FIRST GPU-BASED END-TO-END AO SIMULATIONS TO DIMENSION THE E-ELT MICADO SCAO MODE

Clénet Y.\textsuperscript{1,a}, Gratadour D.\textsuperscript{1}, Gendron E.\textsuperscript{1}, Rousset G.\textsuperscript{1}, and Sevin A.\textsuperscript{1}

LESIA, UMR 8109 CNRS, Observatoire de Paris, Université Paris Diderot, 5 place Jules Janssen, 92190 Meudon, France

Abstract. We present here first results of end-to-end simulations done to dimension the SCAO mode that will be included in ELT-CAM/MICADO, the E-ELT first light imager. These simulations have been done using YOGA\textsubscript{AO}, a GPU-based end-to-end AO simulation software, and address the SCAO performance evaluation in terms of observing conditions and wavefront sensor configuration. We also make various comparisons between YOGA and YAO, a CPU-based end-to-end AO simulation software, in terms of performances and results.

1 The MICADO SCAO mode

MICADO is the E-ELT first light imager, working in the near-IR at the telescope diffraction limit with a moderate field of view of $\sim$1 arcmin \cite{1}.

A first design of MICADO has been studied during the phase A of the project. While MICADO is to be ultimately coupled to the MCAO module MAORY \cite{2}, forming ELT-CAM, this phase A design included an internal SCAO mode \cite{3} for the following reasons:

– scientific complementarity: the SCAO mode will provide a better AO performance for single target science, such as exoplanets, solar system bodies, stellar physics, AGNs
– complementarity for operations: the MICADO SCAO mode will be usable during the MAORY commissioning phase
– risk management: with its SCAO mode, MICADO will be able to deliver diffraction-limited images at the E-ELT scientific first light even in case of a delay in the MAORY planning

This phase A design of MICADO with its SCAO mode included:

– a relay optics, consisting in a simple Offner relay, to transport the beam from the telescope output focus to the MICADO input focus, accounting for the gravity-invariant position of the MICADO cryostat;
– a field derotator, both for MICADO and the SCAO wavefront sensor (WFS);
– the WFS assembly, made of the dichroic plate, a pupil steering mirror, a field stop, a K-mirror (pupil rotation), pupil imaging lenses, the Shack-Hartmann (SH) microlenses, the camera itself. First simulations indicated no need for an atmospheric dispersion compensator.

The WFS main features were the following:

– Shack-Hartmann
– bandpass: 0.45-0.8 $\mu$m
– global transmission: 0.54

\textsuperscript{a} yann.clenet at obspm.fr
- patrol field of view: 0.45" in diameter

This design is currently under revision and update to take into account the updated telescope design (diameter, exit pupil position, field curvature, interfaces) and additional specifications since the Phase A.

**2 YoGA\_AO, a GPU-based end-to-end adaptive optics simulation software**

YoGA is a Yorick plugin built on the NVIDIA’s CUDA toolkit to provide GPU acceleration from within the Yorick environment for a number of applications such as Fourier transform, matrix operations, random number generation, etc [4].

It comes with YoGA\_AO, an extension dedicated to adaptive optics end-to-end simulation. This extension makes use of the \textit{yog\_o\_jpect} class for basic features and of custom classes for atmosphere, optics, WFS, guiding sources, etc. Running both through scripting and GUI, this implementation allows one to have an easy access to all the simulation parameters from within a Yorick session.

The general software design is inspired by YAO, a CPU-based end-to-end adaptive optics simulation tool developed by F. Rigaut (http://frigaut.github.io/yao/index.html).

The main AO features of YoGA\_AO are the following:
- on-line multiple layers Kolmogorov turbulence generation with various altitudes, strengths, speeds, directions
- multi-directional raytracing through turbulence
- multi-directional SH spots computation using either natural or laser guide stars
- centroiding on SH spots using various algorithms: weighted COG, thresholded COG, brightest pixels COG
- multiple targets

The perspectives of this work is to develop a common framework for AO simulations [5]

**3 YoGA\_AO testing versus YAO**

Before running ELT-scale simulation with YoGA\_AO, we have made several tests of the simulation tool.

The first test concerned the turbulent phase screen generation. In YAO, the phase screens are by default generated by Fourier previously to the simulations, stored on the disk and loaded during at the beginning of the simulation and translated over the pupil during the simulation. In YAO, we have also used an alternative method ("extrude method", [6]) consisting in the extrusion on the fly during the simulation of Kolmogorov-type phase screen ribbons, allowing one to store in memory only the phase screens at the pupil dimensions. In both cases, the used random number generation is done accordingly to the default Yorick implementation, i.e. using the Box-Muller method. In YoGA\_AO, the phase screen generation is following the "extrude method". We have tested in YoGA\_AO two random number generators: xor-shift [7] and Mersenne twister [8].

Figure 1 shows phase screens power spectra for the different random number generators and simulation tools. Comparisons have been made averaging 5000 power spectra for each random
number generator. On the left is compared the extrude method in YAO and in YoGA_AO. One can observe a noticeable dispersion in the power spectra. For YoGA_AO, the Mersenne twister generator provides the least dispersion and was then finally selected. On the right, the spectra were compared to the -11/3 Kolmogorov power law. All power spectra are smoothly going away for the Kolmogorov spectrum at large spatial frequency. At low spatial frequency one can observe a steeper behavior, with a larger amplitude, for the extrude method in YAO and YoGA_AO compared to YAO and its default Fourier method.

![Power Spectra Comparison](image)

**Fig. 1.** Left: circular means of phase screen power spectra for different random number generators: xor-shift for YoGA_AO extrude (black), Mersenne twister for YoGA_AO extrude (red), Box-Muller for YAO extrude (green). In each case, the 3 curves correspond to the mean of 5000 spectra, to the mean of 5000 spectra + their 1 sigma error, to the mean of 5000 spectra - their 1 sigma error. Right: circular mean of phase screen power spectra. Black: YOGA_AO extrude (Mersenne twister), blue: YAO extrude, green: YAO Fourier. In red, a -11/3 power law with an arbitrary normalization.

The second test concerned the SH measurements: providing in YAO and YoGA_AO a numerically identical phase in input of a SH, it resulted in numerically the same measurements.

We have also checked that, with identical dimensioning parameters, the deformable mirror (piezo-stack and tip-tilt) influence functions are numerically the same.

The following test concerned the interaction/command matrices. For a still unknown reason, we noticed a slight difference between YAO’s and YoGA_AO’s ones (cf. Figure 2). We are investigating this bug.

Finally we have run complete simulations with YAO and YoGA_AO with the same following simulation parameters: an 8 m telescope diameter, a 16×16 SH WFS, a 17×17 piezo-stack deformable mirror, a tip-tilt mirror, a least square with SVD truncation control, a 288×288 pupil sampling, 1 atmospheric layer.

To separate the impact of the different phase error terms, we have first made simulations with no noise. Figure 3 shows the simulation results for 1 and 2 frame delay(s) and two wind speed values (20 and 27.8 m/s). YAO and YoGA_AO Strehl ratios measured with the lowest wind speed are almost identical. With the fastest wind speed, YOGA_AO Strehl ratios are diverging with respect to YAO’s ones when the seeing is getting worse, but the difference is less than 4%
Fig. 2. Left: YOGA_AO interaction matrix with respect to the YAO one. In red, $y = x + 0.0002$ and $y = x - 0.0002$. Right: YOGA_AO command matrix with respect to the YAO one. In red, $y = 0.9 x$. Interaction matrix and command matrix points are in arcsec/volt and volt/arcsec respectively.

up to a seeing of 1.3″. Noticeably and unexpectedly, YOGA_AO performance measurements are always better than YAO’s ones.

Fig. 3. No noise simulations: 1 (upper curves) or 2 (lower curves) frame delay, wind speed of 27.8 m/s (left) or 20 m/s (right). Red: YoGA_AO, blue: YAO with Fourier phase screen generation, green: YAO with extrude phase screen generation.

We have then run simulations adding noise, with RON=3e-, but without any pixel threshold. Results with YAO and YoGA_AO are very close one from each other (Figure 4). We have then thresholded the SH measurements, either with the classical thresholding method (value=value-threshold, if value<0 then value=0) or with the so-called ”podium” method (if value<threshold then value=0). YoGA_OA Strehl ratio measurements are close to YAO’s ones, with a difference always smaller than 3%. Noticeably and unexpectedly, YOGA_AO performance measurements are always worse than YAO’s ones.
Fig. 4. Left: Simulations in case of high (upper curves) or low (lower curves) flux, no threshold. Right: Simulations in case of low flux, threshold=3, classical (upper curves) or “podium” (lower curves) threshold. RON=3e-, 2 frame delay, wind speed of 20 m/s. Red: YoGA_AO, blue: YAO with Fourier phase screen generation, green: YAO with extrude phase screen generation

As a first conclusion, YoGA_AO provides almost the same results as YAO but there are still few bugs (as the differences in the interaction/control matrices) which are maybe leading to systematic, but rather small, errors.

The last comparison made between the two simulation pieces of software was of course the excution times. With YAO, installed on a 12 cores/24 threads Xeon X5690 3.7 GHz, the simulations run at ~100 iterations/s for a 128×128 pixel pupil and at ~30 iterations/s for a 288×288 pixel pupil. In comparison, with YoGA_AO using an NVIDIA Tesla M2090 GPU, the simulations run at ~700 iterations/s for a 128×128 pixel pupil and at ~620 iterations/s for a 288×288 pixel pupil.

4 MICADO SCAO mode simulations with YoGA_AO

These comparisons being made, we performed YoGA_AO simulations for the MICADO SCAO mode. The simulation parameters were the following:

- telescope: D=38.5m, central obscuration: 0.29
- atmosphere: 1 layer, v=14.5 m/s, r0=0.129 m (seeing=0.8” at 0.5 μm)
- SH: 77×77 subapertures, λ_{WFS}=0.6μm, zero point: 1.87e10 photons/m²/s, RON=3 e-, classical thresholding
- 78×78 piezo-stack DM (square geometry), tip-tilt mirror
- least square truncated with SVD control algorithm (conditioning number:22)
- pupil sampling: 1232×1232 points (5 points per r0)

The simulation parameter space was the following:

- number of pixels per subaperture: 4, 6 and 8
- pixel field of view: 0.5 \( \lambda_{\text{subaperture}}/d_{\text{subaperture}} \), 1.0 \( \lambda_{\text{subaperture}}/d_{\text{subaperture}} \), 1.5 \( \lambda_{\text{subaperture}}/d_{\text{subaperture}} \)
- loop gain: 0.1, 0.3, 0.5
- loop frequency: 100 Hz, 300 Hz, 500 Hz, 800 Hz, 1000 Hz
threshold level: 0, RON, 2 RON, 3 RON
reference star magnitude: 10, 12, 13, 14, 15, 16

It then represents 3240 simulations!
The execution times of the different steps of a simulation are the following:

- atmosphere initialization: 152 s for the first initialization, 3 s for the following ones (i.e. not requiring the computation of the extrude method parameters)
- WFS initialization: 3 s
- DM initialization (i.e. influence functions computation): 12 s
- RTC initialization: 237 s for the first initialization, 14 s for the following ones (i.e. not requiring a new interaction matrix computation)
- the simulation itself (i.e. without the initializations): ~45 iterations/s
- initialisations + simulation: less than 1 min for a given set of parameters

The results of the different simulations are presented in Fig. 5. These preliminary results show that the MICADO SCAO mode could reach more than 70% Strehl ratio for bright stars. Together with dedicated functionalities (coronographic devices, non sidereal tracking), this mode would then be particularly suited for exoplanet and solar system science. Within the tested parameter space, it appears that the best compromise for the SH configuration would be 4×4 pixels per subaperture and a pixel field of view of λ_{subaperture}/d_{subaperture}.

5 Perspectives

These preliminary results have to be first ascertained by finishing the cross-checking between the YoGA_AO and YAO and making clear that the remaining small discrepancies between them are not bugs in YoGA_AO. Still on the YoGA_AO side, we will implement additional features to better match the E-ELT design and constraints (M4 actuator pattern and influence functions, segmented pupil, vibration control, etc) and to test alternative WFS (pyramid WFS). Most of these developments will be made in the framework of the COMPASS project, led by D. Gratadour, funded by the French National Research Agency, aiming at porting key AO algorithms to GPU and at implementing a low latency data transfer between the WFS camera and the RTC.

References

1. Davies R. et al., SPIE 7735, (2010) 77352A
4. Gratadour D. et al., AO4ELT2 online proceedings, (2011)
5. Gratadour et al, this conference (2013)
Fig. 5. Upper four graphs: Strehl ratio vs. reference star magnitude at different loop frequencies (see the top of each graph), for different numbers of pixels per subaperture (continuous lines: 4, dashed lines: 6, dotted lines: 8) and pixel fields of view (black lines: 0.5 \( \lambda_{\text{subaperture}}/d_{\text{subaperture}} \), red: \( \lambda_{\text{subaperture}}/2d_{\text{subaperture}} \), green: 1.5 \( \lambda_{\text{subaperture}}/d_{\text{subaperture}} \)). A given point (i.e. number of pixels per subaperture and pixel field of view values) corresponds to the best performance over the different loop gain and threshold values. Bottom fifth graph: best Strehl ratios over the entire parameter space vs. reference star magnitude.