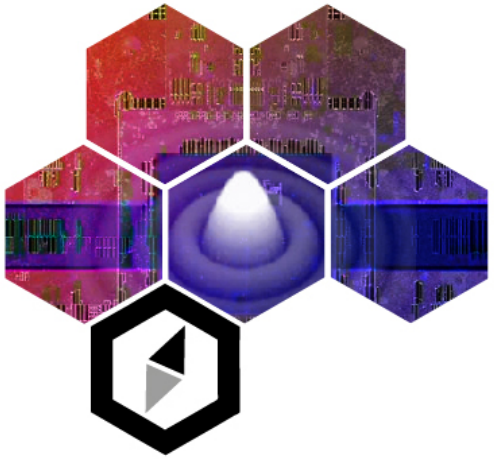
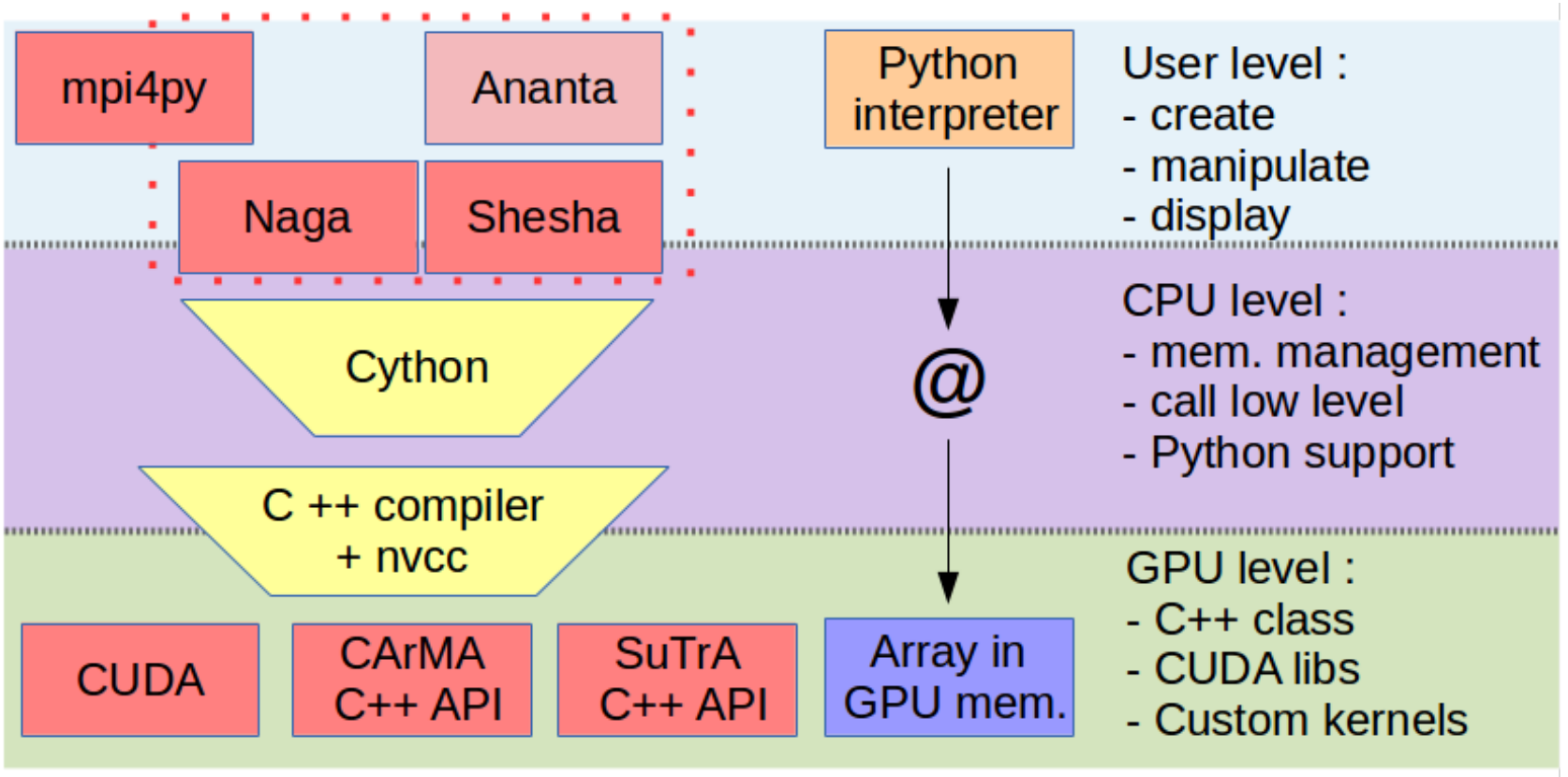


AO simulations for pyramid wavefront sensing on the E-ELT

Sébastien Durand, Florian Ferrera, Fabrice Vidal, Damien Gratadour,
Eric Gendron, Yann Clenet, Arnaud Sevin

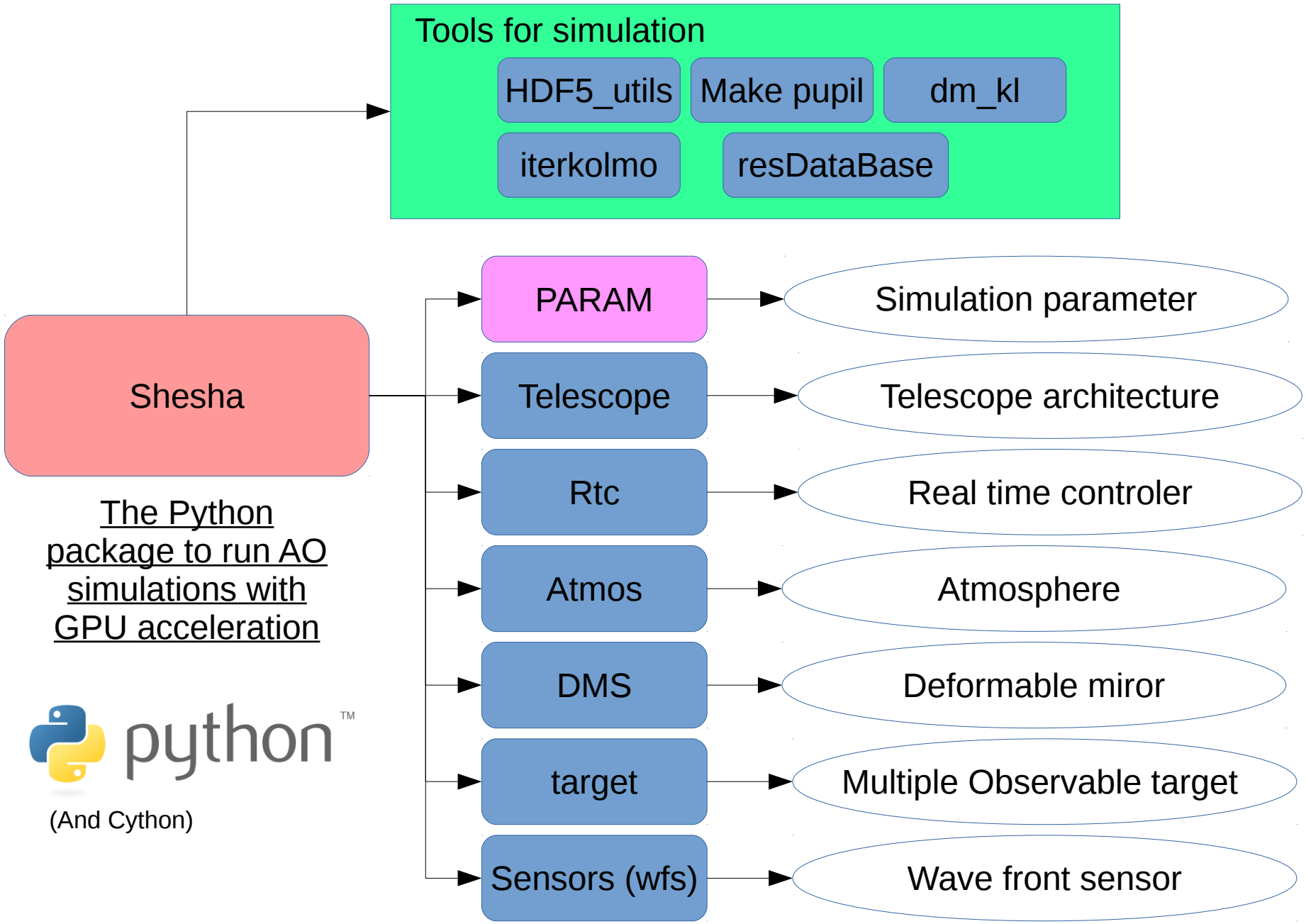
Compass Architecture

COMputing Platform for Adaptive optics SystemS



SuTrA : the AO simulation tool
CArMA : the C++ API for a user-friendly GPU
SHESHA : the Python package to run AO simulations with GPU acceleration
NAGA : the Python general library for GPU computations

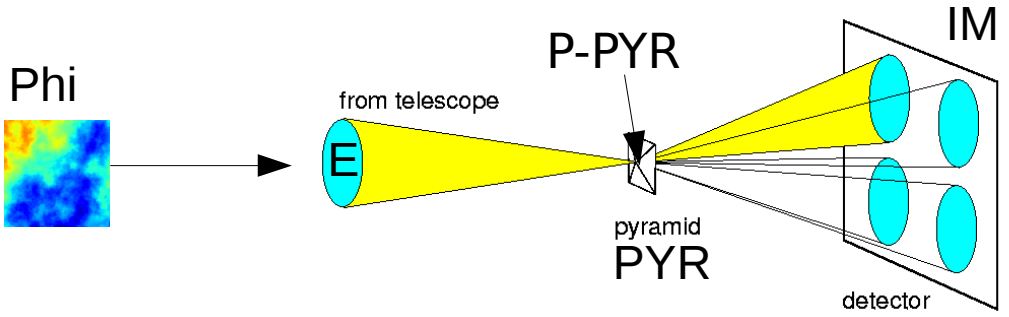
Shesha architecture



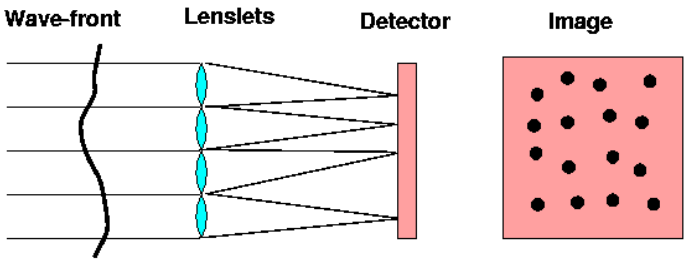
Wave front sensor on Compass

Two codes for two wave front sensors :

- Pyramid → pyrHR code (new)



- Shack-Hartmann → SH code

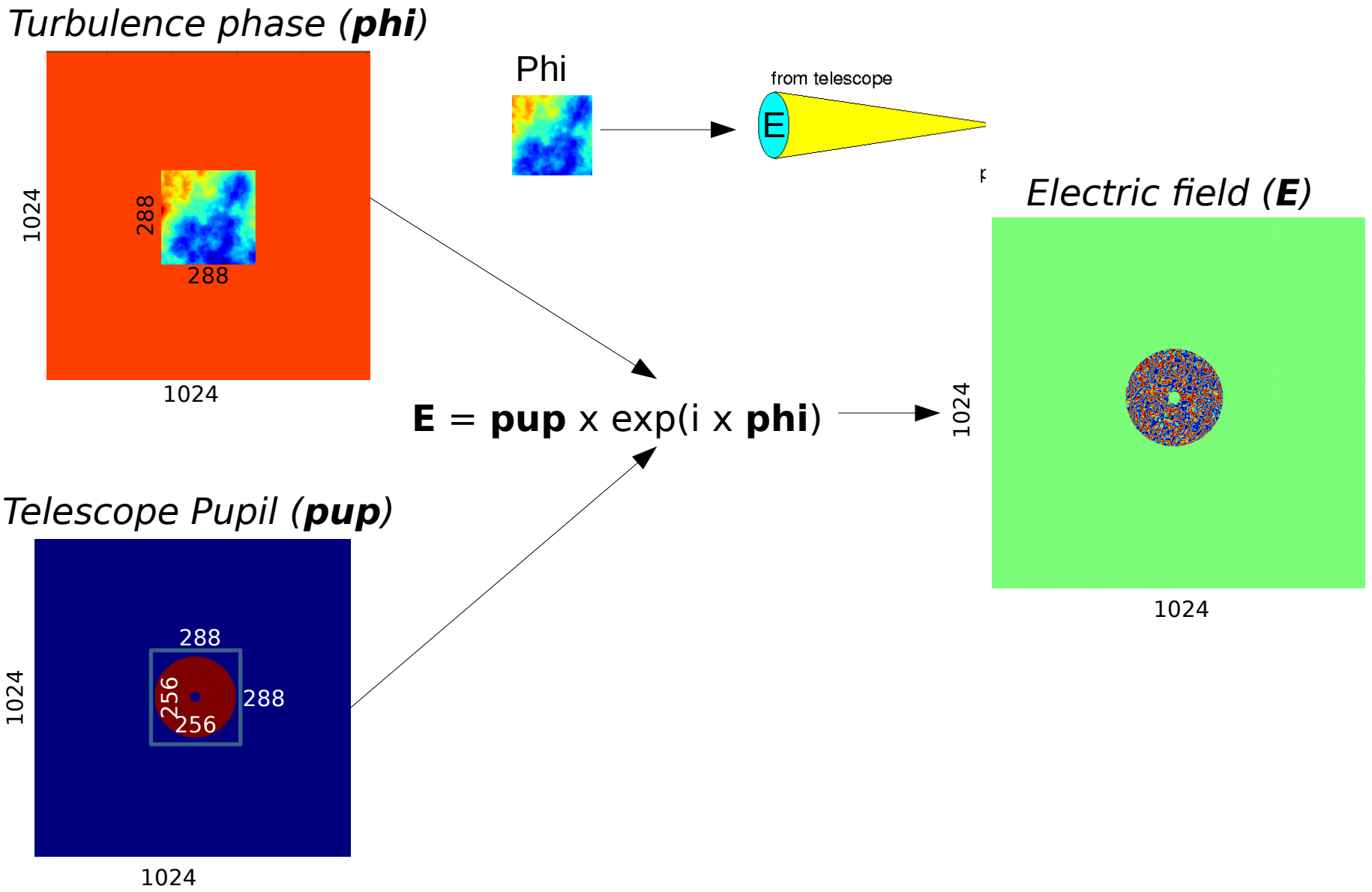


Validation for SH Code

- Validation of SH COMPASS code by comparison with the YAO simulator
- More than 2000 simulations 8 et 39m to compare the response of COMPASS and YAO simulator
- Good match between YAO and COMPASS results

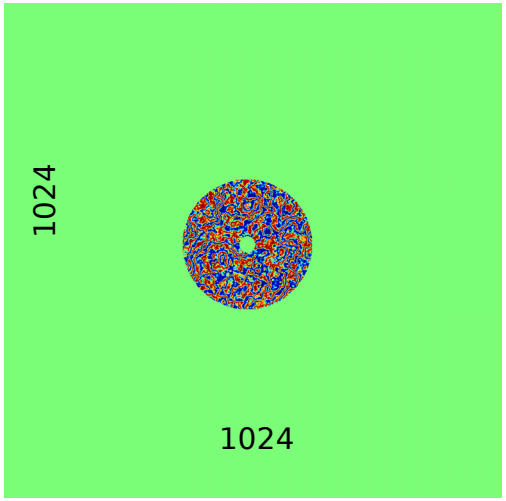
PYRHr Code

(example : 8m 16x16pixel)



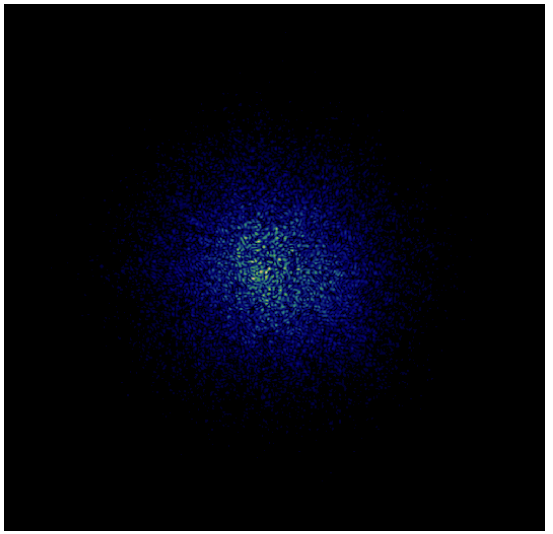
PYRHr Code

*Electric field (**E**)*



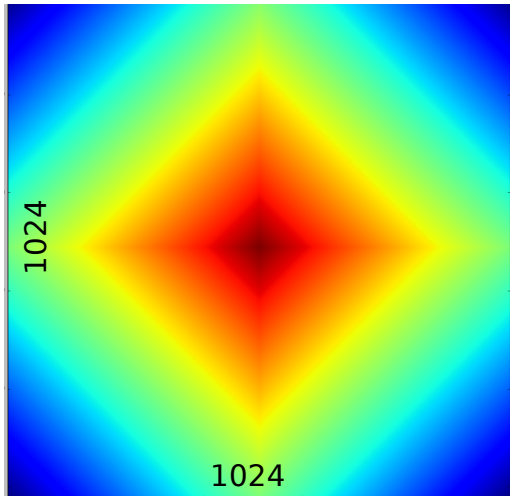
Abs(fft(E))**2

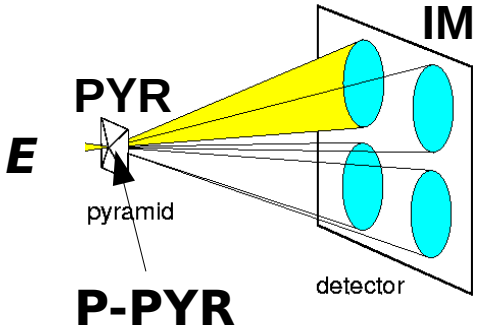
PSF on top of Pyramid (**P-PYR**)



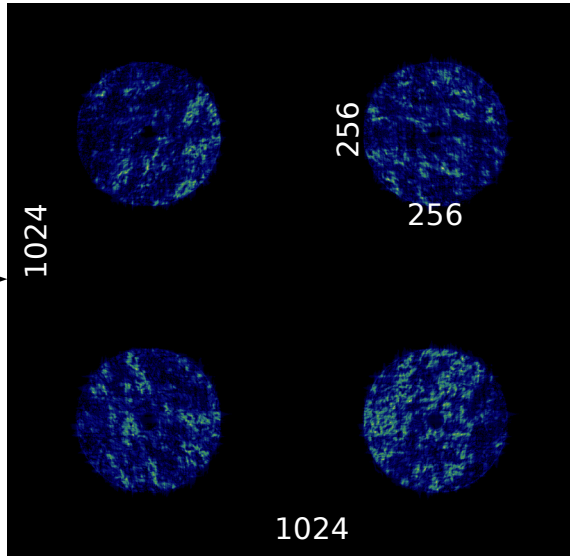
IM= abs(fft(fft(**E**)x exp(i x **PYR**)))**2

Pyramid (**PYR**)



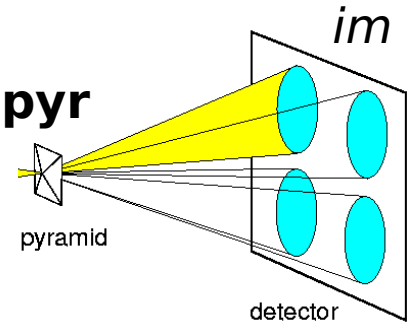


WFS Pyramid image (**IM**)



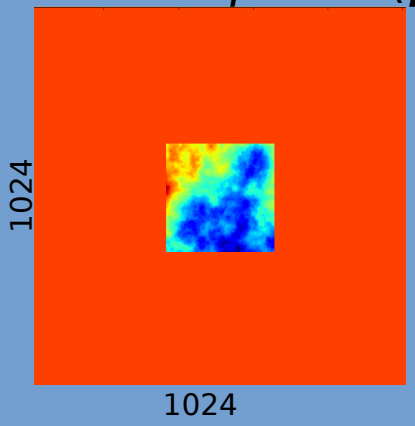
PYRHr Code

PYR Modulation

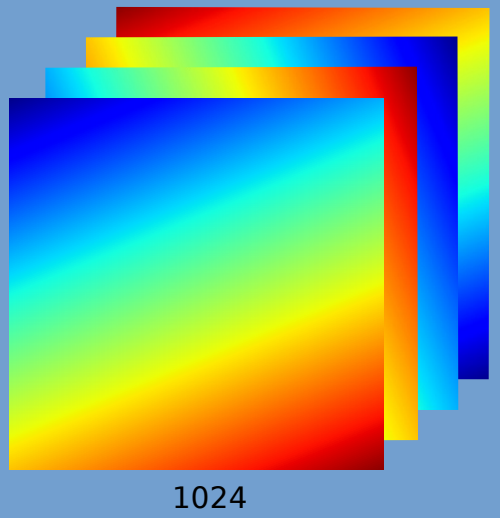


N modulation point

*Turbulence phase (**phi**)*

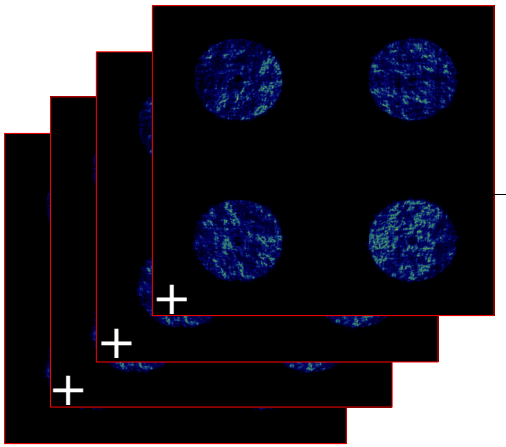


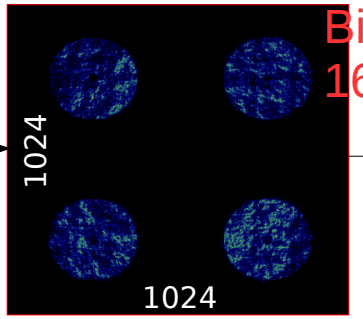
*Tilt Modulation (**mod** x N)*



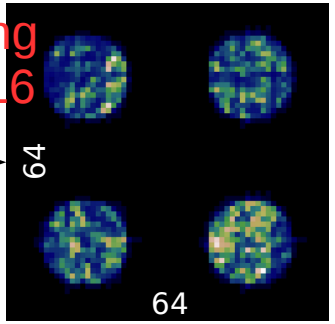
Em = **pup** x exp(i x (**phi**+**mod**))

im = Σ (abs(fft(fft(**Em**)x exp(i x **pyr**)))**2)



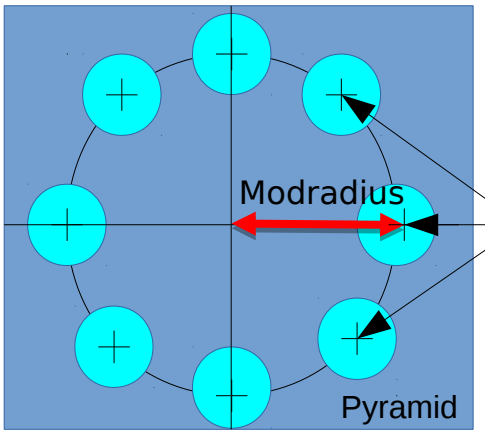


Bining 16x16



*Modulated Pyramid HR image (**im**)*

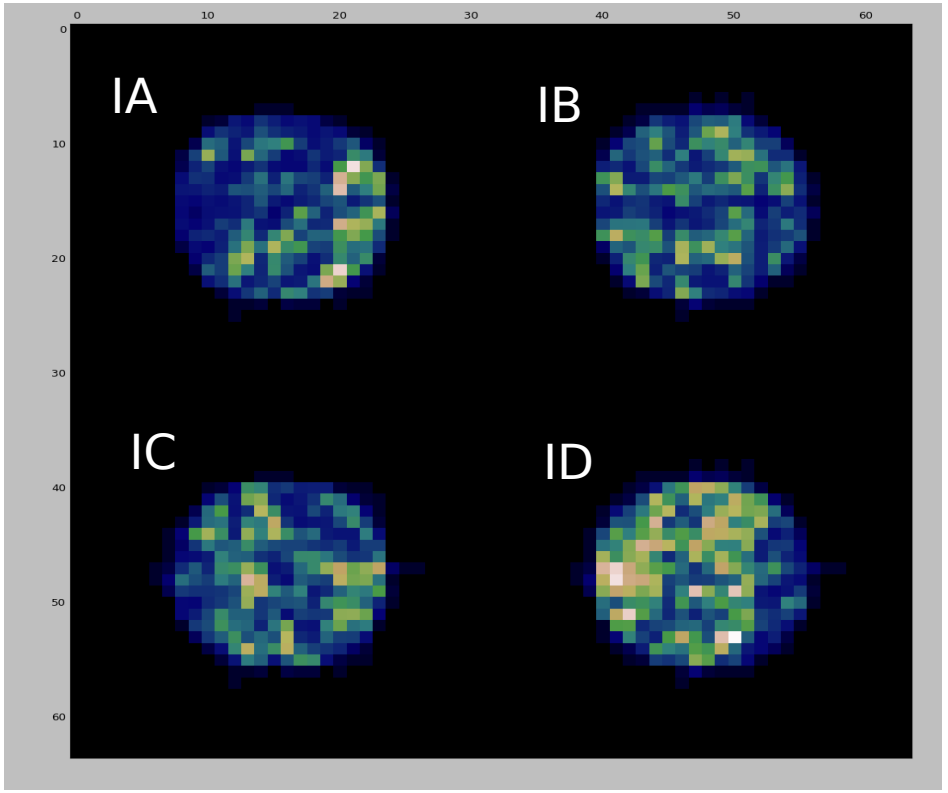
Modulated Pyramid SR image



Modposition (N)

PYRHr Code

Slope computation



Sx = ppup*(**IB**+**ID** -(**IA**+**IC**))/(Itot)
Sy = ppup*(**IC**+**ID** -(**IA**+**IB**))/(Itot)
dwx = mod x sin(0.5 x pi x Sx)
dwy = mod x sin(0.5 x pi x Sy)

With Itot = **IA**+**IC**+**IB**+**ID**

Karhunen Loeve projection

Pyramid simulation were performed with DM controlled on KL basis :

- Computation of transfer matrix from DM volts to KL modes
- Modes filtering
- Projection of the interaction matrix
- Direct inversion
- Projection back to the DM volts basis

Simulations

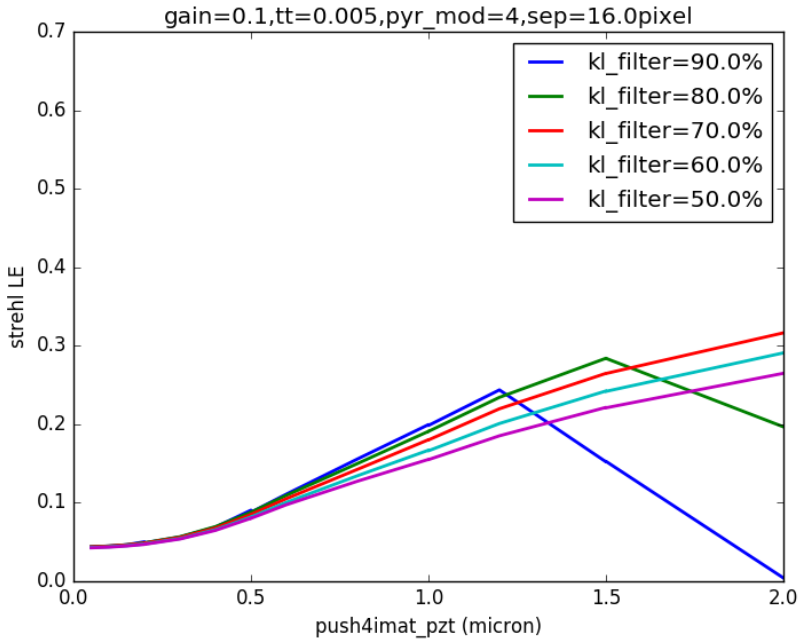
Pyramid response preliminary study parameters for MICADO :

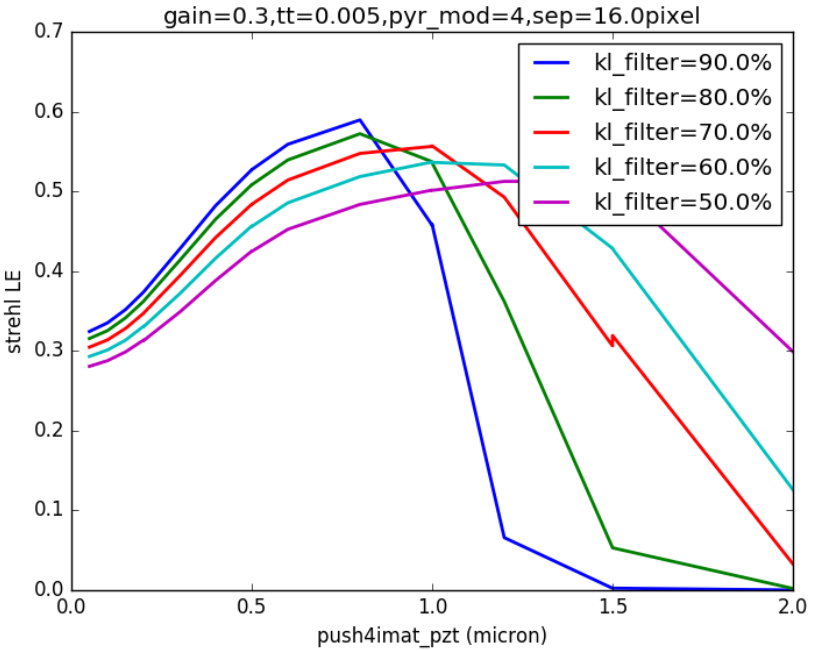
- 8 m class telescope
- 16x16 pyramid pixels
- 220 actuators + tip-tilt
- R0 = 16cm@0.5µm
- Pyramid modulation from 1 to 10 λ/D (modradius)
- Loop gain from 0.1 to 3.0
- Push/Pull value used during interaction matrix from 0,05 to 1,5µm
- 2500 simulations

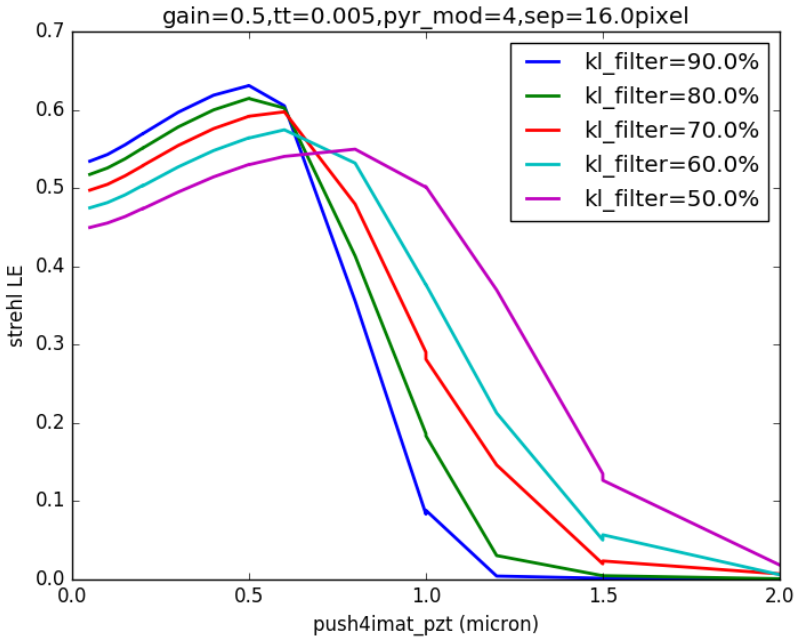
System environment :

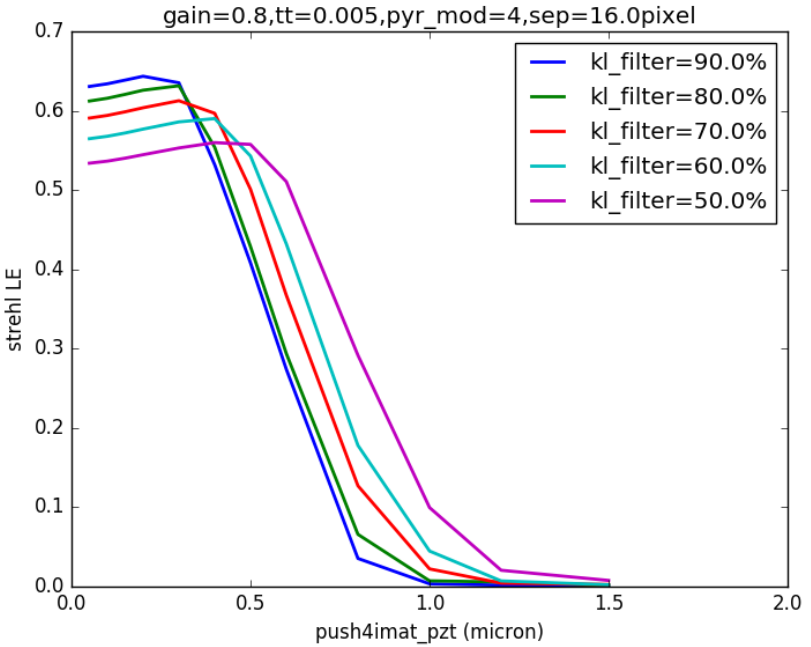
- 2xIntel Xeon E5-2630v4@2,2GHz(10 physical cores)
- 8x GPUs TITAN X Pascal@1.4GHz and 12Go G5X(3584 CUDA Core)
- 64Go Ram DDR4

Results (1)

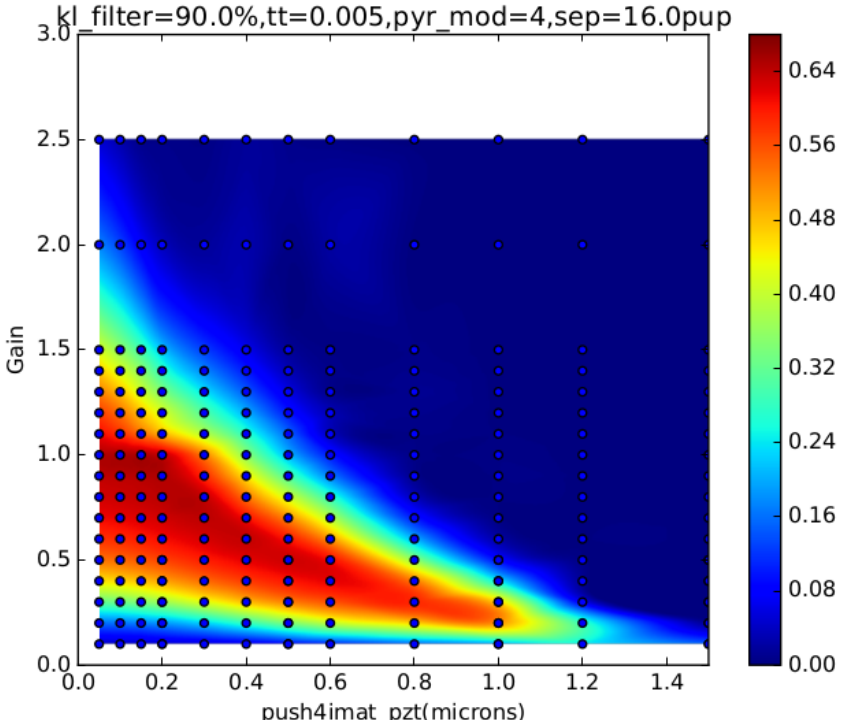

gain=0.1,tt=0.005,pyr_mod=4,sep=16.0pixel
kl_filter=90.0%
kl_filter=80.0%
kl_filter=70.0%
kl_filter=60.0%
kl_filter=50.0%
strehl LE
push4imat_pzt (micron)

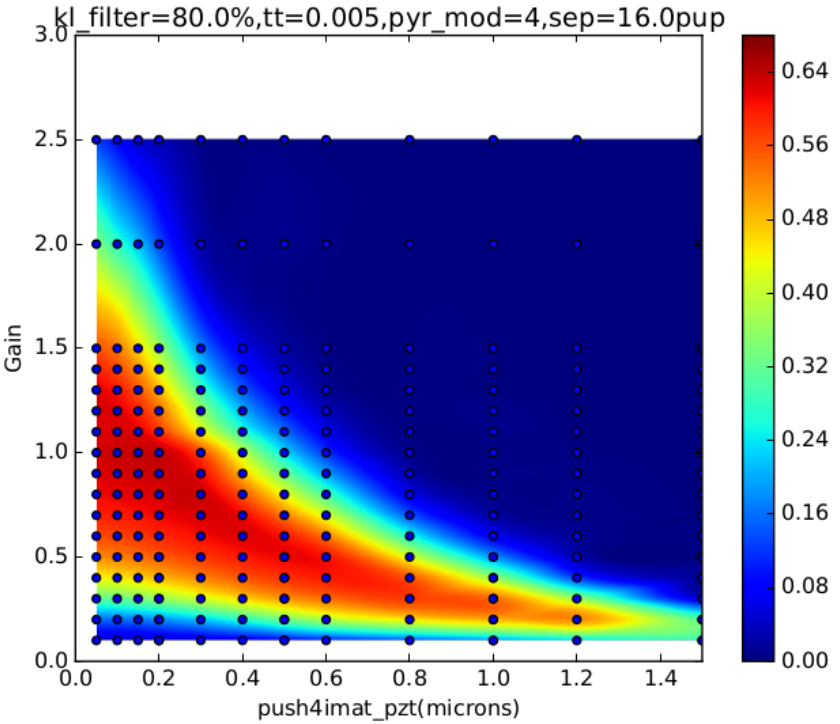

gain=0.3,tt=0.005,pyr_mod=4,sep=16.0pixel
kl_filter=90.0%
kl_filter=80.0%
kl_filter=70.0%
kl_filter=60.0%
kl_filter=50.0%
strehl LE
push4imat_pzt (micron)

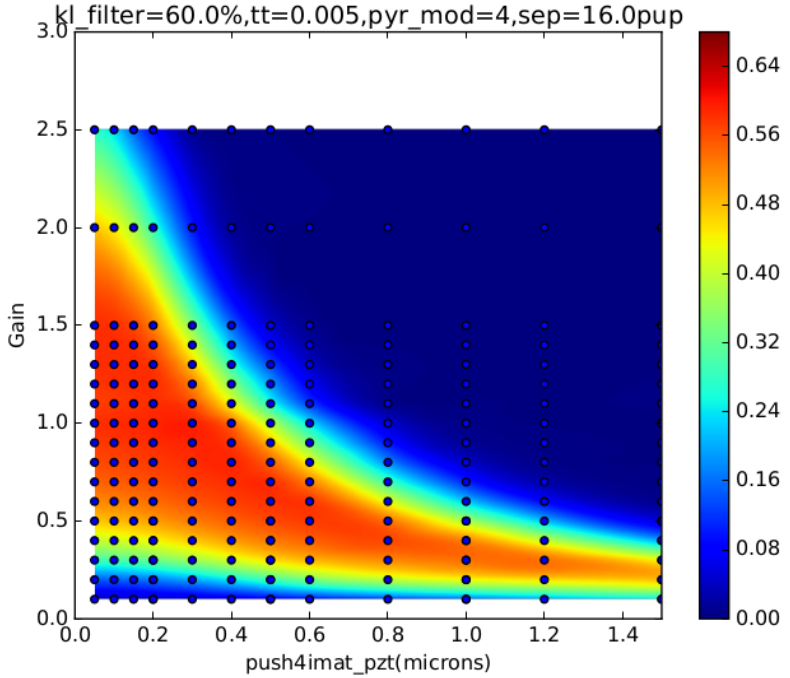

gain=0.5,tt=0.005,pyr_mod=4,sep=16.0pixel
kl_filter=90.0%
kl_filter=80.0%
kl_filter=70.0%
kl_filter=60.0%
kl_filter=50.0%
strehl LE
push4imat_pzt (micron)

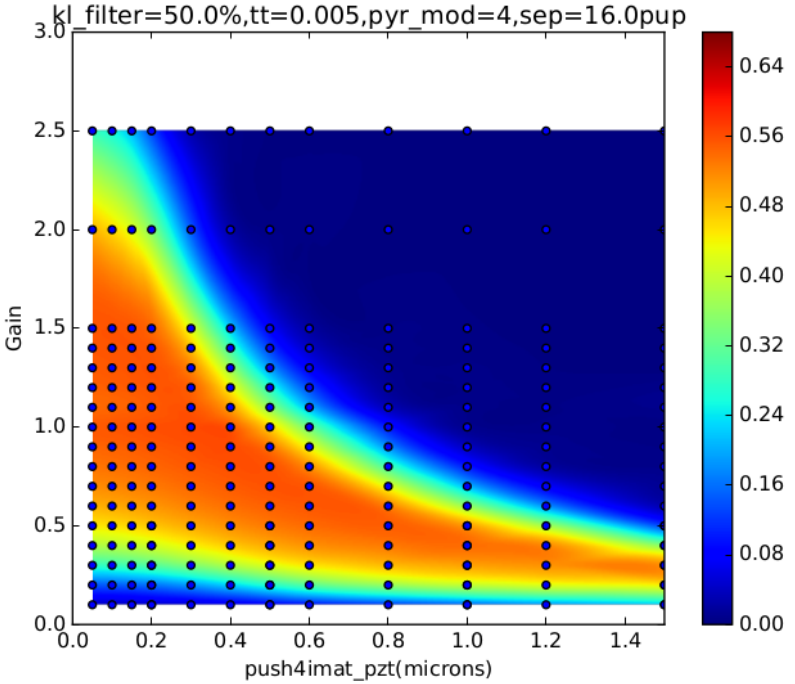

gain=0.8,tt=0.005,pyr_mod=4,sep=16.0pixel
kl_filter=90.0%
kl_filter=80.0%
kl_filter=70.0%
kl_filter=60.0%
kl_filter=50.0%
strehl LE
push4imat_pzt (micron)

Results (2)


kl_filter=90.0%,tt=0.005,pyr_mod=4,sep=16.0pup
Gain
push4imat_pzt(microns)


kl_filter=80.0%,tt=0.005,pyr_mod=4,sep=16.0pup
Gain
push4imat_pzt(microns)


kl_filter=60.0%,tt=0.005,pyr_mod=4,sep=16.0pup
Gain
push4imat_pzt(microns)

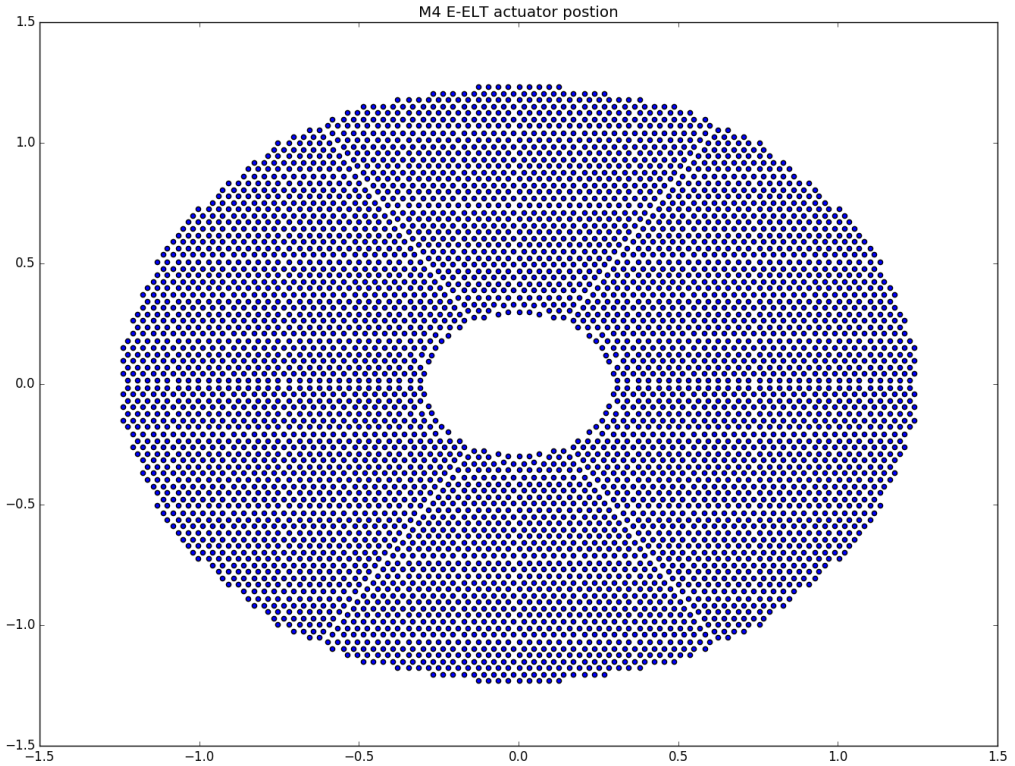

kl_filter=50.0%,tt=0.005,pyr_mod=4,sep=16.0pup
Gain
push4imat_pzt(microns)

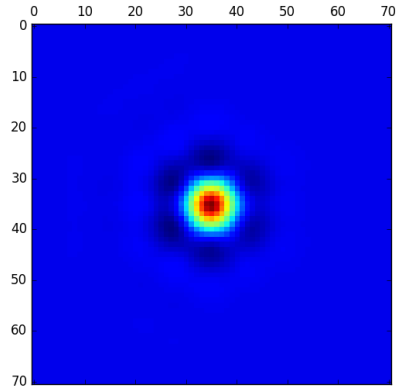
Results Exploitation

- Appearance of a symmetrical configuration for the subpupils which cancels the diffraction effect of the pupil pyramid one on the other.
- Linear domain of the pyramid allows to increase the gain of the RTC loop
- In saturation domain creation of an artificial gain that forces to lower the gain of the RTC loop.

M4 implementation

- Using ESO-M4-262903 packages
- Program to convert ESO-IDL packages into panda-frame to be read by COMPASS
- Custom DM (need : influence function, position, size, dimesion in pupil, resolution and optical center)





M4 influence function

Perspectives

- Perform full scale SCAO MICADO simulations:
  - E-ELT pupil
  - Phase aberrations on M1 segments
  - M4 influence functions
  - Pupil rotation