphere is composed of masses of air which are continuously changing in temperature and position according to the wind direction. Due to its motion vortices are generated, which in turn generate additional vortices increasingly smaller. These phenomena cause that the refractive index in the atmosphere to be different for each position and time.

When the light travels through the atmosphere, it is slightly distorted because the refractive index is changing. This distortion is particularly problematic when we need to observe stars or when we need to observe details from a distant source, imposing a limit to what we can observe.

The adaptive optics systems are one of the alternatives that we have to counteract the limitations produced by the atmosphere. A basic AO system is composed of one WFS to measure the distortion of incoming light, a DM to apply corrections and a RTC to calculate the commands for the DM. Fig. 1 shows a classic AO setup for a telescope.

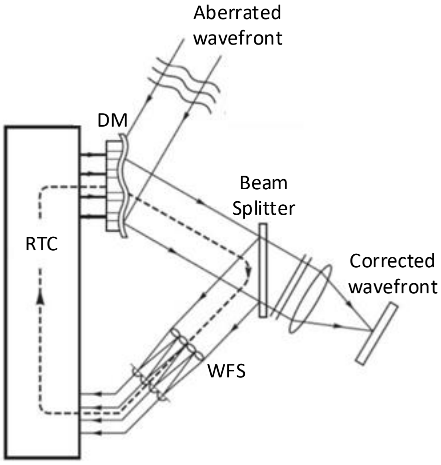


Fig. 1. Schema of a classic AO setup.

2.b. Optoelectronic devices

Below, we present the main characteristic of the optoelectronic devices selected for the experimental implementation.

- Wavefront sensor (WFS)

There are some alternatives to measure the wavefront of the light. Between them we have curvature sensors, pyramid type and Shack-Hartmann. For this system we use a WFS "Thorlabs WFS-150", which is a Shack-Hartmann sensor with 39x31 sub-apertures, pitch of 150μm, CCD sensor and range of wavelength between 300 to 1100 nm.

The Shack-Hartmann sensor uses a lens array to focus the incoming light on an image sensor. The sensor is calibrated with a known planar wavefron. In this case the focus of each lens is located in a position at the image sensor and all these positions are saved as X-Y coordinates of calibration XY_{cal}. When a distorted wavefront of light reaches the array of lenses, each lens focuses on a XY_{rx} position, which is different than XY_{cal}. The differences between XY_{cal} and XY_{rx} give the slopes of the wavefront being tested. Figure 2 illustrates the principle of operation of a Shack-Hartmann sensor.

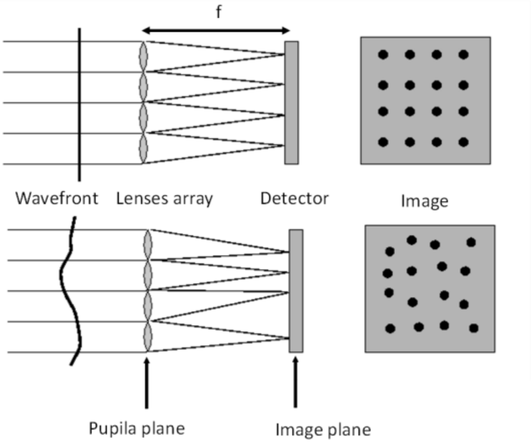


Fig. 2. Schema of a Shack-Hartmann sensor.

- Deformable mirrors (DMs)

These mirrors are able to change their surfaces using actuators behind of a reflective coating. To build a DM it is possible to use continued or segmented surfaces, and for the actuators the alternatives can be piezoelectric materials, magnetic actuators or even ferromagnetic nanoparticles suspended in a liquid carrier. Since the 90s a new technology based in MEMS (Microelectromechanical systems) has been made available, which is very compact and significantly reduces cost [6,7].

For this system we use a "Boston Micro Machines Multi-DM", which is a DM with 140 actuators, stroke of 3.5µm, based in MEMS and with a total active area of 4.4 x 4.4 mm.

- Phase screen

Phase screens or phase plates are optical components to provide phase shifts to a light beam. Introducing phase shifts is essential to test and validate adaptive optics systems in a controlled environment.

The shift is generated by a transparent material with a refractive index ηps. This material is shaped by a surface to give different optical path length to the beam.

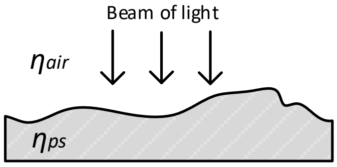


Fig. 3. Phase screen outline.

For this system we use the "Lexitek Phase Plate LS-100" which is a circular plate of 100 mm of diameter, with a phase array of 4096x4096 on its surface, r0=0.4 mm emulating a Kolmogorov turbulence statistics, and it is rotated by a step-by-step motor capable of reaching speeds between 0.15 to 10 rpm.

- Fast steering mirror

This is a controlled mirror that may be oriented at rates over 1 KHz. Atmospheric turbulences produce big phase changes on the wavefront, with the strongest being the tip-tilt component. This component moves the focus on the focal plane of the image sensor. For large exposition times the image on the scientific sensor spreads out, causing a poor Strehl ratio.

We can use a steering mirror to correct the tip-tilt component and let the DM to correct the high aberrations modes. Using this configuration it is possible increase the dynamic range of the AO system.

For this system we use a "Newport FSM-300 Fast Steering Mirror" wich has 25.4 mm of diameter, range of ±1.5°, resolution < 1µrad and a closed loop bandwidth ≥ 800Hz@10mV.

- Image sensor (CAM)

We use a "Thorlabs DCC1545M" as imager. This sensor is a CMOS camera, monochrome, 1280x1024 pixels and with USB connectivity.

2.c. Converting wavefront slopes from the WFS into commands of DM

To do this operation, it is necessary to determine the amount of aberration on the incoming beam of light in terms of its phase, and then apply the inverse phase on the DM to achieve a planar wavefront. On the other hand, most of practical situations may not require the calculation of the phase and instead, a projection from the WFS signals to DM commands is performed.

Consider the one-dimensional vector "y" with M signals from the WFS and another one-dimensional vector "a" with N actuators of the DM. The interaction between the actuators commands, the modified wavefront and the signals on the WFS can be connected in a rectangular matrix B of dimensions MxN, matrix B is also called "interaction matrix". Now the relation between actuators commands and WFS signals is [1]:

$$y = B \cdot a \tag{1}$$

In practical terms:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix};\ a = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix};\ B = \begin{bmatrix} B_{11} & B_{12} \dots & B_{1N} \\ B_{21} & B_{22} \dots & B_{2N} \\ \vdots & \vdots \ \dots & \vdots \\ B_{M1} & B_{M2} \dots & B_{MN} \end{bmatrix} \tag{2}$$

During the calibration procedure each actuator is activated individually and the signal response is added to the matrix B. For example, consider that we activate the actuator number 1, then the M signals captured from the wavefront are placed at the column 1 on the matrix B. In Shack-Hartmann sensors deviation in dx-dy are provided for every sub-aperture (every lens), so if the WFS has S sub-apertures, then the columns 1 is composed by [dx1 dy1 dx2 dy2 .. dxS dyS]T, then M= 2 · S. Column 2 is the response to actuator number 2, and so on.

When calibration is complete, it is possible to calculate WFS signals to command actuators using (1), then:

$$a = B^{-1} \cdot y \tag{3}$$

Where B^{-1} is the pseudo-inverse of B and it can be calculated using linear algebra algorithms. Truncated least squares techniques are used in this case. The matrix B^{-1} is also called "control matrix".

2.d. Software structure

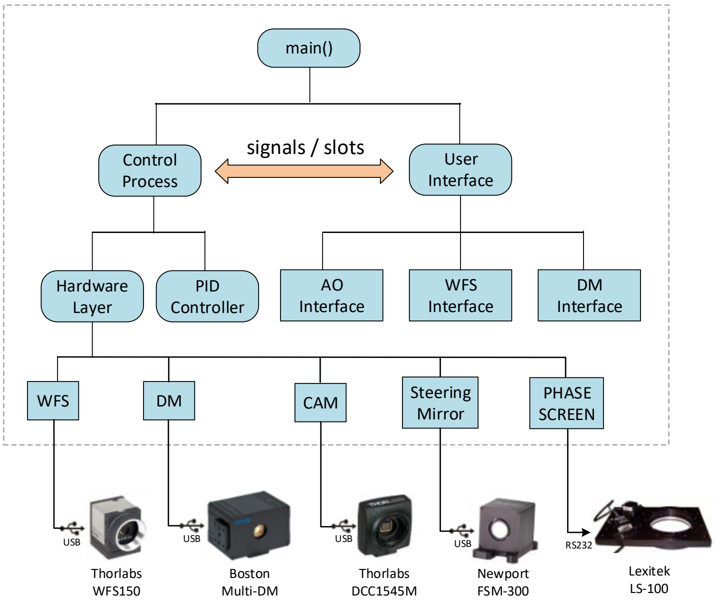


Fig. 4. Software structure.

Software has been built using two parallel processes in order to give flexibility to the system; these are called "Control process" and "User Interface". The control process mainly works with generic C/C++

functions and classes; for example, it could be run with a graphic or a console interface. The user interface works with the specific framework selected. Figure 4 shows the general structure of the software.

The Control Process module executes three modes: i) closed loop, ii) single step and iii) calibration modes. It generates signals to notify events like new data, change of status of hardware or any fault conditions. A PID controller is used for the closed loop. The Hardware Layer module provides a generic interface to hardware and also an abstraction to the specific components used, communicating with each device using its specific libraries. The libraries for the hardware devices are provided by the manufacturer.

User interface uses a window system to control the AO, it shows status and data received from each sensor and also presents complementary graphic information.

The signals/slots communication interface is specific for the Qt environment; this can be replaced by a generic event system or a callback mechanism when others environments are used.

3. Implementation of the software

3.a. Linear algebra operations

Handling large matrices requires optimized processing in order to achieve a real time operation. To speed up calculations we use special libraries called BLAS (Basic Linear Algebra Subprograms). BLAS libraries are executed at a low level according to the particular processor. For Intel processors a commercial framework called MKL (Math Kernel Library) is available. This framework includes high performance BLAS libraries specifically developed by the manufacturer.

There are several alternatives for BLAS libraries like OpenBLAS, OpenCV, GSL, ATLAS, among others. These libraries provide C-style interfaces. An alternative to get friendlier interface is a wrapper called Armadillo. Armadillo converts C-style interfaces from BLAS libraries to a format similar to MATLAB, this allows us to implement many MATLAB algorithm in C language quickly [5].

A test for two BLAS alternatives is shown on table 1. This test was performed for a pseudo-inverse calculation of a matrix size of 1,000,000 rows and 16 columns. Time elapsed for each configuration is shown as the result.

TABLE 1. Test for Linear Algebra libraries

Configuration	Time [sec]
Armadillo+ATLAS	1.9
Armadillo+OpenBLAS	0.9

According to the best performance on an Intel I5 processor we use OpenBLAS and Armadillo.

3.b. Hardware access

The standard way to access to a particular hardware is through libraries provided by the manufacturer. In some cases, the specification of the manufacturer follows standard protocols, for example RS-232, USB CDC or HID, TCP/IP, etc. In cases where the hardware uses standard protocols we can use the libraries provided by the operative system.

When we are developing software to access to hardware we need i) the specification functions that usually is a header file (.h), ii) the function implementation in libraries (.lib, .dll, .so, etc) and iii) the documentation necessary. If the hardware does not follow a defined standard, all these components must be provided by the manufacturer.

3.c. Calibration procedure

Calibration is performed to determine i) the interaction of each actuator of the DM on the reflected beam of light, ii) the number of actuators covered by the beam, and iii) the area of the WFS effectively illuminated. To get the calibration we excite each actuator to 50% of its total stroke. Then, we use the WFS to evaluate the influence of this actuator on the slopes. If there is a significant influence, the excited actuator is marked as active and its corresponding slopes are saved. Hence, we get a vector of slopes for each individual actuator which corresponds to a set of spots focused by the sub-apertures. All sub-apertures not generating a clear spot on the detector are discarded.

With the active actuators of the DM and the active sub-apertures of the WFS it is possible to build the interaction matrix B from (2) according to section 2.c.

3.d. Closed Loop Control

In closed loop mode the RTC is continuously measuring the output of the WFS. These measurements are separated in high and low modes, where each mode is processed by two different control loops. The high modes are corrected by the DM and the low ones by the Steering mirror. Each loop uses a "leaky" integrator in order to generate the control signals. The complete controller is implemented as a module of the "Control Process" described in section 2.d. Figure 5 shows a block diagram for the control loop.

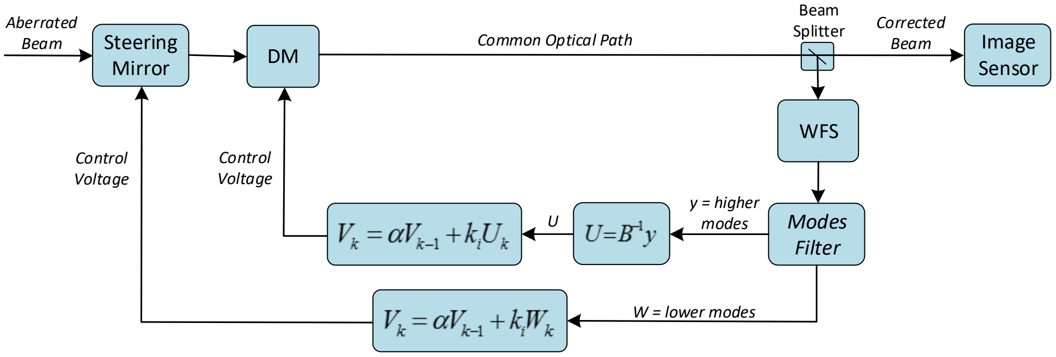


Fig. 5. Closed Loop Control.

3.e. Implementation with free Qt development tools

Qt is a popular framework to develop software in C/C++ language, it has bindings for several other languages like Python, Java, PHP, among others. Their main characteristics are multiplatform (Linux, Windows, MacOS), free, simplicity and robustness [3].

To start the implementation, the first step is to test every hardware interface separately. Table 2 shows the libraries provided by the manufacturer for Microsoft Windows users.

TABLE 2. Libraries provided by the manufacturers

Hardware	Manufacturer	Header	Library	Observations
WFS	Thorlabs	wfs.h visa.h	WFS_32.lib visa.lib	NI visa compatible
DM	Boston MM	-	CIUsbLib.tlb	Reimplemented as dm.lib
CAM	Thorlabs	uc480.h	uc480.lib	-
Steering Mirror	Newport	cbw.h	cbw32.lib	Controlled by Measurement Computing DAC USB-1208FS
Phase Screen	Lexitek	-	-	Serial RS232

Libraries .lib are directly linked by Qt, they have C-standard functions interfaces. Library for the DM is a little more complex. It has a Microsoft standard COM object (Component Object Model) as interface. COM objects provided more flexibility than standard interfaces because they support notifications and allow implementation in C++ as a regular object, but these objects were developed for Microsoft Tools and they have some compatibility problems with others frameworks.

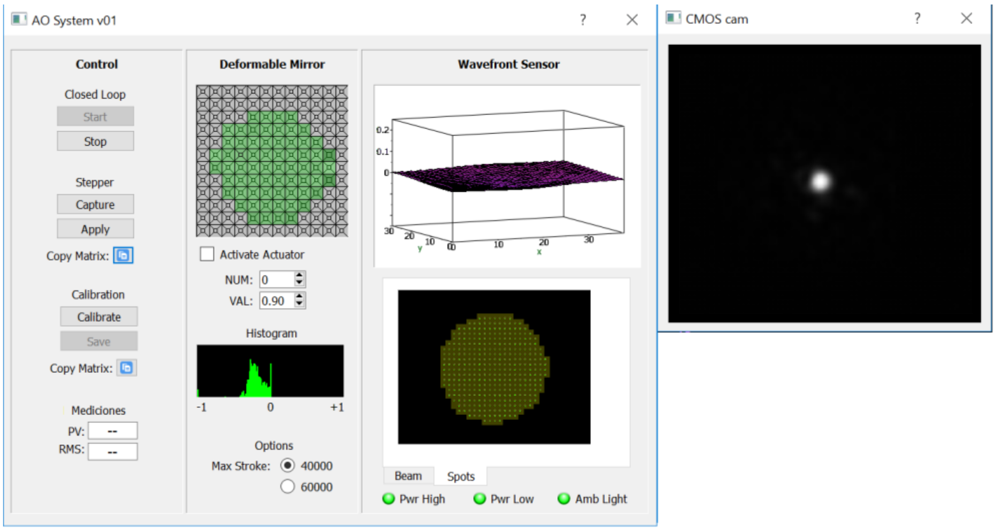
For the DM we use Microsoft Visual Studio Express (free edition) to reimplement "CIUsbLib.tlb" to a C-standard library in order to give compatibility with Qt. This library implements COM interface internally and it has just 3 access functions: open, write and close.

The phase screen is controlled with a serial RS-232 interface which is handled with QtSerialPort class, this class is common for the 3 operating systems supported by Qt.

When we have tested all hardware modules separately, we start to code the architecture of section 2.d.

4. User Interface

The interface has four different subsections to control and visualize the operation of the AO system. These subsections are called: "Control", "Deformable Mirror", "Wavefront Sensor" and "CMOS cam". All subsections are shown in figure 6.



PV: --
RMS: --

□ Activate Actuator
NUM: 0
VAL: 0.90

Max Stroke: ● 40000 ○ 60000

Fig. 6. User interface.

4.a. Control subsection.

The main section called "Control" allows selecting between the modes of the AO system. The implemented modes are: "Closed Loop", "Stepper" and "Calibration". To execute the first two modes the system must be calibrated properly.

Closed loop mode performs a continuous control on the DM in order to keep the phase measured on the WFS so close to zero as possible. Stepper mode tries to bring the phase measured on the WFS to zero in just one step. The button "Capture" performs a measurement with the WFS and the "Apply" button sends the calculated command to the DM.

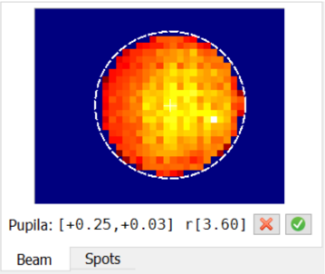
The modes "Stepper" and "Calibration" have buttons called "Copy Matrix" that export the matrix result of each operation to the clipboard of the operative system as tabulated text.

4.b. Deformable mirror subsection

On top, an actuators diagram shows in colour scale the level of signal applied to each actuator. The green actuators are those that are active and generate influence on the light. All actuators are commanded with levels from -1 to +1 and the hardware layer translates these levels into real voltage. Dark green is close to -1 and light green is close to +1. Saturated actuators are marked as red.

Actuators can be commanded individually with the checkbox "Activate Actuator" and the edits "Num" and "Val". A histogram of values of the actuators is shown to allow the evaluation of the status of the complete DM. The stroke of each actuator can be adjusted with "Max Stroke" option, according with the manufacturer specification.

4.c. Wavefront subsection



Pupila: [+0.25,+0.03] r[3.60]

Fig. 7. Pupil definition and beam for the WFS.

On top, a 3D surface shows the reconstructed wavefront from the spots deviation measured. This view shows aberration in real time, it can be rotated and zoom in on to select the desired point of view. The bottom graphic shows the spots diagram captured from the WFS, each point is located at the actual spot position and the area coloured around it shows that the spot is active and influenced by an actuator on the DM. The diagram captured from the WFS also shows the beam intensity in a colour map and in the same view it is possible to specify the size of the pupil. Figure 7 shows the beam on the WFS and the pupil size selected.

4.d. CMOS cam subsection

Here is shown the real time image from the camera. In this case the configurations of the camera are fixed and set to minimal exposure time. Using the mouse, it is possible pan and zoom the image.

5. Experimental results

In this section we present the tests performed, the setups used and the experimental results obtained. The maximum speed achieved for closed loop was 8 Hz, mainly limited by the speed of the WFS.

5.a. The "Classic AO Loop" implementation

Figure 8 shows the implemented test bench for the classic AO system. Intensity of the laser source is controlled by a polarizer, next a pinhole and a lens produces a planar wavefront. The phase screen introduces aberrations on the planar wave front and then it reaches a diaphragm which is the pupil of the system. The pupil is transported to the DM through a 4F lens system, next the DM is translated to de WFS using another 4F lens system. At the end, a beam splitter directs part of the beam into a scientific camera with a lens that produces an image to analyze the results. Table 3 shows the components and it respectively functions.

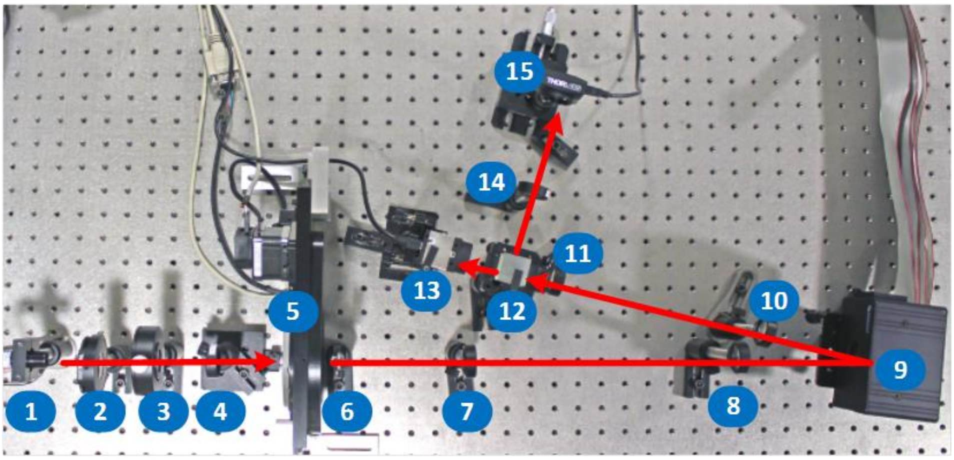


Fig. 8. Implemented test bench.

TABLE 3. Components of "Classic AO Loop" test bench

Num.	Component	Function
1	Laser	Light source of 670nm
2	Polarizer	Variable attenuator
3	Pinhole	Disperser
4	Lens	Collimating
5	Phase Screen	Simulate aberration
6	Diaphragm	Pupil
7-8	Lens Array	4F
9	DM	Phase correction
10-11	Lens Array	4F
12	Beam splitter	Light to WFS and Image sensor
13	WFS	Wavefront sensor
14	Lens	Focus
15	Image Sensor	Analyse results

Two test were performed, in the first one i) the phase screen remains static when the closed loop is turn on, and in the second one ii) the phase screen is continuously rotating at 0.15 rpm while the closed loop at 8 Hz tries to keep corrected the wavefront.

Figure 9 shows a sequence of control performed in closed loop mode by the AO system. This sequence was taken with consecutively frames every 240 ms and while the phase screen is static.

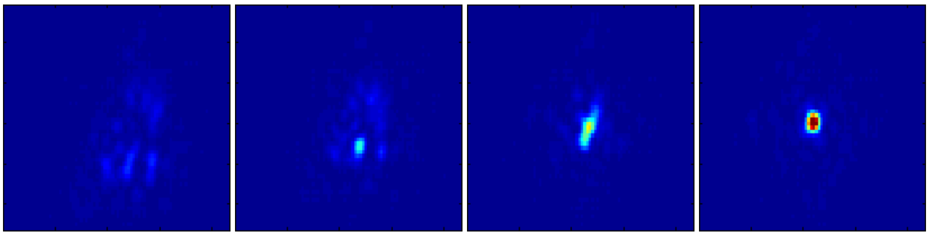


Fig. 9. Control sequence with the phase screen static.

Figure 10 and 11 shows the relative intensities after an integration of 30 frames, it was taken while the phase screen is rotating at 0.15 rpm and the closed loop runs at 8 Hz. At the left of each image there is a reference with the AO control loop off.

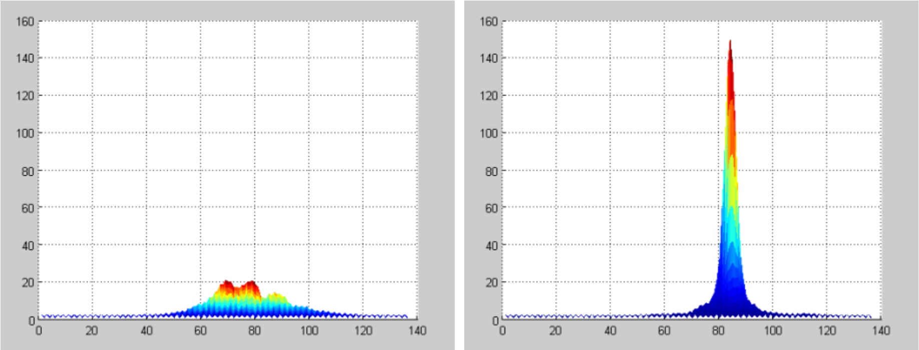


Fig. 10. Profile from relative intensities after an integration of 30 frames. At the left the AO loop is off and at the right it is on. The units of x-axis are in pixels (1px = 5.6 μm).

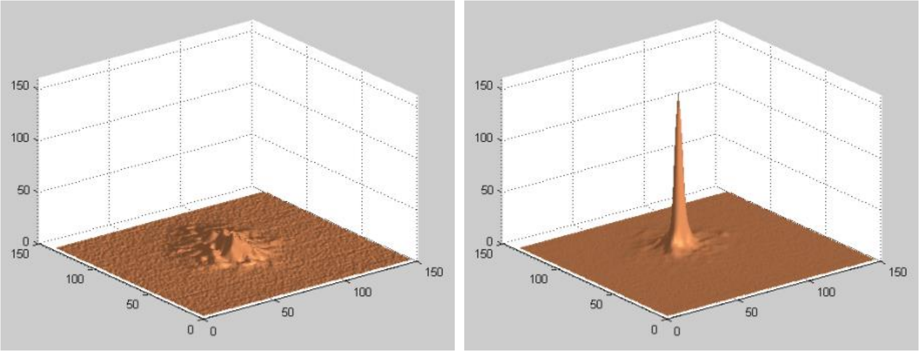


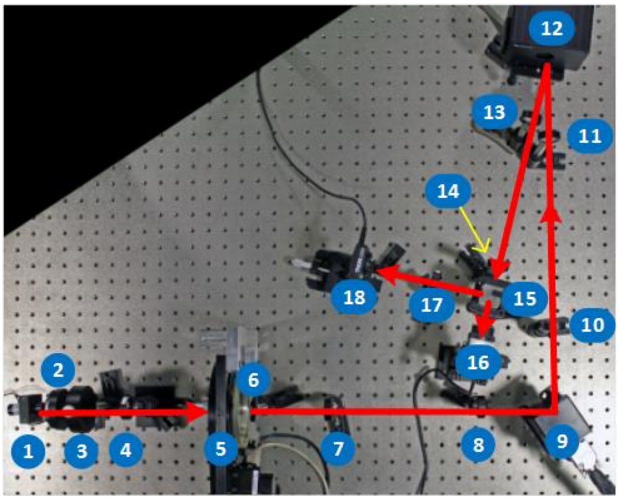
Fig. 11. Surface plots for relative intensities after an integration of 30 frames. At the left the AO loop is off and at the right it is on. The units of x-y-axis are in pixels (1px = 5.6 μm).

5.b. High modes correction with one DM and the low modes with a Steering mirror

This setup is similar to the classic configuration described in 5.a, but in this case the aberrations of the input beam are separated in lower and higher aberration modes according to Zernique decomposition. The low aberration modes contain the tip-tilt component and these are corrected by the steering mirror, the high aberration modes (defocus and uppers) are corrected by the DM. In this configuration the dynamic range of the AO system is strongly increased (compared with 5.a). The control loop for this configuration is shown in figure 5 and the assembled setup in figure 12.

Same as 5.a, two test were performed: i) the AO loop is turn on while the phase screen is static, and in the second one ii) the AO loop is running at 8 Hz while the phase screen rotates at 0.15 rpm.

Figure 13 shows a control sequence when the AO loop is turned on with the phase screen static. These frames were taken every 240 ms.



Num.	Component	Function
1	Laser	Source of 670nm
2	Polarizer	Variable attenuator
3	Pinhole	Disperser
4	Lens	Collimating
5	Phase Screen	Simulate aberration
6	Diaphragm	Pupil
7-8	Lens Array	4F
9	Steering Mirror	Tip-Tilt correction
10-11	Lens Array	4F
12	DM	Phase correction
13-14	Lens Array	4F
15	Beam splitter	Sample to WFS
16	WFS	Wavefront sensor
17	Lens	Focus

Fig. 12. Implemented test bench and its components.

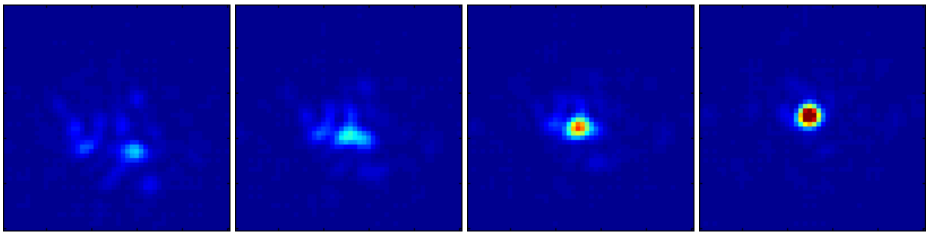


Fig. 13. Control sequence with the phase screen static.

Figure 14 and 15 shows the relative intensities after an integration of 30 frames, they were taken while the phase screen is rotating at 0.15 rpm and the closed loop running at 8 Hz. At the left of each image there is a reference with the AO control loop off.

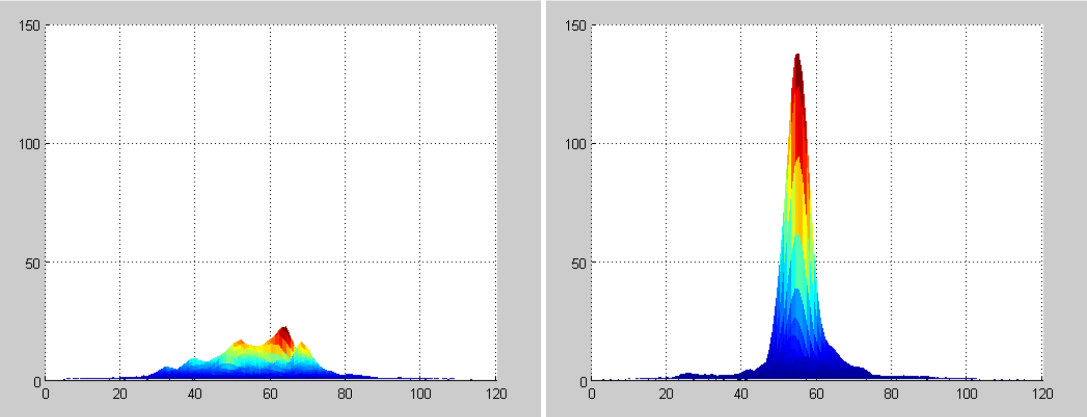


Fig. 14. Profile from relative intensities after an integration of 30 frames. At the left the AO loop is off and at the right it is on. The units of x-axis are in pixels (1px = 5.6 µm).

This test shows similar results as in 5.a, this is because the limited range of aberrations induced by the phase screen.

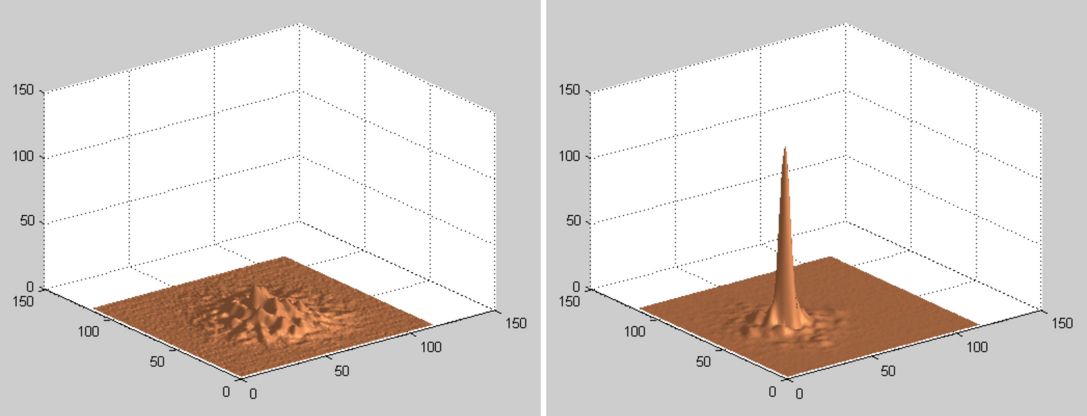


Fig. 15. Surface plots from relative intensities after an integration of 30 frames. At the left the AO loop is off and at the right it is on. The units of x-y-axis are in pixels (1px = 5.6 µm).

6. Conclusions

This article describes a complete laboratory AO system using only open source development tools. The maximum speed achieved for closed loop was 8 Hz, where most of this limitation is caused by the WFS frame-rate. The performance of the system met the expectation for research and educational purposes.

The tests performed in 5.a and 5.b show the flexibility of the modular architecture, which in this case, it allows to change easily the configuration when a steering mirror is added to the classic setup. In the same way, it is possible to test others configurations with more than one DM, WFS or steering mirror. Besides, other AO control algorithms or wavefront manipulation techniques could be easily implemented and tested.

An approach to improve the performance on this AO system could be the use of a parallel processing configuration with multiples GPUs. This configuration can be implemented using hardware like NVIDIA and its platform CUDA. To benefits from parallel processing it is suggested to read the raw image from the WFS and delegates the slopes calculations to the GPUs, instead of the one thread calculation on the CPU that is currently performed. This alternative may be considered as the next step to improve this flexible AO system.

There are several AO kits available on the market, for example we have the AO kit from Thorlabs, which has similar specifications compared with the proposed system. It includes a Shack-Hartmann WFS with 31x39 sub-apertures, a DM with 12x12 actuators and it can reach a speed of 15 Hz. A more sophisticated version is available with the ALPAO manufacturer; this can work at speeds above 800 Hz. Although these two examples have excellent user interfaces, they don't have available the source code for the controllers.

Acknowledgements

Héctor González-Núñez acknowledges the support of FONDECYT (2015 Postdoctoral Grant 3150343). Andrés Guesalaga thanks CONICYT, grant Anillo ACT-1417.