Mon. Not. R. Astron. Soc. **356,** 1263–1275 (2005) doi:10.1111/j.1365-2966.2004.08524.x

Modelling astronomical adaptive optics – I. The software package CAOS

M. Carbillet,[1⋆] C. Vérinaud,[2] B. Femenía,[3] A. Riccardi[4] and L. Fini[4]

[1]*Laboratoire Universitaire Astrophysique de Nice–UMR 6525, Parc Valrose, 06108 Nice Cedex 02, France*
[2]*European Southern Observatory, Karl-Schwarzschild-str. 2, 85748 Garching-bei-München, Germany*
[3]*GTC Project, Instituto de Astrofísica de Canarias, C/ vía Láctea s/n, 38200 La Laguna, Spain*
[4]*INAF–Osservatorio Astrofisico di Arcetri, Largo Enrico Fermi 5, 50125 Firenze, Italy*

Accepted 2004 October 18. Received 2004 October 15; in original form 2004 July 2

ABSTRACT

The software package CAOS (code for adaptive optics systems) described in this paper is a software ensemble of modules designed for end-to-end simulation of generic astronomical adaptive optics (AO) systems, including a complete atmosphere turbulence model, sodium laser-guide star upward and downward propagation, observed object definition, Shack–Hartmann and novel pyramid wave-front sensors detailed modelling, wave-front reconstruction and subsequent time-filtering tools, and wave-front correction via different kinds of correctors; but also image formation, Fizeau interferometry, coronagraphy, etc. Consequently, it is more likely to be used as a tool dedicated to detailed optical astronomy studies than as an instrument simulator, as it is based on a wide range of unprecedented physical modelling. After a brief but global description, with particular emphasis on the most interesting physical modelling features, its use is illustrated through a chosen application, namely the question concerning the opportunity of having a tip–tilt-dedicated sensor in a 8-m class telescope pyramid-based AO system.

Key words: instrumentation: adaptive optics – instrumentation: high angular resolution – methods: numerical – telescopes.

1 INTRODUCTION

Adaptive optics (AO) is now a mature technology widely used in optical astronomy, and is part of every new optical telescope project. A number of impressive results, for example, with the NAOS AO system on board one of the Very Large Telescopes (VLT) (see, e.g., Gendron et al. 2004; Lacombe et al. 2004; Lagrange et al. 2004) has already been obtained, and a number of novel AO concepts are being studied for cutting-edge applications, such as extrasolar planet searching (e.g within the framework of the VLT-Planet Finder) or wide-field high-angular-resolution astronomy [the European Southern Observatory, ESO, multiconjugate, MC, AO demonstrator MAD for VLT, the interferometric MCAO system LINC-NIRVANA for the Large Binocular Telescope (LBT), the Gemini laser-based system, etc.], pushing the limits of the original concept.

Within this framework, and due to the complexity of this kind of system, the performance analysis has a number of problems that are not open to analytical solution. In fact, this kind of study involves the complex nature of the turbulent atmosphere, and also the fact that a number of competitive instrumental concepts are often to be deeply analysed and compared (e.g. pyramid wave-front sensors versus Shack–Hartmann ones, piezo-stacked mirrors versus adaptive secondaries, different reconstruction strategies, etc.). Consequently, Monte Carlo-based numerical simulations are necessary, not always end-to-end but often extremely detailed at least for a particular physical process or piece of hardware behaviour to be studied. Moreover, studies concerning the coupling of AO systems with the subsequent instrumentation (interferometry, coronagraphy, spectroscopy, etc.), are also of fundamental importance.

Some numerical tools have already been presented, though not always as widely distributed and developed as CAOS, and often strictly limited to a given type of AO. Among them we can cite the following: PAOLA, an IDL-written semi-analytical code dedicated to extremely large-aperture AO (Jolissaint & Véran 2002); LOST, also IDL-based and dedicated to layer-oriented MCAO simulations (Arcidiacono et al. 2004); and the ESO parallel C codes, dedicated to extremely large-aperture MCAO (Le Louarn et al. 2004).

In this paper we present a generic numerical tool for AO and AO-related studies: the software package CAOS, which integrates state-of-the-art physical and numerical modelling developments and is already distributed among a large community of AO-concerned astronomers/researchers. As a matter of fact, it has to be noted that a number of studies using the tool in its successive versions have already been, or are currently being, performed. Among them we can cite the following: a first evaluation of the performance of the first-light AO system of LBT (Carbillet et al. 2003); studies concerning the impact of partial AO correction on the interferometric imaging capabilities of LBT (Carbillet et al. 2002a); optimization of the deformable mirrors conjugation altitude in MCAO (Femenía & Devaney 2003); the proved possibility of sensing the differential piston using a pyramid sensor (Vérinaud & Esposito 2002); and simulation studies concerning the ESO MCAO demonstrator MAD (Carbillet et al. 2002b; Vérinaud et al. 2003). Moreover, it

⋆E-mail: marcel.carbillet@unice.fr

can also be used to provide a precise estimate of the astrophysical performance, in terms of the signal-to-noise ratio and exposure times, and/or as a solid preparation for observations with existing AO-equipped instruments (see, for instance, Habart et al. 2004), and for the preparation of the near future observations with, for instance, the LBT first-light AO system.

Our tool also represents a first necessary building block for further scientific studies of fundamental importance to optical astronomy (and especially AO), such as the crucial problem of very-high dynamic range imaging (as planned for the VLT-Planet Finder project, and more generally for exoplanets and protoplanetary discs observations). A number of definitive answers have to be found regarding the type of AO system that has to be used, the observing conditions (magnitude, wavelength, atmospheric turbulence, etc.) and the coupling/propagation of error when coronagraphy is contemplated.[1] The promise of wide-field optical astronomy, using MCAO for which a number of concepts have been proposed, is also another important area for which our tool can contribute.

The software package CAOS is essentially a set of modules designed to be used within a graphical programming environment – the CAOS application builder (Fini, Carbillet & Riccardi 2001; Fini & Carbillet 2003), where the data flow between modules can be defined, and the parameters of each module can be set. The present version of the software package CAOS is 5.0, and it is the result of six years of work involving considerable effort in the modelling of a number of the physical processes involved, and in the numerical writing and distribution of the tool. The graphical part of our software 'system', the CAOS application builder, has now reached version 4.1, and also accommodates the (somewhat more recently developed) software package AIRY (Correia et al. 2002), dedicated to Fizeau interferometric image restoration studies. The software package CAOS has now reached a very high level of maturity, and it is time to present it to the whole astronomical community.

The scope of this paper is not just the presentation of the software package CAOS, but also a description of its most interesting and novel features, together with an illustration of its use through a chosen example. Therefore, the paper is organized as follows. In Section 2 we briefly describe the global structure and main features of our programming environment, while in Section 3 we highlight the most interesting features of the various modules that have been developed. In particular, we give some details concerning the detailed modelling of the novel pyramid wave-front sensor for which there is wide interest. Section 4 is then dedicated to a case study concerning the utility of a typical tip–tilt sensing device in addition to pyramid wave-front sensing, within the framework of an 8-m class telescope. Finally, our concluding remarks are given in Section 5.

2 THE CAOS SYSTEM AND ITS RELATED PROGRAMMING ENVIRONMENT

The structure under the CAOS 'system' is modular. This means that each elementary physical process of a given simulation is modelled within a specific module; for example, in the AO case, the turbulence in each atmospheric layer, the propagation of light from an object to the observing telescope and through the turbulent layers, the physical characterization of the object itself, the wave-front sensing, the wave-front reconstruction, the time-filtering of the resulting deformable mirror commands, the wave-front correction, etc. Taking advantage of the CAOS application builder, a simulation can be built by connecting together the required occurrences of the desired modules, respecting only the logical constraints given by their formalized input/output types. Complex simulation applications are thus simply created by assembling the elementary building blocks (representing the modules) in a straightforward manner, so that the user can concentrate on the scientific aspects of her/his problem, while mundane coding problems are managed by some automatic tools.

Each module comes with an individual graphical user interface (GUI) in order to set its own physical and numerical parameters, during the design step or independently at a later moment. In practice, each module is defined by a standard group of function calls, a collection of parameters, and a typed definition of input(s) and output(s). It can support up to two inputs and two outputs, and it is represented within the application builder as a rectangular box with coloured input handles (on its left-hand side), and output ones (on its right-hand side). Each colour describes one of the pre-defined types of input/output: wave-front, image, commands, etc. The software package CAOS also contains a library of utilities, and a detailed hypertext help which can be called from each individual module GUI, and a set of examples of typical simulation projects that can be used as a starting point for new applications.

After the simulation design step is completed, the block diagram is analysed by the application builder and the IDL code implementing the simulation program is generated. It can be modified 'by hand' in order to complete some additional task not strictly provided using the existing modules. The whole structure of the simulation can be saved as a project that can be restored for later modifications and/or parameter upgrading. Beside the main simulation project one or more calibration project(s) might be previously designed and run.

Fig. 1 shows the CAOS application builder (background), together with the automatically generated code (foreground): two IDL-written routines, one for the calls to the various modules required, and the other one for the general administration of the simulation project (graphical representation and global parameters). As already mentioned, this code is generated at the end of the design phase for the simulation, a phase during which each required module is picked up from the 'module' menu in which all the modules of all the installed software packages are present, and then put in one of the pre-defined boxes within the application builder itself. The occurrence of a module is hence represented by a box with the name of the module (e.g. the turbulent atmosphere module is represented by the box named ATM in the figure) and with pre-defined inputs and outputs. By clicking on the occurrence of a module (e.g. the occurrence of module ATM we were evoking before was number 001), a GUI is opened in order to chose the various physical and numerical parameters related to the module itself (and only it). Global parameters (common to all modules) are limited to a strict minimum: only the total number of iterations and the current iterate number, and only if necessary.

It is clear from Fig. 1 how additional code can easily be implemented directly within the two automatically generated routines. Moreover, new modules can also be implemented very easily thanks to the template module included with the CAOS application builder distribution. Consequently, it has to be noted that new software packages (dedicated to a given thematics not treated by the existing software packages) are also easy to build up. Also note that

[1] A very detailed physical modelling of the instrumental response and the subsequent exact morphology of the resulting AO-corrected point spread function is fundamental here and is only achieved through the complete physical analysis provided by an end-to-end simulation tool such as the software package CAOS.

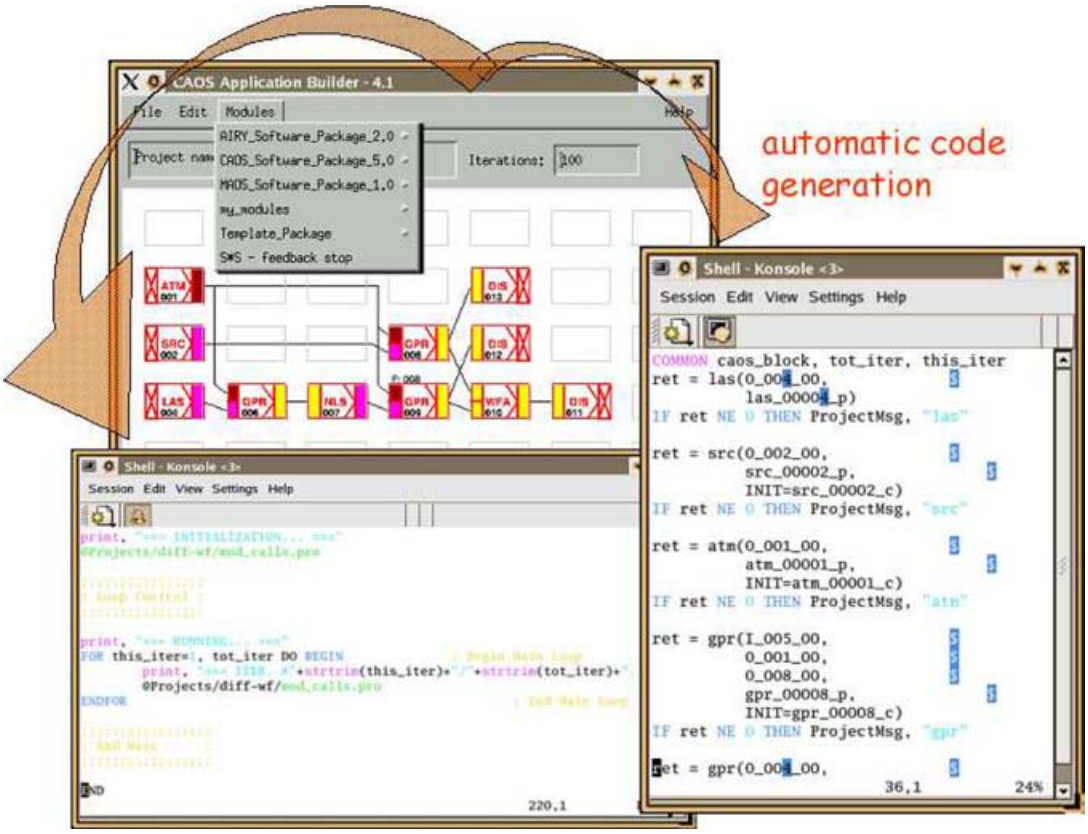

Figure 1. Background: the CAOS application builder showing a typical simulation design using the modules of the software package CAOS, and together with the list of software packages installed that appears when pushing the button 'modules'. Foreground: the two routines that are automatically generated at the end of the simulation design phase.

modules from different software packages can work together in a unique project, given the input/output structure type compatibility.

Everything, from the application builder to each of the software packages, is implemented in the IDL language, but efforts are being planned in order to port the whole code to the newly developed GDL language, which is supposed to reach complete equivalence to IDL version 6.0 soon (see http://sourceforge.net/projects/gnudatalanguage/).

3 DESCRIPTION OF THE MODULES OF THE SOFTWARE PACKAGE CAOS

In this section we describe the tasks of each module of the software package CAOS (version 5.0), going in detail for some of the features we believe to be most interesting. A complete and detailed description of each module can be found by directly looking at the hypertext help for each module given together with the package and downloadable from the CAOS web-page http://www.arcetri.astro.it/caos.

Table 1 shows a complete list, together with a very brief description, of the modules present in the modules library of the software package CAOS 5.0. Different classes of modules have been defined and details are given in the following subsections.

3.1 Wave-front perturbation

Module ATM (atmosphere building) generates the turbulent atmosphere that then produces the corrupted wave-fronts for the whole simulation through module GPR (geometrical propagation), which propagates the light from a given source/object modelled within module SRC (source definition) and through the modelled turbulent atmosphere.

Going into a bit more detail, we see that ATM is a module with no input except its parameters chosen via its GUI, and a unique output composed of a given number of turbulent layers together with their altitudes, wind velocities, etc. In fact, we assume that the turbulence is mainly located within a few relatively narrow layers – at least for good astronomical sites. The finite number of turbulent layers is a consequence of the modelling of the profile of the structure constant of the refraction index fluctuations, $C_N^2(h)$, as an ensemble of discrete values. Each of these values corresponds to a turbulent layer of the atmosphere, and each turbulent layer can be physically simulated as a random phase screen for which the power spectrum follows the von Kármán–Kolmogorov model.

The first step in building the turbulent atmosphere is to generate the phase screens that will simulate the behaviour of each turbulent layer. We have implemented, up to now, two methods: a fast Fourier transform- (FFT-) based method, with low-spatial-frequency boosting (also known as 'subharmonics adding'), and a Zernike-polynomials-based method, using an alternative definition of the Zernike polynomials for high orders. We refer to Carbillet & Riccardi (in preparation) for a detailed discussion concerning these two methods.

Once the phase screens have been generated, the turbulent atmosphere, that is the output of module ATM, is built by arranging the required number of phase screens/turbulent layers with their altitudes and taking into account the C_n^2 profile chosen by the user.

Table 1. Descriptive list of the modules of the software package CAOS, version 5.0.

Module	Purpose
Wave-front perturbation and image formation	
ATM, atmosphere building	– to build the turbulent atmosphere (see also utility PSG, phase screen generation)
SRC, source definition	– to characterize the guide star/observed object
GPR, geometrical propagator	– to geometrically propagate the light through the turbulence
IMG, imaging device	– to make an image of the observed object
LGS-oriented modules	
LAS, laser characterization	– to define the laser characteristics
NLS, Na-layer spot definition	– to characterize the sodium-layer behaviour
Wave-front sensing	
PYR, pyramid wave-front sensor	– to simulate the pyramid sensor behaviour
SLO, slope computation	– to compute the slopes from the pyramid signals
SHW, Shack–Hartmann wave-front sensor	– to simulate the Shack–Hartmann sensor behaviour
BQC, barycentre/quad-cell centroiding	– to compute the slopes from the SH spots centroiding
Wave-front reconstruction and correction	
REC, wave-front reconstruction	– to reconstruct the wave-front
TFL, time-filtering	– to apply time-filtering during wave-front reconstruction
DMI, deformable mirror	– to correct from the wave-front perturbations
Tip–tilt-specialized modules	
TCE, tip–tilt centroiding	– to compute and reconstruct the tip–tilt
TTM, tip–tilt mirror	– to correct from the tip–tilt
Calibration-oriented modules	
CFB, calibration fibre characterization	– to define a fibre to be used for calibration purpose
CSQ, command sequencer	– to generate a sequence of commands
MCA, make calibration	– to make and save the calibration matrix
MDS, mirror deformation sequencer	– to generate a sequence of mirror deformations
SCD, save calibration data	– to save the calibration data
Other modelling modules	
IBC, interferometric beam combiner	– to combine the light from two pupils
COR, coronagraphic module	– to simulate various coronagraphic concepts
AIC, achromatic interfero-coronagraph	– to simulate the achromatic interfero-coronagraph
BSP, beam splitter	– to split the light beam
Additional utilities	
WFA, wave-front adding	– to add or combine together wave-fronts
ATA, atmosphere adding	– to add or combine together atmospheres
IMA, image adding	– to add or combine together images
STF, structure function calculator	– to compute the structure function from propagated wave-fronts
SAV, save structure	– to save any type of input/output structure (XDR format) (see also utility RST, restore structure)
DIS, generic display	– to display any type of pre-defined input/output

If temporal evolution is needed, this is performed once (the first time), ATM will then just shift each of the layers by an ad hoc quantity taking into account the base-time (minimum atmosphere/turbulence evolution time defined within this module but imposed on all the subsequent simulation branches) and their associated velocity vectors. If no temporal evolution is needed, no base-time and no velocity vectors are asked for and each time ATM is called it outputs a statistically independent ensemble of turbulent layers.

In the case of a natural object, its angular coordinates (off-axis, position angle), photometric and spectroscopic characteristics (magnitudes in the various Johnson bands from U to M, spectral type – assuming the blackbody approximation), and its morphology can be set within module SRC. In addition, background magnitudes are also set for further computation during the process of image formation and/or wave-front sensing. Objects with a given morphology can also be defined, by either choosing one of the existing options or by loading a previously computed model resulting in a two-dimensional map. A finite distance can also be set; in such a case it is assumed to be a laser guide star (LGS) at such a distance from the telescope pupil.

The propagation of light is performed by the module GPR, taking into account the positioning of each part (telescope, projector, possibly a sodium layer, turbulent layers) and the time evolution. The propagation is performed in a geometrical way, and the so-called pixel magnification is taken into account in the LGS case in order to simulate the cone effect.

A module permitting Fresnel propagation instead of a geometrical one, and hence capable of simulating the scintillation effects on the subsequent physical processes (e.g. wave-front sensing and/or very high-contrast imaging) is being written, but will be part of a forthcoming release.

3.1.1 The LGS case

When the use of a laser guide star is contemplated, it can be modelled at different levels of detail. A first level of detail is achieved by taking into account the finite distance of the LGS only (i.e. defining a point

source at a finite distance within the module SRC). A deeper analysis also takes into account the two-dimensional morphology of the LGS (also using module SRC). And a complete end-to-end analysis considers the whole process of propagation of the laser beam from a telescope projector (using module LAS), through the turbulence, and to the sodium layer (modelling the three-dimensional resulting spot using module NLS, which can integrate a physical-model-based or an on-site-measurement-based profile), and then propagating the light from the sodium layer to the observing telescope and through the turbulent atmosphere again. The example project given in Fig. 1 is also an illustration of the difference between the latter case (whole modelling of a sodium LGS with modules LAS and NLS) and either simple downward propagation from a natural guide star or a simpler modelling of the LGS (considering only downward propagation with or without a two-dimensional morphology for the LGS using only module SRC). Note that one occurrence of module GPR (number 006 in the figure) is for the upward propagation of the laser beam through the turbulence to the sodium layer (hence the associated parameters denote the position, size and shape of the projector telescope), while the other one (number 009 in the figure) is for downward propagation of it (hence the associated parameters concern the observing telescope and are the same as GPR number 008). As a result, the simulation project in Fig. 1 (thanks also to module WFA, which can perform a difference between two propagated wave-fronts) permits the study of all the effects on the LGS AO system, and in particular the cone effect and the tip–tilt indetermination problem.

3.2 Imaging process

Module IMG (imaging device) performs the simulation of the image formation process on a square array detector (e.g. a CCD or an infrared detector array). Module IMG requires no information concerning the actual optical layout of the camera as explicit use of the aberration-free thin-lens approximation is made and its diameter is assumed to be large enough not to become the element defining the pupil of the optical system. Consequently, the intensity pattern on the focal plane (i.e. on the detector array) is computed using monochromatic Fraunhofer diffraction theory (Goodman 1968) involving a FFT operation when evaluating the point spread function (PSF) of the optical system telescope+atmosphere, and two additional FFT operations in order to perform the convolution between the PSF and the object distribution when imaging extended two- or three-dimensional sources. Both the aberration phase function used to compute the PSF and the object brightness distribution are fed to module IMG via its single input. Three-dimensional LGS spots are discretized in a set of two-dimensional arrays (each of them referred to as sublayers) and module IMG proceeds as if each sublayer were an extended two-dimensional source, so that for each sublayer we compute a PSF aberrated by atmospheric turbulence plus the defocus aberration caused by the fact that the detector is conjugated at a given plane, while the source spans behind and beyond such a plane. Each layer PSF is convolved with its corresponding LGS slice giving a two-dimensional image per LGS slice. The final image is obtained by a weighted average of all the LGS slice images, where the LGS slice intensities have been used as weights.

The sampling of the images obtained with FFTs will depend on the size of the array storing the input aberrated wave-front; let us denote this resolution by $\Delta_{\rm FFT}$. In general $\Delta_{\rm FFT} \neq \Delta_{\rm CCD}$, where $\Delta_{\rm CCD}$ is the detector pixel size. To avoid interpolation routines, the original wave-front array is padded with zeros in such a way that when performing the FFTs the resolution that we arrive at is $\Delta_{\rm CCD} = n\,\Delta'_{\rm FFT}$, where n is an integer greater than or equal to 1. When dealing with two- and three-dimensional sources the object brightness functions are bilinearly interpolated to the same resolution $\Delta'_{\rm FFT}$ (here we are implicitly assuming that object brightness functions are very smooth compared with PSFs). After convolution the image at resolution $\Delta'_{\rm FFT}$ is easily rebinned to yield an image of resolution $\Delta_{\rm CCD}$. The resulting field stop of the imaging system is assumed to be the detector itself, so that the field of view (FoV) is obtained as the number of pixels times the angle subtended by each pixel.

The parameters set by the IMG GUI are the linear number of pixels in the detector, the detector pixel size $\Delta_{\rm CCD}$, the distance from the entrance pupil to the height the detector is conjugated at, the integration and delay times, the observing band, the average quantum efficiency in the observing band and a selection of noise sources. If an even number of pixels is chosen, the optical axis is assumed to be at the junction of the four central pixels, while in the case of an odd number of pixels the optical axis is coincident with the central pixel. The sky background contribution is added to the resulting image before considering the different noise sources, the information on the background value in the observing band having been previously chosen within module SRC or any equivalent module. The noise sources considered by module IMG are Poisson photon noise, Gaussian dark-current and read-out noise (RON). All of them are assumed to be spatially white as zero correlation between pixels is assumed. The user is prompted to supply the seeds (for random number generation) for each noise source and the rms values for the Gaussian statistics describing the read-out and dark-current noise. Note that the Poisson noise processes are indeed modelled via a Monte Carlo approach. It is up to the user to decide if the image on the detector array takes into account any of the above sources of noise. Repeatability of noise is guaranteed using the same user-defined initial seed values.

3.3 Wave-front sensing

Two types of wave-front sensors have been widely modelled and developed within the software package CAOS, namely the well-known Shack–Hartmann sensor, and the relatively novel pyramid sensor. In the following we detail both of them, though with closer attention being paid to the physical modelling of the more interesting case of the latter.

3.3.1 Shack–Hartmann wave-front sensing

The basic scope of an end-to-end Shack–Hartmann module within CAOS is to compute the guide object image under each subaperture of the Shack–Hartmann array, combine them into a full sensor image, resize it to the CCD scale, integrate the image over the integration time, consider the various noise contributions, and, after waiting for a possible time delay, deliver/output the resulting noisy sensor images from which the slopes under each subaperture will be calculated.

With respect to previous versions, our present Shack–Hartmann module (SHW) – limited to square subapertures in a grid arrangement – is faster and its formal implementation is closer to that of module IMG, permitting easier understanding and maintenance of the whole package. Module SHW delivers a single output consisting of an array of images, one per lens in the lenslet array. This array of images is referred to as the Shack–Hartmann spots and will normally be fed to a barycentre quad-cell centroiding (BQC) module, which will compute the local slopes from the Shack–Hartmann spots by employing either a barycentre or a quad-cell algorithm (for further details see Section 3.5).

The parameters set by the SHW GUI are the distance from the entrance pupil to the height the CCD is conjugated at, the number of subapertures along the pupil diameter (i.e. the linear number of subapertures), the minimum illuminance required for a subaperture to be considered active, the FoV seen by each subaperture, the number of pixels per subaperture side (i.e. the linear number of pixels), the CCD plate scale (i.e. the pixel size $\Delta_{\rm CCD}$), the observing band, the average CCD quantum efficiency on this band and selection of noise sources as in module IMG. Moreover, as with module IMG, SHW assumes the optical axis of each subaperture to be at the junction of its four central pixels if an even number of pixels is used, while the optical axis is taken on the centre of the central pixel when the linear number of pixels per subaperture is odd. Note, however, that the use of an odd number of pixels per subaperture is not considered to be the most favourable arrangement (Hardy 1998).

Module SHW uses the same ideas as module IMG in order to avoid interpolations when going from the resolution obtained with the FFT operations ($\Delta_{\rm FFT}$) to the resolution imposed by the sensor CCD pixel size $\Delta_{\rm CCD}$. In fact, module SHW can be viewed as a generalization of module IMG, the only important difference being the fact that IMG outputs both the image and the PSF, while SHW delivers only the image from the lenslet array. An important implementation difference with respect to IMG occurs when the simulation is designed so that the linear number of pixels along a pupil diameter is not an integer multiple of the number of subapertures along the diameter (for instance, 256 wave-front pixels and 10 subapertures along the pupil diameter). In such cases module SHW is forced to perform a bilinear interpolation of the wave-front which will have an impact on the computation time and could slightly change the actual statistics of the assumed turbulence model. When this occurs a warning message is then triggered without stopping the simulation. Another difference with respect to the IMG module occurs when using a three-dimensional LGS. In this case SHW computes the projected two-dimensional map as seen from each subaperture. The spot produced by a given subaperture is then obtained as the convolution of the PSF for this subaperture with the corresponding LGS two-dimensional map projection as seen by this subaperture.

3.3.2 Pyramid wave-front sensing

The pyramid wave-front sensor, invented by Ragazzoni (1996), is based on the Foucault test method for optical systems. The main difference with respect to the original Foucault test is the replacement of the knife edge by a glass pyramidal optical element where the function is to emulate two knife edges in two perpendicular directions placed in the focal plane. Fig. 2 describes the set-up of a pyramid sensor: the pyramid is located in an image plane and the four pupils, corresponding to each of the four pyramid facets, are imaged by a relay lens on the pupil plane detector. A dynamic modulation of the beam (following either a circular or a square path) is applied by means for example of a tip–tilt mirror placed in a pupil plane. On the same figure the main steps of the algorithms implemented in module PYR are represented, which gives the final image on the detector. The modulation path is discretized by the user. The first step consists in computing the electric fields Ic_m on the pyramid for each point of the modulation of index m. This is performed by a simple Fourier transform (FT) of the complex amplitude that corresponds to the incoming wave-front φ to which is added the tilt $tilt_m$ corresponding to the position on the modulation path:

$$Ic_m(u, v) = \mathrm{FT}(\exp[-\mathrm{i}(\varphi + tilt_m)]). \quad (1)$$

The second output of module PYR is the resulting image on the top of the pyramid during one modulation cycle, represented by a discretized circle in Fig. 2. It is computed by summing up the squared modulus of all Ic_m values. The computation of the final four pupil images on the detector can be obtained in two ways, summarized in Fig. 3.

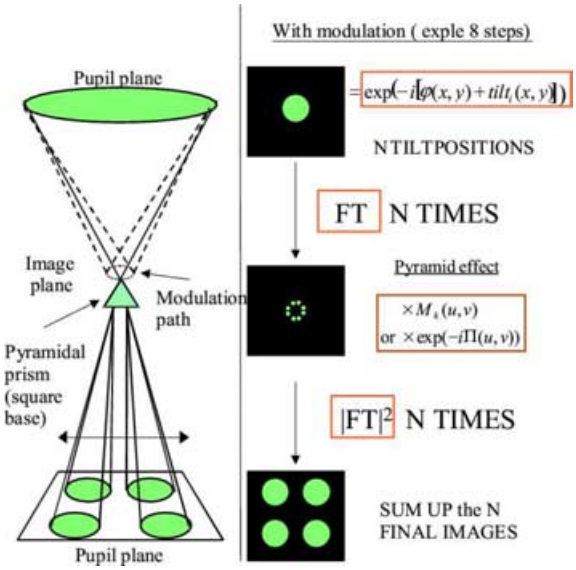

Figure 2. The pyramid wave-front sensor set-up (left) and algorithms (right) for simulating the modulation with N discretized points: the pupil plane image corresponding to each modulation point is computed independently from the others and the whole set of images are then finally summed.

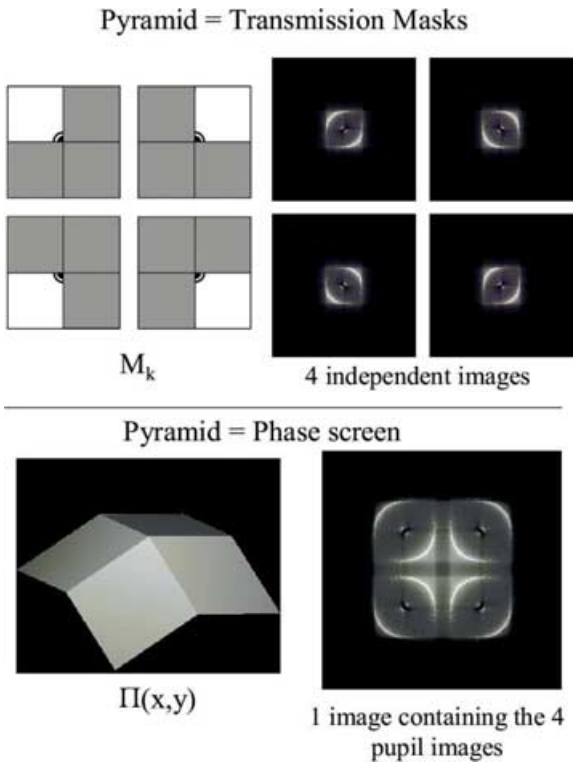

Figure 3. The two methods for computing the pyramid signals. Top, each facet of the pyramid is equivalent to a binary transmission mask and four independent pupil images are computed. Bottom, the pyramid is considered as a phase mask and only one image containing the four pupils is computed.

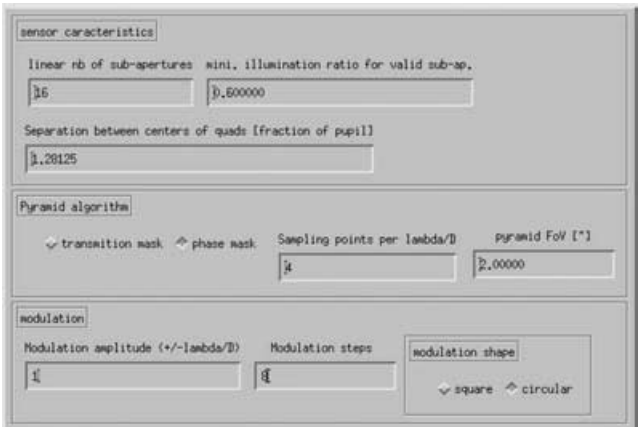

Figure 4. Part of the GUI of module PYR, showing the main relevant parameters. The other physical and numerical parameters (concerning the various noises, the spectral sensitivity, the time integration and delay, etc.) are not shown here.

The first method, already described in Esposito & Riccardi (2001), considers the pyramid as four independent transmission masks $M_k(u, v)$. Each of the four pupil images $Iccd_k(u, v)$ is then computed by summing the results for each modulation points:

$$Iccd_k(u, v) = \sum_m |\mathrm{FT}(Ic_m(u, v) \times M_k(u, v))|^2. \quad (2)$$

The second method, which is fairly new, considers the pyramid as a single phase mask $\Pi(u, v)$. Thus all the four pupils are computed in one step on one image $Iccd$ from which the $Iccd_k$ needs to be extracted for signal computation:

$$Iccd(u, v) = \sum_m |\mathrm{FT}(Ic_m(u, v) \times \exp[-\mathrm{i}\Pi(u, v)])|^2. \quad (3)$$

The latter method permits one to take into account the interference between the four pupil images, which depend on the separation between them. While in the first method, this interference is neglected as if the four images were infinitely distant from each other. After the computation of the $Iccd_k$ images, these are pixelized according to the number of subapertures defined, and the noise is computed as in module IMG. The measurements are then computed within module SLO:[2]

$$\begin{aligned} S_x(x, y) &= [(I_1(x, y) + I_2(x, y)) - (I_3(x, y) + I_4(x, y))]/I_0, \\ S_y(x, y) &= [(I_1(x, y) + I_4(x, y)) - (I_2(x, y) + I_3(x, y))]/I_0, \end{aligned} \quad (4)$$

where $I_i(x, y)$ is the intensity in the subaperture located at (x, y) in the quadrant i, integrated during a modulation cycle and I_0 is either the average intensity per subaperture of the incoming beam or the total intensity in the subapertures in the four quadrants.

The user can choose one or other algorithm and set other parameters in the pyramid parameters GUI (part of it is shown in Fig. 4): the separation between the four pupils (centre to centre) in the case of the phase mask algorithm, the modulation angle and the number of modulation points[3] and the PSF sampling parameter. The value of this last parameter fixes the size of the array on which the FTs are computed, which is simply the PSF sampling parameter times the number of simulation pixels in the pupil. As its name indicates, it is also equal to the number of points sampling the Airy spot.

[2] SLO denotes SLOpe computation, even though the exact nature of pyramid sensor measurements can be significantly different from wave-front slopes (see Vérinaud 2004).

[3] A typical value for the number of steps in circular modulation in order to obtain a rather uniform path is 8 points per λ/D of modulation angle.

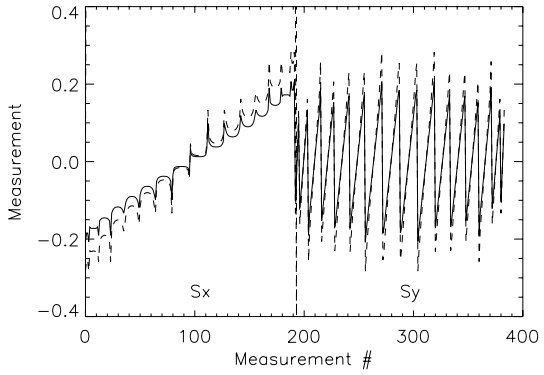

Figure 5. Measurements vector due to interference for a null incoming phase for a centre-to-centre pupil separation of 1.3. Solid line, PSF sampling = 32; dashed line, PSF sampling = 4.

The advantage of using the phase mask algorithm is twofold. At first, at a given PSF sampling it is nearly four times faster than the amplitude mask algorithm. It also permits one to take into account interference between light diffracted from one pupil to the others. Diffracted light is mainly present when no beam modulation is used and when residuals at the sensing wavelength are low. In this case, interference produces an enhancement of the light towards the centre of the image containing the four pupils (see Fig. 3, bottom part) such that even when the incoming phase is perfectly flat the measurement is not null, and can even reach rather high values, as can be seen in Fig. 5. In this figure the null measurement vector for two different values of the PSF sampling parameter is plotted. Note that the shape of the plot has no particular physical meaning and depends on the chosen arrangement of the measurement vector. However, by chance it is close to a measurement vector produced by a defocus which produces a similar enhancement of light towards the centre. On this plot we can see that when the sampling is too poor the interference is overestimated, because of aliasing in the FT. The relative error in the function of the PSF sampling is given in Fig. 6. From this result we can deduce that in order to obtain a 1 per cent accuracy for the interference modelling, the PSF sampling value should be at least 16. Thus when simulating high Strehl-ratio systems, for which

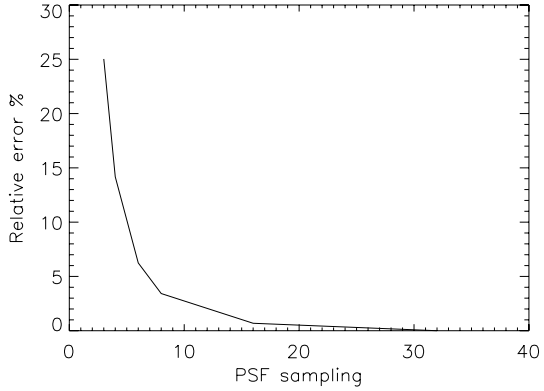

Figure 6. Error on the null measurements in function of the PSF sampling parameter. The reference measurement used to estimate the error has been computed for a PSF sampling value of 32.

interference may affect the signal, one may use a rather high PSF sampling (8–16), while for moderate-size systems with lower Strehl ratios, a PSF sampling value of 4 is generally sufficient.

3.4 Wave-front reconstruction and correction

3.4.1 Wave-front reconstruction

After measuring of the wave-front perturbations from the wave-front sensor, the computed slopes (either in the Shack–Hartmann or in the pyramid case) are sent to a wave-front reconstructor that must translate them into a series of mirror commands where the aim is to compensate for atmospheric deformations. These commands are obtained from both the wave-front sensor slope measurements and the interaction matrix that has been obtained previously by calibrating the system. Such a calibration is obtained by applying each elementary mirror deformation considered (by pushing up the actuators one by one or by applying combined modes one by one), and recording the slope measurements obtained: thus an interaction matrix is built (see Section 3.6).

A standard truncated singular-value decomposition (SVD) is implemented within the module REC in its present release, but other more refined strategies are being investigated, such as the optimal modal gain integrator (Gendron & Léna 1994) and a Kalman kind of control (Le Roux et al. 2004).

3.4.2 Time filtering

Module TFL (time filtering) implements a recursive digital filter in the time domain. Because AO systems work in a closed-loop regime, module TFL can be used as a servo control law to temporally filter the instantaneous estimation of the wave-front error to be compensated, before applying it to the wave-front corrector. The design of the servo control law is crucial in order to ensure stability of the loop and optimal performance for a given system, atmospheric conditions and reference source (Gendron & Léna 1994; Ellerbroek & Rhoadarmer 1997). The use of module TFL, however, is not limited to simulating a control law in servo configurations, it can be used whenever a time filtering is needed, such as in the case of noise filtering or to simulate the dynamical response of a deformable mirror, for instance.

Module TFL implements a traditional approach for the design of a digital recursive infinite impulse response (IIR) filter of the form

$$y_k = b_0 x_k + b_1 x_{k-1} + \cdots + b_M x_{k-M} - a_1 y_{k-1} - \cdots - a_N y_{k-N}, \tag{5}$$

using the bilinear approximation (Tustin transform, Oppenheim & Shafer 1989) of a rational analogue filter $H(s)$, defined in Laplace space, used as prototype. In equation (5) y_k and x_k are the output and input variables at the kth step, b_i and a_j are the weights of the digital filter. The bilinear transformation corresponds to the application of the trapezoidal integration rule to the differential equation associated to the analogue rational filter $H(s)$.

The prototype transfer function $H(s = \sigma + \mathrm{i}\omega)$ for the digital filter design, can be chosen from one of the following.

(i) A generic analogue filter in terms of gain G, zeros z_m and poles p_n of the form

$$H(s) = G\frac{\prod_{m=1}^{M}(s + z_m)}{\prod_{n=1}^{N}(s + p_n)}, \tag{6}$$

where the gain is a real positive number and the zeros and poles can be real or complex. In the latter case the zeros (or poles) are forced to be entered as complex conjugate couples in order to ensure real coefficients for the digital filter.

(ii) A single pole at zero frequency with a user-defined gain G, i.e. a pure integrator with $H(s) = G/s$. This is a simplified input modality with respect to the previous one.

(iii) A proportional–integrator–derivative (PID) filter given by $H(s) = K_\mathrm{p} + \frac{K_\mathrm{i}}{s} + K_\mathrm{d}\frac{\omega_0}{s+\omega_0}s$, where K_p, K_i and K_d are the gains for the proportional, integral and derivative component of the filter, respectively. A low-pass correction $\frac{\omega_0}{s+\omega_0}$ with cutting frequency ω_0 is introduced to filter out the contribution of the high-frequency noise in the derivative component.

Because the sampling frequency $\omega_\mathrm{s} = 1/T$ of the digital filter is unknown when the parameters for module TFL are chosen, the zero and pole frequencies are entered normalized to ω_s. In addition, for each of the three previous cases, the corresponding digital filter is computed and the recurrence relationship is shown. Nevertheless, the user can plot the Bode diagram (amplitude and phase versus frequency) of the prototype analogue filter. In the Bode diagram it is also possible to show the contribution of the effective delay and amplitude attenuation that is introduced by the finite frequency of the digital loop and the wavefront signal smoothing due to the non-zero CCD integration time.

3.4.3 Wave-front correction

Wave-front correction is performed by module DMI, which corrects the incident wave-front given the commands usually computed by the reconstructor REC and the time-filtering module TFL. Thus, it has two inputs: the incident wave-front to be corrected, and the commands coming from REC through TFL. It also has two outputs: the corrected wave-front that results from the difference between the two inputs (and considering the defined mirror stroke and mirror deformation series – mirror modes or influence functions), and the correction mirror shape itself (useful in order to duplicate the correction for wave-fronts coming from different objects observed together). As the mirror is mainly defined by its modes or influence functions computed elsewhere (during the calibration stage by module MDS or in any other manner, including laboratory measures), any kind of mirror can be considered here: piezo-stacked actuator mirrors, piezo-electric bimorph mirrors, electromagnetic actuated adaptive secondary mirrors, or simply ideal mirrors defined by Zernike or Karhunen–Loeve polynomials. Modelling of hysteresis is not implemented.

3.5 The tip–tilt case

Hereafter we will describe the dedicated modules TCE (tip–tilt centroid) and TTM (tip–tilt mirror) to estimate and correct from the overall tip–tilt on the entrance pupil.

When high-order sensing is performed with a LGS the use of a dedicated tip–tilt becomes nearly mandatory because of the LGS position indeterminacy (see, e.g., Rigaut & Gendron 1992). Such indeterminacy arises from the partial cancellation of the turbulence-induced motion when the laser propagates upwards from the launching telescope to the LGS altitude (i.e. focusing distance) and then downwards to the observing telescope. Several methods (see, e.g., Foy et al. 1995; Esposito 1998) have been proposed to sense the tip–tilt modes from the LGS by somehow determining the absolute LGS position, but note that even so there would be a remaining error in the tip–tilt determination because the LGS samples a cone turbulence volume (i.e. the same concept as the cone effect but applied to the tip–tilt modes). In view of these difficulties most (if not all) current LGS-based AO systems resort to determining the tip–tilt modes from an NGS. In systems with a dedicated tip–tilt sensor, a detector

placed at an alternative AO focal plane toward which some fraction of the light is diverted by a beamsplitter (module BSP in CAOS) is assumed. Given the use of a fraction of the precious natural guide star (NGS) light by the tip–tilt sensor, it is therefore a major issue to check on the necessity of such dedicated sensors when the NGS is also used to sense high-order modes. This will be the subject of the example application in Section 4.

In CAOS a dedicated tip–tilt sensor is modelled by module IMG coupled with module TCE. TCE takes the image from IMG and applies either a barycentre or a quad-cell algorithm to compute the centroid of the image as a measure of the overall tip–tilt on the entrance pupil. The barycentre algorithm estimates the x-centroid displacement $\theta_{x,\mathrm{C}}$ (equivalently for the y-centroid displacement $\theta_{y,\mathrm{C}}$ by replacing x by y) from

$$\theta_{x,\mathrm{C}} = \frac{\sum_{i=1}^{N_p^2} I_i \theta_{x,i}}{\sum_{i=1}^{N_p^2} I_i}, \tag{7}$$

where the sum extends to all pixels of the CCD detector and I_i, $\theta_{x,i}(\theta_{y,i}$ for the y centroid) correspond to the intensity, angular x coordinate (angular y coordinate) of pixel i of CCD, respectively. This diffraction approach is equivalent to the geometrical optics result for the centroid, provided that scintillation effects are negligible (Voitsekhovich, Orlov & Sánchez 2001). One should also bear in mind that pixelization effects and atmospheric coma-induced terms cause a loss of correlation between the centroid estimator and the actual tip–tilt, resulting in additional error sources (the latter effect commonly being referred to in the literature as centroid anisoplanatism). Equation (7) corresponds to the discrete case of the centroid position vector on the image plane, defined as

$$\boldsymbol{\Theta}_{\mathrm{C}} = \frac{\int I(\boldsymbol{\Theta})\boldsymbol{\Theta}\,\mathrm{d}^2\theta}{\int I(\boldsymbol{\Theta})\,\mathrm{d}^2\theta}. \tag{8}$$

If the quad-cell detector has been chosen, then module TCE estimates the image displacement according to the expressions

$$x_Q = \frac{I_\mathrm{r} - I_\mathrm{l}}{I_\mathrm{r} + I_\mathrm{l}} \quad \text{and} \quad y_Q = \frac{I_\mathrm{t} - I_\mathrm{d}}{I_\mathrm{t} + I_\mathrm{d}}, \tag{9}$$

where I_r and I_l are the total intensities to the right and left of detector centre, respectively, and I_t and I_d are the intensities on top and down from the detector centre, respectively. As shown by Tyler & Fried (1982) the quantities $I_\mathrm{r} - I_\mathrm{l}$ and $I_\mathrm{t} - I_\mathrm{d}$ are proportional to the tilt angles along the x and y axes, respectively. The denominators in x_Q and y_Q are the total intensity on the image plane, acting as normalizing factors that yield quadrant-cell measurements as estimators of relative tip–tilt angles. A calibration process is needed in order to obtain the required scaling factor, which allows the recovery of estimates of absolute tip–tilt from relative tip–tilt angle measurements. This calibration is performed by feeding a plane wave with the tip–tilt mirror at different known tilt angles. A curve of tilt angle versus TCE output is built and the required calibration constant Q is the slope of the curve in the region where a linear fit is suitable. It is possible to find an analytic estimate for Q under the assumption that the image on the detector is a Gaussian with a full-width at half maximum (FWHM) equal to the seeing angle:

$$Q = \frac{1}{4}\sqrt{\frac{\pi}{\ln 2}}\left\{\frac{\mathrm{Erf}(2\sqrt{\ln 2}\,x)}{1 - \exp[-(2\sqrt{\ln 2}\,x)^2]}\right\}\frac{\lambda}{r_0}, \tag{10}$$

where $\mathrm{Erf}(z) \equiv \frac{2}{\sqrt{\pi}}\int_0^z \mathrm{e}^{-t^2}\,\mathrm{d}t$ is the error function and x is the ratio of pixel size to the seeing angle (i.e. $x \equiv \frac{\Delta_{\mathrm{CCD}}}{\lambda/r_0}$). In the TCE GUI one has to either provide the file storing the calibration curve or the value of the Q-constant.

Finally, the tip–tilt correction can be operated using module TTM in exactly the same way as for module DMI, as we will also see in Section 4.

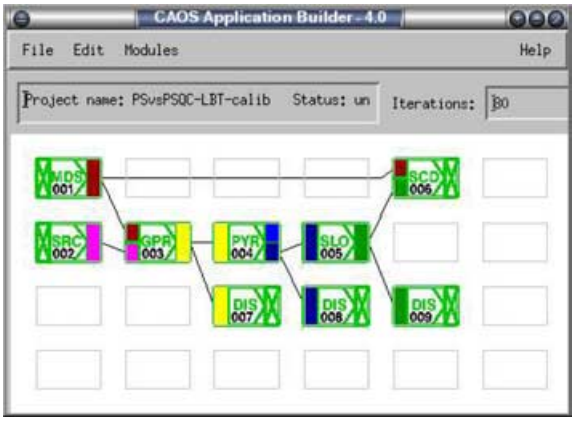

Figure 7. Project representing the calibration of a pyramid-based AO system; within the CAOS application builder and using the modules of the software package CAOS.

3.6 Modelling of the AO system calibration

Fig. 7 shows a typical calibration project within the CAOS application builder and using the modules of the software package CAOS. The case represented actually concerns the calibration of a pyramid-based AO system, the one that will then be used in Section 4.

During calibration of an AO system, typically a series of mirror deformations are sent by module MDS to the wave-front sensor module. The signals corresponding to each deformation are then sent to the module in charge of computing the resulting series of x and y slopes (from each equivalent subaperture), and then to module SCD (save calibration data), to which the mirror deformation series from module MDS also arrives. At the end of the simulated calibration, SCD saves the mirror deformations on one hand and the obtained interaction matrix on the other hand, in two separated files that will then be used by the deformable mirror and the wave-front reconstructor.

Pseudo-inversion of the interaction matrix by means of SVD, with the possibility of filtering out undesired modes, can then be operated during the very first iteration (the initialization iteration) of the subsequent actual simulation.

3.7 Other modelling modules

3.7.1 Fizeau interferometry

Fizeau interferometry has been implemented within module IBC, which permits one to combine the light beams from two telescopes which respective positions as a part of the wave-front input/output structure from the corresponding modules GPR. IBC is implemented in the simplest manner: it forms a new diluted (and atmospherically perturbed) pupil by positioning the two incoming wave-fronts. It also permits one to partially correct from the residual differential piston by selecting which percentage of it has to be maintained in the final diluted pupil. By cascading different modules IBC it is possible to combine more than two telescopes together.

This module has been used for a wide series of simulation studies concerning the Fizeau interferometer LBT (see Carbillet et al. 2002a; Correia et al. 2002; Anconelli et al. 2004), and, thanks to the multiple interferometric stage, for the VLTI (see Lardière et al. 2004).

3.7.2 Coronagraphy

Two coronagraphic modules have recently been implemented within the software package CAOS, namely module COR and module AIC.

Following the common formalism of Lyot (1939) coronagraphy and Roddier & Roddier (1997) phase mask coronagraphy introduced by Aime & Soummer (2003), who basically remark that the main difference between the two techniques resides in the coronagraphic mask present in the first focal plane (in between the first and the second pupil plane where a Lyot stop is put). Such a mask is opaque for Lyot while it is π-dephasing for Roddier & Roddier. It is then easy to model both coronagraphs by a series of three FTs: the first one translates the passage from the first pupil plane to the first focal plane (where the mask applies), the second one then goes to the second pupil plane (where the Lyot stop applies), and hence the third one goes to the second (and final) focal plane. By generalizing this concept to the four-quadrant phase mask (4QPM) technique (Rouan et al. 2000) – with the only difference being that no Lyot stop has to be applied – we have simply modelled all the three coronagraphs within module COR.

The case of the achromatic interfero-coronagraph (AIC) is different because of its interferential nature. Its implementation within CAOS was already presented in Vérinaud & Carbillet (2003), adapted from Gay & Rabbia (1996) and Baudoz, Rabbia & Gay (2000). The input, the numerical parameters, the physical parameters related to the companion star, and the output are similar to the previous case (module COR), except for the reflection and transmission factors.

More details concerning these two implementations can be found in Carbillet (2004).

3.7.3 Additional utilities

As can be noted from Table 1, a number of utilities have been developed: a generic display (module DIS) of all the kinds of input/output data used within the software package CAOS, a structure function calculator module (STF) that permits one to compute the structure function of the simulated wavefronts (or other types of obtained wavefronts) and compare the result with the theoretical formula from a Kolmogorov or von Kármán model, some modules that permit one to add/combine together wavefronts (module WFA), images (module IMA) or atmospheres (module ATA), and a couple of modules for saving (module SAV) and reading (utility RST) any kind of input/output data. Note that RST is not a proper module as it is not used through a dedicated GUI, as is also the case for utility PSG, made for generating phase screens that can be fed to module ATM later.

4 EXAMPLE APPLICATION: IS A TIP–TILT-DEDICATED SENSOR NECESSARY WHEN CONSIDERING A PYRAMID-BASED SYSTEM?

In this section we present a study based on the software package CAOS, and made within the framework of a modern 8-m class telescope. The question we want to address here is the gain we could achieve by implementing an additional dedicated RON-free quad-cell (QC) tip–tilt sensor, when already considering a system based on the pyramid sensor (PS) using a standard CCD. We first give some simple preliminary considerations, and the actual performance is then evaluated by means of end-to-end CAOS simulations, taking into account an ensemble of effects that cannot be completely modelled analytically, and thus need detailed numerical simulations. The practical parameters taken into account here are that of the LBT first-light AO system (Carbillet et al. 2003; Esposito et al. 2003).

4.1 Preliminary considerations

The main source of limitations when considering AO correction are the uncorrected atmosphere residuals, the photon starving, and the RON. For photon noise and also the contribution of RON, the total corresponding phase residual error variance $\sigma^2_{\rm noise}$ can be expressed as (Rousset 1999)

$$\sigma^2_{\rm noise} = \left(\frac{\pi\theta d}{\lambda_{\rm s}}\right)^2 \left(\frac{1}{n_{\rm ph}} + \frac{4\sigma_{\rm e}^2}{n_{\rm ph}^2}\right) [\mathrm{rad}^2], \tag{11}$$

where θ is the angular spot size on the sensor, $n_{\rm ph}$ is the number of photons per integration time and per subaperture, $\sigma_{\rm e}$ is the RON expressed in e^- rms, $\lambda_{\rm s}$ is the sensing wavelength, rad stands for radians, and the term proportional to $1/n_{\rm ph}$ is due to photon noise and the term proportional to $4\,\sigma_{\rm e}^2/n_{\rm ph}^2$ is due to the RON. Dark-current and background noise contributions are neglected here. Note that by considering this model in our case, we thus clearly assume that the pyramid roughly acts as a Shack–Hartmann sensor with 2×2 pixel under each subaperture. This is far from the physical actual situation, but it permits a couple of useful preliminary considerations.

The main consideration concerns RON. As we wish to consider a RON-free QC for measuring tip–tilt, it is clear that the corresponding measurement will only be affected by photon noise, while it will also be affected by RON when using the PS. This simple consideration implies that, if only tip–tilt were to be measured, the QC would be the best solution. However, as a portion of light must also be sent to the PS in order to measure the higher-order modes, this obviously does not apply straightforwardly. Let us call β the fraction of light sent to the QC, and let us re-write equation (11) for the specific cases of the QC and the PS, we have

$$\begin{aligned} \sigma^2_{\rm QC} &= \left(\frac{\pi\theta d}{\lambda_{\rm s}}\right)^2 \frac{1}{\beta n_{\rm ph}}, \\ \sigma^2_{\rm PS} &= \left(\frac{\pi\theta d}{\lambda_{\rm s}}\right)^2 \left(\frac{1}{n_{\rm ph}} + \frac{4\sigma_{\rm e}^2}{n_{\rm ph}^2}\right). \end{aligned} \tag{12}$$

Note for the QC expression that $d = D$ and $n_{\rm ph} = N$, where N is the total number of available photons and D is the telescope diameter. From equation (12), and stating that the scope is $\sigma^2_{\rm QC} < \sigma^2_{\rm PS}$ in order to have a better correction of the tip–tilt when using the QC, a minimum value of β respecting this condition can be calculated. Naturally the value found would not be rigorous, as here we are neglecting a large number of physically important processes/factors. For example the fact that we are interested in closed-loop operation during which different gains will be applied to the QC and the PS, and especially that the responses of both sensors are different a priori. This is sufficient to assess the existence of a minimum value of β for which a better correction of the tip–tilt is performed.

Nevertheless, β cannot reach any value, as from the higher-order modes sensing front, a minimum number of photons are also necessary in order to have an overall estimation (tip–tilt+higher orders) which is still better when using the 'PS+QC' configuration than when using the 'PS alone' configuration. Hence a trade-off between

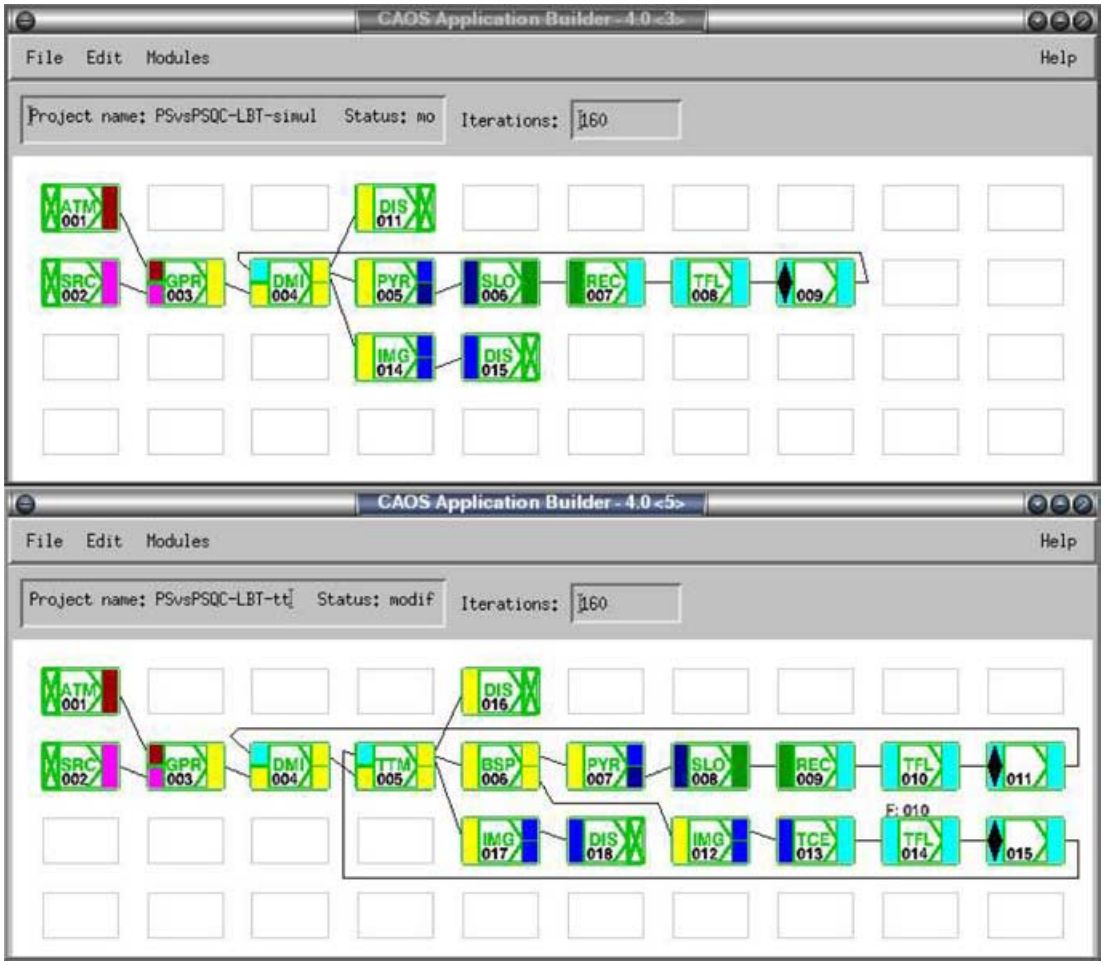

Figure 8. Projects representing the complete end-to-end simulations of the 'PS alone' case (top), and the 'PS+QC' case (bottom); within the CAOS application builder and using the modules of the software package CAOS.

those two conditions has to be found, leading to a range of values of β for which this trade-off is realized.

In the next section we will verify the existence of this range of values, also trying to find an optimal value for β, by performing complete end-to-end simulations. Thus we will compare the results obtained with that obtained when no tip–tilt separated sensor is used.

4.2 End-to-end simulation using CAOS

Fig. 8 shows the two main simulation projects that were used in order to compare the two system configurations considered here. A third project (not shown here but similar to that shown in Fig. 7) is necessary in order to calibrate the system, as already described in Section 3.6. The main difference between the two simulation projects of Fig. 8 resides in the tip–tilt branch present in the bottom part of Fig. 8: a beamsplitter (module BSP) is introduced just after the two correcting mirrors (module DMI for the higher-order modes and module TTM for the tip and tilt), sending part of the light to the tip–tilt sensor (modelled within module IMG). Hence the resulting image is sent to module TCE for measuring the corresponding tip–tilt and computation of the necessary tip–tilt commands. Module TFL is then used to filter these commands, the loop is closed thanks to the dedicated special module (part of the application builder), and the resulting correction is applied by TTM. Note that the latter could not be present in a real-life situation, as the higher-order deformable mirror could manage both types of commands, but it is clearer at least from the simulation and analysis point of view.

The main parameters of the simulation performed are reported in Table 2, and they correspond to one of the optimized situations for the first-light AO system of LBT (see Carbillet et al. 2003). Note that these optimized situations (in terms of sensor configuration, exposure time, number of modes corrected, etc.) were determined with respect to the AO guide star magnitude, the average expected atmospheric conditions, and the detailed characteristics of the system that is actually being built.

4.3 CAOS results and discussion

Fig. 9 shows the final result of our series of simulations in terms of wave-front rms residual error obtained after boot-strapping of the AO system, and hence during its stable regime. The rms plot as function of the different simulated values of β clearly shows the lower error value obtained for $\beta = 0.1$, confirming the existence of the trade-off foreseen previously.

Nevertheless, it also indicates that the achieved gain is rather modest. We have run a number of simulations with different standard conditions leading to the same conclusion, hence suggesting that a dedicated tip–tilt sensor when considering an 8-m class telescope pyramid-based AO system such as the one of LBT, does not yield

Table 2. Main parameters of the end-to-end CAOS simulations.

Turbulent atmosphere parameters	
Fried parameter r_0 (at 500 nm)	15 cm
Number of turbulent layers	2
Ground layer velocity	$\sim$8 m s^{-1}
Ground layer C_N^2 profile relative percentage	70 per cent
High layer velocity	$\sim$16.5 m s^{-1}
High layer C_N^2 profile relative percentage	30 per cent
Wave-front outer-scale L_0	20 m
Telescope parameters	
Effective diameter	8.22 m
Obstruction ratio	0.11
AO guide star parameters	
Spectral type	K5
R-magnitude	14
Deformable mirror parameters	
Type of adaptive mirror	Secondary mirror
Number of actuators	672
Wave-front reconstruction parameters	
Number of modes reconstructed	80
Time-filter type	Pure integration
Closed-loop gain	0.5
Wave-front sensing parameters	
Central sensing wavelength	750 nm
Bandwidth	300 nm
Total average transmission	0.41
Sensor configuration	15 $\times$15 (176 subap.)
Exposure time (ms)	2.5
$\Rightarrow$ corresponding $n_{\rm ph}$	$\simeq$18
Pyramid RON (e$^-$ rms)	4.5
Pyramid simulation method	Transmission mask
Pyramid modulation (λ/D)	4

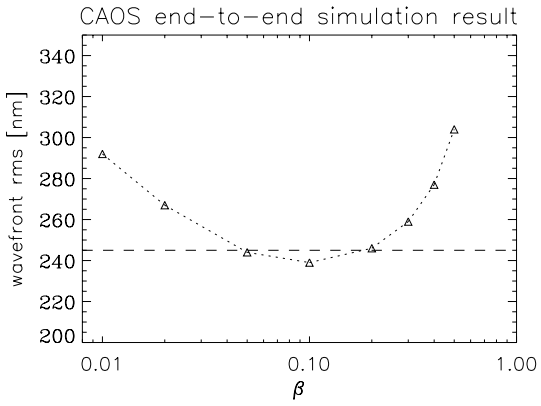

Figure 9. Wave-front rms residual resulting from the complete CAOS end-to-end simulation, in function of β. Note that the optimum value is reached for $\beta \simeq 0.1$.

a great improvement of the global system performance. Performing the same study but considering a Shack–Hartmann-based AO system would probably lead to a different conclusion, but this goes beyond the scope of this paper.

5 SUMMARY AND CONCLUDING REMARKS

In this paper we have presented the latest release of the software package CAOS, a numerical tool capable of simulating a wide range of astronomical optics problems, especially for AO and AO-related astronomy.

We have presented an example application of our tool concerning the opportunity of adding a tip–tilt sensing branch to an existing 8-m class telescope pyramid-based AO system. We have proved the existence of an optimum value of the distribution of light between the higher-order branch and the tip–tilt branch, by means of complete end-to-end numerical simulations using the modules of the software package CAOS. Moreover, we have arrived at the conclusion that the gain achievable is rather modest, suggesting that the addition of a tip–tilt branch, at least in the particular case considered here, does not lead to a significant improvement of the system performance.

The software package CAOS, version 5.0, is downloadable from the dedicated web site http://www.arcetri.astro.it/caos. It is presently delivered together with the CAOS application builder (version 4.1): the IDL-based graphical environment within which it was developed. Subscription to the dedicated mailing list is appreciated for new users. Note that the compatibility of the CAOS application builder has been extended (from present version 4.1) to Windows XP and Mac OS X platforms, in addition to native Unix/Linux.

Finally, let us note that parallel strategies are studied for CAOS (actually for all the software packages developed, with some parallel tools at a global level, i.e. at the level of the application builder itself). Some of the parallelization tools that are being developed within this framework could, in the future, be used by any existing IDL-based code.

ACKNOWLEDGMENTS

The authors wish to acknowledge the people who have contributed to early versions of the software package CAOS, namely Élise Vernet-Viard, Françoise Delplancke and Simone Esposito, but also to the people from which preliminary versions of the code benefited from their own, namely Enrico Marchetti on one hand and François Rigaut on the other hand. Serge Correia is acknowledged for his contribution to the interferometric module IBC, and Olivier Lardière for his contribution to the coronagraphic module COR. The authors also wish to thank users who gave feedback and contributions in fixing the code: Miska Le Louarn, Jeff Clifford, Stefan Hippler, Lyu Abe, Malcolm Smith, Brice Le Roux and Doug Looze. Finally, thanks are due to an anonymous referee for constructive comments on the original manuscript.

REFERENCES

Aime C., Soummer R., 2003, in Aime C., Soummer R., eds, EAS Publ. Ser. Vol. 8, Astronomy with high-contrast imaging. EDP Sciences, Les Ulis, p. 79

Anconelli B., Bertero M., Boccacci P., Carbillet M., 2004, A&A, in press

Arcidiacono C., Diolaiti E., Tordi M., Ragazzoni R., Farinato J., Vernet É., Marchetti E., 2004, Appl. Opt., 43, 4288

Baudoz P., Rabbia Y., Gay J., 2000, A&AS, 141, 319

Carbillet M., 2004, in Aime C., Soummer R., eds, EAS Publ. Ser., Vol. 12, Astronomy with high-contrast imaging. EDP Sciences, Les Ulis, p. 137

Carbillet M., Correia S., Boccacci P., Bertero M., 2002a, A&A, 387, 744

Carbillet M., Femenía B., Esposito S., Brusa G., Correira S., 2002b, in Vernet É., Ragazzoni R., Esposito S., Hubin N., eds, ESO Conf. Workshop Proc. 58, Beyond conventional adaptive optics. ESO, Garching-bei-munchen, p. 259

Carbillet M., Vérinaud C., Esposito S., Riccardi A., Puglisi A., Femenía B., Fini L., 2003, in Wizinowich P. L., Bonaccini D., eds, SPIE Proc. 4839, 131

Correia S., Carbillet M., Boccacci P., Bertero M., Fini L., 2002, A&A, 387, 733
Ellerbroek B. L., Rhoadarmer T. A., 1997, J. Opt. Soc. Am. A, 14, 1975
Esposito S., 1998, in Bonaccini D., Tyson R. K., eds, SPIE Proc. 3353, 468
Esposito S., Riccardi A., 2001, A&A, 369, L9
Esposito S. et al., 2003, in Wizinowich P. L., Bonaccini D., eds, SPIE Proc. 4839, 164
Femenía B., Devaney N., 2003, A&A, 404, 1165
Fini L., Carbillet M., 2003, in Payne H. E., Jedrzejewski R. I., Hook R. N., eds, ASP Conf. Ser. Vol. 295. Astron. Soc. Pac., San Francisco, p. 347
Fini L., Carbillet M., Riccardi A., 2001, in Primini F. A., Harnden F. R., eds, ASP Conf. Ser., Vol. 238, Astronomical data analysis software and systems X. Astron. Soc. Pac., San Francisco, p. 253
Foy R., Migus A., Biraben F., Grynberg G., McCulloch P. R., Tallon M., 1995, A&AS, 111, 569
Gay J., Rabbia Y., 1996, C. R. Acad. Sci. Paris, 332 (IIb), 265
Gendron É., Léna P., 1994, A&A, 291, 337
Gendron É. et al., 2004, A&A, 417, L21
Goodman J. W., 1968, Introduction to Fourier Optics. McGraw-Hill, New York
Habart É., Natta A., Testi L., Boulanger F., Carbillet M., Dartois E., d'Hendecourt L., 2004, VLT + NACO observing time proposal #072.C-178, 2004, VLT+NACO proposition #072.C-178, Carbon Dust in Protoplanetary Discs
Hardy J. W., 1998, Adaptive Optics for Astronomical Telescopes. Oxford Univ. Press, Oxford
Jolissaint L., Véran J.-P., 2002, in Vernet É., Ragazzoni R., Esposito S., Hubin N., eds, ESO Conf. and Workshop Proc. 58, 201
Lacombe F. et al., 2004, A&A, 417, L5
Lagrange A.-M. et al., 2004, A&A, 417, L11
Lardière O., Mourard D., Patru F., Carbillet M., 2004, in Traub W. A., Monnier J. D., Schöller M., eds, SPIE Proc. 5491, p. 415
Le Louarn M., Vérinaud C., Korkiaskoski V., Fedrigo E., 2004, in Bonaccini D., Ellerbroek B. L., Ragazzoni R., eds, SPIE Proc. 5490, p. 705
Le Roux B., Conan J.-M., Kulcsár C., Raynaud H.-F., Mugnier L. M., Fusco T., 2004, J. Opt. Soc. Am. A, 21, 1261
Lyot B., 1939, MNRAS, 99, 580
Oppenheim A. V., Shafer R. W., 1989, Discrete Time Signal Processing. Prentice-Hall, Englewood Cliffs
Ragazzoni R., 1996, J. Mod. Opt., 43, 289
Rigaut F., Gendron É., 1992, A&A, 261, 677
Roddier F., Roddier C., 1997, PASP, 109, 815
Rouan D., Riaud P., Boccaletti A., Clénet Y., Labeyrie A., 2000, PASP, 112, 1479
Rousset G., 1999, in Adaptive Optics in Astronomy. Cambridge Univ. Press, Cambridge, p. 91
Tyler G. A., Fried D. L., 1982, J. Opt. Soc. Am., 72, 6
Vérinaud C., 2004, Opt. Commun., 233, 27
Vérinaud C., Carbillet M., 2003, in Aime C., Soummer R., eds, EAS Publ. Ser. 8, 209
Vérinaud C., Esposito S., 2002, Opt. Lett., 27, 470
Vérinaud C., Arcidiacono C., Carbillet M., Diolaiti E., Ragazzoni R., Vernet-Viard É., Esposito S., 2003, in Wizinowich P. L., Bonaccini D., eds, SPIE Proc. 4839, 524
Voitsekhovich V. V., Orlov V. G., Sánchez L. J., 2001, A&A, 368, 1133

This paper has been typeset from a TEX/LATEX file prepared by the author.